МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

А.С. Ватьян, Н.Ф. Гусарова, Н.В. Добренко

СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО
по направлению подготовки
45.03.04 - Интеллектуальные системы в гуманитарной сфере
в качестве учебного пособия для реализации основных профессиональных образовательных программ высшего образования бакалавриата

Санкт-Петербург

2022

УДК 004.89
ББК 32.813.5
В21

В21 Ватьян А.С., Гусарова Н.Ф., Добренко Н.В. Системы искусственного интеллекта. – СПб: Университет ИТМО, 2022. – 186 с.

ISBN 978-5-7577-0669-6

Рецензент: Ковальчук Сергей Валерьевич, кандидат технических наук, доцент мегафакультета трансляционных информационных технологий Университета ИТМО

Изложены основы современных концепций построения систем искусственного интеллекта (ИИ), таких как соотношение естественного интеллекта и ИИ, базовые технологии ИИ, стандарты и требования в области ИИ. Особое внимание уделено построению систем объяснимого ИИ. Приведены примеры конкретных реализаций систем ИИ в области обработки медицинской информации. Издание может быть использовано в качестве учебного пособия для реализации основных профессиональных образовательных программ высшего образования бакалавриата по направлению 45.03.04 - Интеллектуальные системы в гуманитарной сфере.

ISBN 978-5-7577-0669-6

Содержание

ВВЕДЕНИЕ

В 2019 году в Российской Федерации принята Национальная стратегия развития искусственного интеллекта на период до 2030 г., утвержденная указом Президента Российской Федерации от 10 октября 2019 г. № 490 [Стратегия, 2019], которая определяет цели и основные задачи развития искусственного интеллекта (ИИ) в Российской Федерации. В развитие этой Стратегии в декабре 2020 года принята Перспективная программа стандартизации по приоритетному направлению «Искусственный интеллект» на период 2021–2024 гг. [Программа, 2020]. В рамках Программы запланирована разработка 217 стандартов, которые будут регламентировать безопасность систем ИИ для людей и для окружающей среды. Стандартизация коснется внедрения ИИ в различных областях человеческой деятельности, таких как транспорт, медицина, образование, строительство и ряд других.

Принятие этих документов означает, что ИИ в РФ приобрел государственную поддержку. Как отмечается в Стратегии, развитие ИИ является необходимым условием технологической независимости и конкурентоспособности страны.

Что такое искусственный интеллект? С 2020 г. определение ИИ в РФ стандартизовано [ГОСТ Р 59277–2020]:

«Искусственный интеллект (artificial intelligence): Комплекс технологических решений, позволяющий имитировать когнитивные функции человека (включая самообучение, поиск решений без заранее заданного алгоритма и достижение инсайта) и получать при выполнении конкретных практически значимых задач обработки данных результаты, сопоставимые, как минимум, с результатами интеллектуальной деятельности человека.

Примечание. Комплекс технологических решений включает в себя информационно-коммуникационную инфраструктуру, программное обеспечение (в том числе, в котором используются методы машинного обучения), процессы и сервисы по обработке данных, анализу и синтезу решений».

Детальное рассмотрение этого определения показывает, что оно опирается не только на терминологию ИТ (информационно-коммуникационная инфраструктура, программное обеспечение, машинное обучение), но и на целый ряд понятий из смежных дисциплин (когнитивные функции человека, интеллектуальная деятельность человека, инсайт). Неформально говоря, ИИ призван программными средствами имитировать естественный интеллект.

Многие вопросы соединения естественного интеллекта и ИИ остаются открытыми даже в постановке, а ответы на них меняются по мере развития науки в целом, и соответственно изменяются подходы к трактовке и реализации систем ИИ. Тем не менее, общей базой для ИИ, безусловно, является развитие современных информационных технологий и разработка новых вычислительных средств.

Сегодня ИИ – это чрезвычайно многогранная и комплексная сфера деятельности, аккумулирующая в себе результаты многих наук, таких как генная инженерия, биотехнологии, медицина, нанотехнологии, робототехника, микроэлектроника, психология, социология и др. Создавая технологическое решение с применением ИИ, разработчик должен быстро и в то же время глубоко погрузиться

в методы и подходы, применяемые специалистами целевой предметной области, сохраняя при этом непредвзятый взгляд ИТ-профессионала на основные ее закономерности. Другими словами, практически каждая технологическая задача в области ИИ – это интеллектуальный вызов, что делает работу в этой сфере увлекательной и в то же время высококонкурентной. В этом убеждает опыт авторов настоящего пособия по разработке ИИ-решений в области медицины.

Пособие построено в соответствии с программой курса «Системы искусственного интеллекта», читаемого на мегафакультете трансляционных информационных технологий Университета ИТМО в 2022 г. Большое внимание в пособии уделяется реализации требований стандартов, в том числе разработке систем доверительного и объяснимого ИИ. Материал пособия иллюстрируется примерами конкретных технологических решений в различных сферах приложения ИИ, в том числе авторскими разработками.

1. ЕСТЕСТВЕННЫЙ ИНТЕЛЛЕКТ И ЕГО КОГНИТИВНЫЕ МОДЕЛИ

Определение ИИ ставит вопросы соотношения и объединения естественного интеллекта (ЕИ) и ИИ, в том числе: что такое интеллект вообще и естественный интеллект, в частности? что входит в когнитивные функции человека? в какой степени и что именно имитировать? В данном разделе кратко охарактеризованы сведения, которые предоставляют смежные науки, прежде всего психология и нейрофизиология, для описания ЕИ как средства переработки информации.

1.1. Мозг как физическая и психологическая основа естественного интеллекта

Нейрофизиологические характеристики мозга. В конце 1930-х–начале 1950-х гг. проводились исследования в области нейрофизиологии, которые показали, что мозг можно рассматривать как электрическую сеть нейронов, при этом правомерна схема реального нейрона как структурной единицы центральной нервной системы, представленная на рис. 1.1.

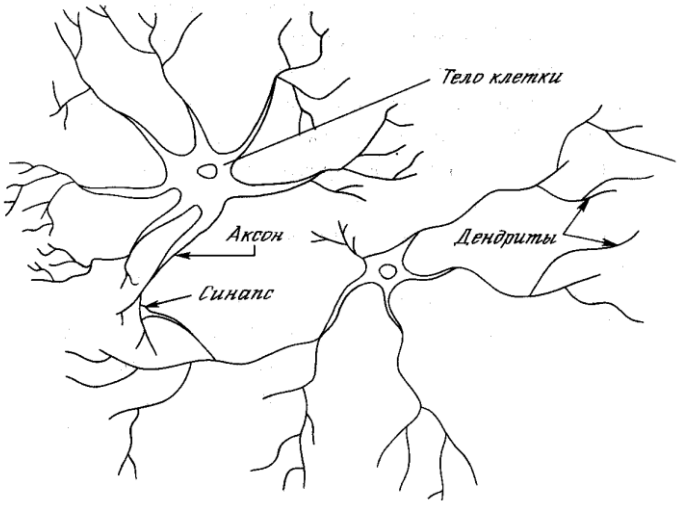


Рис. 1.1. Синаптическое соединение нейронов

Биологический нейрон в ответ на возбуждение может генерировать нервный импульс, распространяющийся вдоль аксона. Его форма и скорость распространения не зависят от того, как и из-за чего он возник. Доходя до конца аксона, он вызывает выделение веществ, называемых нейромедиаторами. Воздействуя на дендриты других нейронов, они могут в свою очередь вызвать появление в них нервных импульсов. Другими словами, на этом уровне развития когнитивных наук нейрон трактовался как управляемый логический элемент.

На сегодняшний день эта модель выглядит крайне упрощенной. Нейрофизиологи выявили множество механизмов управления нейроном, и они действуют в сложной иерархической взаимосвязи. К настоящему времени уже есть основания рассматривать каждый нейрон как отдельный компьютер или даже как компьютерную сеть.

Мозг появился в ходе эволюции у червей, у которых он представляет собою скопление 3000 нейронов. Мозг позвоночных в своем развитии проходит три стадии, соответствующие ходу эволюции. У рептилий мозг обеспечивает только поддержание гомеостаза и основные физиологические потребности животного. Мозг низших млекопитающих управляет обонянием, эмоциональным поведением, примитивным научением по принципу «вознаграждение/наказание». И только у высших млекопитающих, в том числе у человека, появился так называемый передний мозг, который в состоянии обеспечить процессы познания, в том числе осознания себя, а также высшие психические функции. Передний мозг состоит из двух полушарий, покрытых корой толщиной около 3 мм. У современного человека все эти типы мозга сосуществуют в виде отдельных слоев (отделов).

С точки зрения системного подхода именно кора головного мозга представляет собою основное аппаратное средство переработки информации, аналогом которого в компьютере является сочетание центрального процессора и памяти. В последние годы выявлены также другие механизмы регулирования информационных потоков в мозге, не имеющие аналогии в современном компьютере – в том числе механизмы параллельной фильтрации всех сигналов, идущих в мозг, с целью оценки их относительной важности. Очевидно, что это существенно экономит информационные резервы мозга.

Фундаментальным достижением психологии XX века явилось открытие функциональной асимметрии полушарий головного мозга: установлено, что каждое полушарие имеет свою специализацию по выполняемым функциям. Это позволяет рассматривать мир с двух различных точек зрения – с формально-логической (левое полушарие) и пространственно-образной (правое полушарие) позиции – и создает основу для творческой активности.

Динамика развития познавательных функций. Человек рождается, имея готовыми к действию только две формы психики – сенсорику (систему органов чувств) и моторику (систему управления двигательной активностью), остальные приобретает в ходе индивидуального развития. При этом все органы чувств готовы полностью, а моторика развивается посредством взаимодействия генетически заданных моторных актов и недифференцированных движений на базе генетически заданных механизмов памяти. Например, на основе сосательного рефлекса у младенца формируется навык питья из чашки, на основе ладонного рефлекса – навык хватания предметов, и т.д.

Перцепция (восприятие целостного образа) тоже развивается во взаимодействии сенсорики и моторики на базе генетически заданных механизмов. У новорожденного существует генетически заданная реакция на цвет (он отличает красное от черного и т.д.), на форму (отличает круг от квадрата и т.д.), на движение. Показано, что такие же генетические механизмы имеют место и у животных (в частности, они избирательно реагируют на прямые линии, на границу и т.д.).

У ребенка генетически задан общий механизм сличения объектов окружающей среды: новорожденный предпочитает движущиеся фигуры – неподвижным, центрированные комплексы – диффузным, сложные комплексы – простым,

объемные – плоскостным, и т.д. Имеется также генетически заданный путь развития речи, чего нет у животных: младенец имеет первичную голосовую реакцию, т.е. существует определенная структура его крика, сходная с интонациями повествовательной речи взрослого.

У животных психические функции даны им при рождении и развертываются в соответствии с генетически заданной программой, а у человека они только заданы, то есть не проявятся без определенных социальных условий и раньше определенного времени. При этом развертка генетически заданных функций у младенца происходит очень быстро (до 1 месяца), а дальше требуется взаимодействие со взрослым как социализирующий момент.

Механизмы классификации и структурирования у человека базируются на той картине мира, которая существует у него на текущий момент развития, т.е. новую информацию он воспринимает соответственно тем категориям образов, которые постепенно устанавливаются после рождения.

В динамике развития психики человека вначале формируется образная система, а потом понятийно-логическая. Психология достоверно показала, что при восприятии новой информации у человека сначала возникает единый образ – гештальт, а затем происходит его детализация. Базой служат врожденные общие алгоритмы, относящиеся ко всем модальностям и основанные на принципах равновесия и простоты (например, квадрат и круг – эталоны простых форм):

- разделение образа на фигуру и фон. Мозг имеет тенденцию структурировать сигналы таким образом, что все, что меньше, имеет более правильную конфигурацию или имеет для человека какой-то смысл, воспринимается как фигура, а все остальное – как гораздо менее структурированный фон;
- заполнение пробелов. Мозг старается свести фрагментарное изображение в фигуру с простым и полным контуром. Например, отдельные точки, расположенные по контуру креста, воспринимаются как цельный крест;
- группировка элементов по разным признакам (близости, сходства, единого направления). Продолжение беседы в общем шуме голосов возможно только потому, что мы слышим слова, произносимые одним голосом и тоном.

На основе гештальтов человек строит стратегии классификации, которые тесно связаны с индивидуальным жизненным опытом.

Таким образом, механизмы классификации и структурирования, реализуемые психикой, кардинально отличаются от подхода, принятого в науке, где в первую очередь за счет сходства и различия формируется общая система понятий, а затем она используется для логического вывода.

Более привычные сигналы распознаются автоматически, почти тотчас же (это феномен «обыденных сведений» или «здравого смысла», на котором споткнулись ранние ИИ системы). В других случаях, когда информация новая, неполная или неоднозначная, наш мозг действует путем выдвижения гипотез, которые он одну за другой проверяет, чтобы принять ту, которая кажется ему наиболее правдоподобной или наиболее приемлемой.

Нейрофизиология памяти и сознания. Запоминание (т.е. формирование долговременной памяти) имеет молекулярные механизмы, и многие из них

связаны с процессами, происходящими не между клетками, а внутри клетки, когда сигнал передается от мембраны геному. Формирование памяти проходит как бы две фазы: синтеза белка и экспрессии генов. На первой стадии, сразу после обучения (на стадии кратковременной памяти) активируются так называемые ранние гены. Вслед за этим идет вторая волна активации – после действия продуктов ранних генов на геном (на так называемые поздние гены). Иначе говоря, клетка перестраивает программу своей работы под влиянием ситуации обучения. В результате память человека выражается в проводимости конкретных нейронных синапсов.

Интересно, что этот процесс: (1) имеет место в отдельных нейронах, которые можно идентифицировать; (2) не является одномоментным, а продолжается в течение 5–20 минут; если в это время человеку дается новая задача, которую он должен запомнить, то эта новая задача мешает запоминанию старой информации; (3) идентичен при развитии мозга в детском состоянии и при обучении взрослых.

Таким образом, каждый акт познания (т.е. каждое решение отдельной задачи) для человека – это маленький эпизод морфогенеза и следующего развития. Это, а не скорость операций, обеспечивает мозгу способность к генерации новых решений в динамически меняющейся среде.

Сформированная (консолидированная) память стабильно хранится в сетях дифференцировавшихся нейронов (до сотни миллионов нейронов). Но в момент извлечения старой памяти активируются молекулярные механизмы, похожие на те, которые активируются в момент запоминания, т.е. происходит своего рода перезапись.

Большой экспериментальный материал показывает, что емкость и длительность долговременной памяти в принципе безграничны. Однако большей ее частью человек в обычных для себя условиях воспользоваться не может, так как не имеет к ней доступа. В памяти человека механизма произвольной адресации нет, а основой адресации является контекст. Другими словами, информация всегда воспроизводится на основе той структуры, в составе которой она запоминалась.

Таким образом, память – только одна из характеристик работы больших систем нервных клеток, это искусственно выделенный аспект работы мозга. Нет такого «куска» мозга, который бы занимался только памятью (как жесткий диск в компьютере). То же самое касается сознания. Сознание – это не след, а процесс, который современная нейробиология также может визуализировать с высокой детальностью – до уровня отслеживания активности отдельной нервной клетки во время отдельного поведенческого акта.

С позиций психологии мышление – это психологический процесс с открытием нового (возможно, субъективно, то есть только для мыслящего) знания и решение проблем на основе переработки полученной информации.

Современная психология сформировала модель мышления, основанную на асимметрии полушарий мозга. В процессе мышления участвуют оба полушария. В каждый момент времени обработка информации происходит только в одном полушарии. При доминировании левого полушария у человека результаты его

деятельности могут быть выражены вербально (в словах) и осознаны. При доминировании правого полушария результаты не вербализуются и не осознаются.

Переходы из левого в правое полушарие и обратно происходят скачком, при этом только при переходе из правого (образного) в левое (вербальное) возникает ощущение внезапности полученного решения. В этом случае мы считаем, что решение найдено интуитивно или путем «инсайта». Другими словами, механизм интуиции связан с накоплением и обработкой информации в одном полушарии до момента достижения некоторого порога, при котором полуфабрикат решения скачком передается в другое полушарие для реализации, завершения или осознания.

1.2. Интеллект и когнитивные функции

Большинство психологов определяют интеллект как способность индивидуума адаптироваться к окружающей среде. Общество, в котором мы живем, придает главное значение абстрактному мышлению, индивидуализму, духу конкуренции, учебным и профессиональным успехам. Тесты, по которым оценивается уровень интеллектуального развития (IQ), отражают адаптируемость человека к обществу именно в смысле этих ценностей. В обществе с иными представлениями понятие о нормальном интеллекте было бы, очевидно, существенно иным.

Большинство исследований свидетельствуют о иерархическом характере интеллекта. Так, [Cattell, 1963] выделяет свободный интеллект, который определяется общим уровнем развития коры больших полушарий, то есть успешностью решения задач на восприятие и нахождение отношений элементов, и связанный интеллект, который приобретается в процессе овладения культурой (вербальный и арифметический факторы, а также другие тесты, требующие обученности). В модели [Carroll, 1993] на вершине иерархии расположен общий уровень интеллекта, под ним, на промежуточном уровне – восемь широких способностей, каждая из которых, в свою очередь, включает ряд узких способностей.

Понятие когнитивных функций [Лурия, 1973; Lezak, 1983] введено в нейропсихологии для описания процессов переработки информации у здоровых и больных людей.

Под когнитивными функциями принято понимать наиболее сложные функции головного мозга, с помощью которых осуществляется процесс рационального познания мира [Lezak, 1983]. К когнитивным функциям относятся память, гнозис, речь, праксис и интеллект. Нейропсихологи придают им следующую трактовку:

- Память – это способность головного мозга усваивать, сохранять и воспроизводить необходимую для текущей деятельности информацию. Выраженные нарушения памяти на события жизни обозначают термином «амнезия».
- Гнозисом называется функция восприятия информации, её обработки и синтеза элементарных сенсорных ощущений в целостные образы.
- Речь – это способность обмениваться информацией с помощью высказываний. Нарушения речи (афазии) чаще всего развиваются при патологии лобных или височно-теменных отделов головного мозга.

– Праксис – это способность приобретать, сохранять и использовать разнообразные двигательные навыки. Патология лобных долей приводит к нарушению способности построения двигательной программы, а патология теменных долей – к неправильному использованию своего тела в процессе двигательного акта при сохранной программе движений.
– Под интеллектом понимают способность сопоставлять информацию, находить общее и различия, выносить суждения и умозаключения. Интеллектуальные способности обеспечиваются интегрированной деятельностью головного мозга в целом.

Когнитивистика (когнитивная наука) – междисциплинарное научное направление, объединяющее теорию познания, когнитивную психологию, нейрофизиологию, когнитивную лингвистику и теорию искусственного интеллекта [Kiely, 2014]. В когнитивистике совместно используются компьютерные модели, взятые из теории ИИ, и экспериментальные методы, взятые из психологии и физиологии высшей нервной деятельности, для разработки точных теорий работы человеческого мозга.

Вначале ученые-когнитивисты уподобляли соотношение между мозгом и познанием взаимосвязи между компьютерным оборудованием и программным обеспечением. При этом существовали два конкурирующих объяснения познания – либо как процессы манипулирования символами (когнитивизм), либо как нейронная сеть тормозящих и возбуждающих связей (коннекционизм).

Посткогнитивистские теории используют все современные результаты психологии и нейрофизиологии, кратко описанные выше.

Высшие когнитивные функции в целом отвечают за планирование, реализацию, координацию и контроль целенаправленного поведения. Когнитивное функционирование описывается как взаимодействие между одновременно происходящими процессами «сверху вниз» и «снизу вверх». Нисходящие процессы управляются абстрактными концепциями и схемами более высокого уровня. Они относятся к той роли, которую знания и ожидания, сформированные предыдущим опытом, играют в обработке информации. И наоборот, восходящие процессы отражают роль конкретных сенсорных входов более низкого уровня в управлении познанием. Кроме того, учитывается многомерный и иерархический характер когнитивных конструкций (в том числе многослойная модель интеллекта).

Другим важным примером иерархической и многомерной природы познания являются исполнительные функции. Исполнительные функции связаны с выполнением рутинных повседневных действий – взять ложку, вымыть посуду, включить кофеварку – и вследствие этого казались первым исследователям ИИ достаточно простыми в моделировании. Однако современная когнитивистика [Salthouse, 2005] рассматривает их как набор сложных процессов более высокого порядка, отвечающих за планирование, реализацию, координацию и контроль целенаправленного поведения. Исполнительные функции включают специфические когнитивные процессы, наиболее тесно связанные с функционированием лобных долей, такие как торможение, рабочая память и внимание. В частности,

исполнительные функции важны для суждений, принятия решений, решения проблем и оценки ситуации.

Наблюдаемый сейчас прогресс в когнитивистике направлен на то, чтобы описать и объяснить процессы в мозгу человека, ответственные за высшую нервную деятельность. Это позволит создать системы сильного ИИ, который будет обладать способностями к самостоятельному обучению, творчеству, свободному общению с человеком.

В завершение этого раздела нельзя не сказать о развитии ИИ и когнитивных наук в нашей стране. Основателем советской школы ИИ считается Дмитрий Александрович Поспелов (1932–2019). Его собственные выдающиеся научные результаты на стыке разных дисциплин – это подходы ситуационного управления [Поспелов, 1986], прикладной семиотики и псевдофизических логик, хорошо разработанные сейчас (некоторые из них представлены в настоящем пособии), а также базовые идеи новых когнитивных наук, включая когнитивную семантику на нестандартных биполярных шкалах, модели рассуждений «здравого смысла», триаду «объяснение-обоснование-оправдание» и её связи с выделенными им уровнями понимания.

Еще в 1982 г. он отметил, что исследования в ИИ должны быть нацелены на «изучение психики человека с целью ее имитации в технических системах, решающих определенный набор практических задач, традиционно считающихся интеллектуальными». Лишь спустя 25 лет, в самом конце первого десятилетия XXI века за рубежом появилось понятие «общий искусственный интеллект», в рамках которого стали развиваться более скромные идеи компьютерного моделирования целостного интеллекта как открытой системы, связанные с исследованием сознания. Наконец, в 1981 г. он выделил главный пробел в существующих возможностях создания ИИ – отсутствие математического аппарата для моделирования динамики формирования структуризированных представлений [Наппельбаум, Поспелов, 1981].

1.3. Соотношение знаний и данных в обработке информации

Д.А. Поспелов еще в 1986 г. предложил простую и практически применимую модель процесса обработки информации у человека [Поспелов, 1986], которая не потеряла своей актуальности до сих пор, а, напротив, подтверждается новыми достижениями когнитивистики.

Рассмотрим пример: человек увидел что-то и воспринял «это» как корову. Весь информационный комплекс, который возникает у человека в этой ситуации, оказывается логически непротиворечивым. В то же время, если все его знания о коровах, соответствующие разным ситуациям – корова на лугу, корова – детская игрушка, корова Ио из древнегреческого мифа и т.д. – совместить в рамках одной конкретной ситуации, то получится логически противоречивая система. Тем не менее, человек спокойно живет в мире этого «абсурда». Значит, эта информация так организована в его сознании, чтобы избежать возможных логических противоречий.

Одна из возможностей такой организации информации показана на рис. 1.2. Каждая вершина сети представляет собою некоторую замкнутую логическую систему (знаковую систему, мир). В ней хранится логически непротиворечивая информация, совместимая в рамках одной ситуации. Принципиально такую логическую систему можно формализовать и воспроизвести техническими средствами (например, нарисовать корову на бумаге). Но возможны и другие миры (M_{i+1}, M_{i+2}, …), в которых действуют другие аксиомы и семантические правила, а также своя интерпретация.

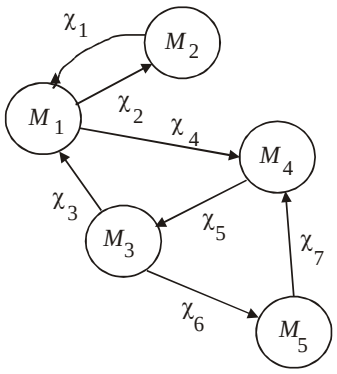


Рис. 1.2. Схема преобразования логических систем

Связи χ_j между вершинами задают переходы от одной логической системы к другой. Некоторые из таких связей могут быть формализованы и, следовательно, объективизированы (т.е. выведены в сферу материального). Тогда соответствующие переходы $M_i \rightarrow M_{i+1}$ могут быть реализованы техническими средствами без участия человека, а весь граф (рис. 1.2) превращается в семантическую сеть. Например, можно заснять корову, пасущуюся на лугу, на пленку, а затем печатать отдельные кадры на принтере.

Другие связи реализуются через обращение к смыслам (семантическим инвариантам) объектов, входящих в конкретную знаковую систему, и они (по крайней мере, на сегодняшнем уровне развития науки) полной формализации не поддаются. Например, человек увидел корову на лугу и принял решение, какую игрушку купить ребенку.

Возможны также слабо формализуемые, в частности, контекстно-зависимые связи. Это – сфера активных исследований.

Таким образом, согласно Д.С. Поспелову:

- данные есть комплекс информации, совместимый в рамках некоторой формальной системы с учетом всех возможных интерпретаций этой системы.
- знания есть информация, которая хранится во всех возможных мирах, вместе с условиями перехода от одного мира к другому.

Заметим, что в фундаментальном учебнике по ИИ [Russell, Norvig, 2020] знания определяются, по существу, так же:

Знания = факты + доверия + эвристики,

где эвристика понимается как набор правил по отображению интуитивной теории в формальную (логическую, математическую) теорию.

Вопросы для самопроверки

1. Назовите основные нейрофизиологические характеристики мозга как системы переработки информации.
2. Как развиваются познавательные функции человека в процессе его взросления**?** Приведите примеры.
3. На чем основаны механизмы классификации и структурирования у человека?
4. Какой механизм формирования долговременной памяти реализуется в мозге?
5. Какова роль асимметрии полушарий в процессе мышления?
6. Какие свойства интеллекта отражают тесты интеллектуального развития (IQ)?
7. Перечислите когнитивные функции, выделяемые в нейропсихологии.
8. Опишите модель процесса обработки информации у человека, предложенную Д.А. Поспеловым.

2. СТАДИИ РАЗВИТИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Как уже говорилось выше, ответы на основополагающие вопросы об интеллекте менялись по мере развития науки в целом, и соответственно изменялись подходы к трактовке и реализации систем ИИ. Поэтому ИИ прошел в своем развитии несколько стадий, причем каждая из них дала результаты, которые остаются актуальными и по сей день.

Разные источники по-разному подходят к обозначению стадий развития ИИ. Учебник [Russell, Norvig, 2020] предлагает следующее структурирование:

- зарождение искусственного интеллекта (1943–1956 гг.);
- ранний энтузиазм, большие надежды (1952–1969 гг.);
- доза реальности (1966–1973 гг.);
- экспертные системы (1969–1986 гг.);
- возвращение нейронных сетей (с 1986 г. по настоящее время);
- вероятностные рассуждения и машинное обучение (с 1987 г. по настоящее время);
- большие данные (с 2001 г. по настоящее время);
- глубокое обучение (с 2011 г. по настоящее время).

В Университете ИТМО предложена другая трактовка:

- 1960–1980 гг. – фрагментарные удачные решения ИИ для избранных прикладных примеров;
- 1980–1990 гг. – массовые удачные решения ИИ, обучаемые человеком (экспертные системы и пр.);
- 1990–2000 гг. – массовые удачные решения ИИ, обучаемые компьютером (машинное обучение);
- 2000–2010 гг. – унификация решений ИИ для разных предметных областей;
- с 2010 г. – агломерация решений ИИ для разных предметных областей в единое киберпространство.

Объединяя разнородные источники, можно предложить укрупненную схему стадий развития ИИ, представленную на рис. 2.1. В дальнейшем изложении будем ориентироваться на этот рисунок, не забывая о его схематичности и переналожении отдельных решений в разных стадиях.

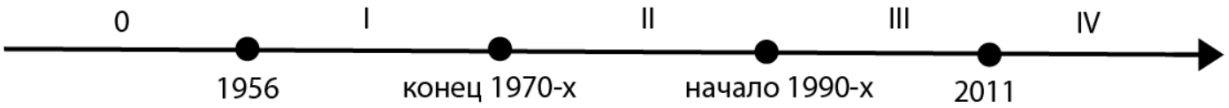


Рис. 2.1. Стадии развития ИИ: 0 – предшественники, I – логика и символы, II – знания, III – интеллектуальные агенты, IV - глубокое обучение, большие данные, когнитивистика, сильный ИИ

2.1. Стадия 0 – Предшественники

Классические философы, начиная с Аристотеля, пытались описать процесс человеческого мышления как механическое манипулирование символами. В

XVII веке Лейбниц, Томас Гоббс и Рене Декарт считали, что всякое рациональное мышление может быть сделано таким же систематическим, как алгебра или геометрия. Известно высказывание Гоббса: «Reason is nothing but reckoning (Разум есть не что иное, как расчет)». Дэвид Гильберт призвал математиков 1920-х и 30-х гг. дать, наконец, строгий ответ на фундаментальный вопрос: «Можно ли формализовать все математические рассуждения?» Однако ответ на этот вопрос, представленный в таких фундаментальных результатах, как доказательство неполноты Гёделя, машина Тьюринга и лямбда-исчисление Черча, оказался неожиданно отрицательным.

Эти результаты доказали, что пределы возможностей математической логики существуют, но, что более важно для развития ИИ, в этих пределах любая форма математического рассуждения может быть механизирована. Тезис Черча-Тьюринга подразумевал, что механическое устройство, перебирающее нули и единицы, может имитировать любой мыслимый процесс математической дедукции. Машина Тьюринга – простая теоретическая конструкция, манипулирующая всего двумя символами – легла в основу первых современных компьютеров.

С точки зрения ИТ главное свойство биологического нейрона (рис. 1.1) состоит в следующем: когда суммарный сигнал, приходящий от других нейронов, превышает некоторое пороговое значение, генерируется стандартный импульс; в противном случае нейрон остается в состоянии покоя. На основе этого свойства в 1943 г. была предложена модель формального нейрона в виде «адаптивный сумматор + нелинейный преобразователь с функцией активации» (рис. 2.2). Вскоре было показано, что комбинации таких нейронов могут выполнять простые логические функции, т.е. были построены первые нейронные сети.

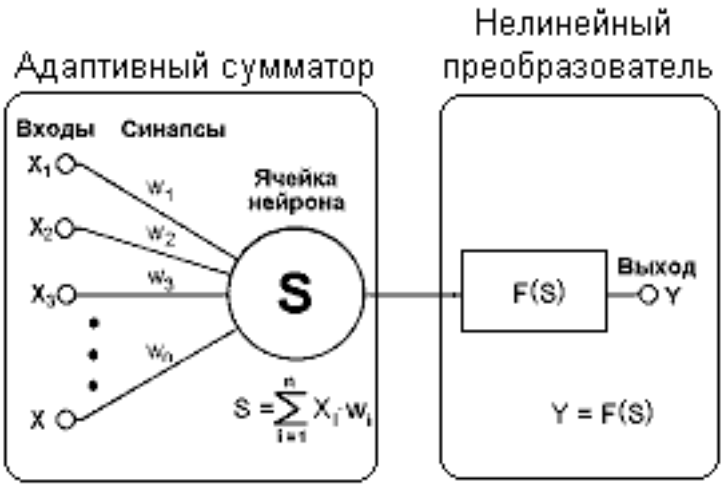


Рис. 2.2. Модель формального нейрона

На основе сочетания этих идей ученые из самых разных областей (математики, психологии, инженерии, экономики и политологии) начали обсуждать возможность создания искусственного мозга, и в 1956 г. область исследований ИИ была заявлена как академическая дисциплина.

2.2. Стадия I – Логика и символы

Большинство ранних программы ИИ использовали один и тот же базовый алгоритм. Для достижения какой-либо цели (например, победы в игре или доказательства теоремы) они продвигались к ней шаг за шагом (совершая ход или дедукцию), словно блуждая по лабиринту. Эта парадигма получила название «рассуждение как поиск (reasoning as search)».

Характерным примером служит планировщик STRIPS (STanford Research Institute Problem Solver) – первая система, которая создавалась именно для решения задачи планирования [Fikes, Nilsson, 1971]. Задача планировщика – найти такую последовательность действий, которая преобразует начальное состояние в такое состояние, в котором достигается заранее заданное целевое условие (в оригинальной статье используются понятия «модель мира» для обозначения состояния и «оператор» для обозначения действия, что вполне созвучно с терминологией модели Д.А. Поспелова – см. рис. 1.2).

Постановка задачи в STRIPS включает три составляющие: начальное состояние, множество действий и целевую формулу. Все они описываются множеством логических выражений, при этом утверждения, не входящие в число этих формул и не являющиеся их логическим следствием, считаются ложными. Это допущение в литературе называют допущением о замкнутости мира (closed-world assumption). Для поиска плана действий использована стратегия, которая определяет различие между текущим состоянием и целью и находит действия, способные уменьшить это различие.

STRIPS подвергался обширной критике как за подход к описанию мира и действий, так и за алгоритм поиска. Во-первых, STRIPS-подобные системы имеют проблемы со строгостью логического описания. Во-вторых, выявилась проблема зависимости подцелей в составной цели. STRIPS-подобные системы разрабатывались исходя из предположения, что подцели всегда можно достигать последовательно: сначала одну подцель, потом, уже из нового состояния – другую, и т.д. Но они не учитывали того, что может потребоваться чередование шагов, служащих для достижения разных подцелей. Утверждение о том, что подцели можно достигать последовательно, называется допущением линейности и верно только в случае независимости подцелей. Наконец, STRIPS-эвристики не спасают от комбинаторного взрыва.

Тем не менее, задача планирования сформулирована в [Fikes, Nilsson, 1971] так, какова она и по сей день. Многие идеи и решения, использованные в STRIPS, и по сей день остаются основополагающими в планировании. Позже были предложены решения для устранения этих недостатков.

Достаточно плодотворный подход к использованию логических выводов в ИИ предоставляет успешный язык логического программирования Пролог (1970-е гг.). Пролог использует подмножество логических выражений (дизъюнкты Хорна и продукционные правила), которые позволяют выполнять поддающиеся обработке вычисления.

Приведем еще несколько примеров ИИ решений этого времени.

Для обработки естественного языка были предложены решения на основе правил и на основе семантических сетей. Например, программа STUDENT [Bobrow, 1964] решала задачи по алгебре из школьного учебника, выраженные

на естественном языке. STUDENT использовала систему, основанную на правилах, с логическим выводом (inference). Сначала предложения на английском языке преобразуются в ядерные предложения, каждое из которых содержит один фрагмент информации. Затем предложения ядра преобразуются в математические выражения. База фактов, поддерживающая преобразование, содержала 52 факта.

Семантическая сеть (рис. 2.3) представляла понятия как узлы, а отношения между понятиями – как связи между узлами. Более продвинутые модели семантических сетей вводили некоторую концептуализацию, например, выделяли объекты реального мира со своими атрибутами, действия в реальном мире со своими атрибутами, время и местоположение и т.п.

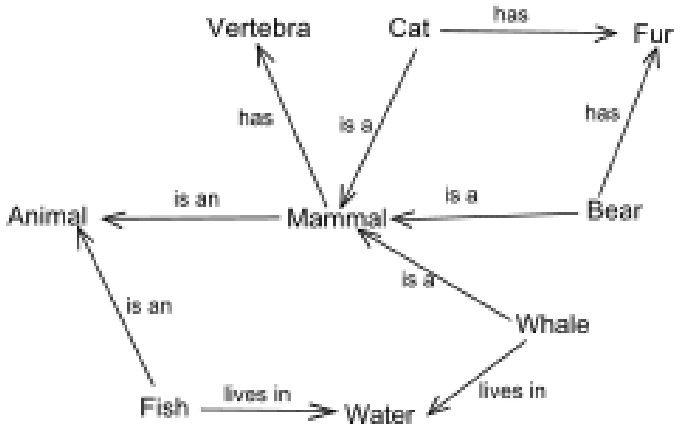


Рис. 2.3. Пример семантической сети

Одной из самых успешных систем раннего ИИ стал робот SHRDLU [Winograd, 1968]. По команде пользователя SHRDLU мог перемещать различные объекты в «мире блоков», содержащем конусы, шары и т. д. Для убедительной имитации «понимания» в SHRDLU использовались весьма простые идеи:

- Концепция «малого мира» (small world). Весь набор объектов и локаций SHRDLU можно было описать, включив всего 50 слов: существительные, такие как «блок» и «конус», глаголы, такие как «поместить на» и «переместиться», и прилагательные, как «большой» и «синий». Возможные комбинации этих основных строительных блоков языка были довольно просты.
- Базовая память для предоставления контекста. Можно было бы попросить SHRDLU «положить зеленый конус на красный блок», а затем «снять конус»; при этом под «конусом» SHRDLU понимал зеленый конус, о котором только что говорили.

В результате взаимодействия памяти и первоначальных правил, которыми был снабжен SHRDLU, он мог отвечать на вопросы о том, что возможно в мире, а что нет. Например, SHRDLU на примерах «понимал», что блоки можно складывать в стопку, а треугольники нельзя, потому что «мир» содержал базовую физику, заставляющую блоки падать. Он также мог запоминать имена, данные объектам, или их расположение.

Однако оптимизм исследователей ИИ вскоре был утрачен, когда более поздние системы попытались работать в ситуациях с более реалистичным уровнем двусмысленности и сложности. Выявились фундаментальные ограничения, которые не могли быть преодолены в 1970-х гг.:

- Ограниченная мощность компьютера – не хватало памяти или скорости обработки на реальные приложения, такие как техническое зрение или обработка естественного языка;
- Неразрешимость и комбинаторный взрыв – эта проблема была математически сформулирована в 1971–1972 гг. Как ответ на нее был разработан язык Пролог, основанный на дизъюнктах Хорна и гипотезе замкнутого мира, который позволяет выполнять поддающиеся обработке вычисления;
- Логика первого порядка не может представить обычные выводы, включающие планирование или рассуждения по умолчанию, без внесения изменений в структуру самой логики. В ответ были разработаны другие логики (например, немонотонная и модальная), которые позволяют в определенной степени обойти эти проблемы;
- Люди не используют классические логические рассуждения в повседневной жизни, например, при понимании историй (story telling) и распознавании объектов.

Таким образом, невероятный оптимизм первых лет породил нереалистично высокие ожидания, и, несмотря на разумные и перспективные идеи и первые успехи в области ИИ, в 1974–1980 гг. наступила «первая зима ИИ».

2.3. Стадия II – Знания

Понятие знаний имеет разнообразные трактовки, некоторые из них были представлены в разделе 1.3. Современные стандарты определяют их следующим образом:

- знания (в искусственном интеллекте) (knowledge): Совокупность фактов, событий, убеждений, а также правил, организованных для систематического применения [ГОСТ 33707–2016, статья 4.398];
- модель знаний (knowledge model): Информационная модель, которая выражает знания в структуре, интерпретируемой компьютером [ГОСТ Р 57309–2016, статья 3.1.1].

Как уже было сказано, люди не используют классические логические рассуждения в повседневной жизни. Чтобы использовать обычные понятия, такие как «стул» или «ресторан», ИИ должен делать все те же нелогичные предположения, которые обычно делают люди. И хотя весь ход развития западноевропейской науки был основан на логике как главной ценности, исследователи ИИ вынуждены были признать, что интеллект вполне может быть основан на способности использовать большие объемы разнообразных знаний различными способами. (Обратим внимание, что понятие знаний здесь использовано как раз в смысле Д.А. Поспелова).

В 1975 г. была предложена [Minski, 1975] структура представления фактов, которая охватывает предположения здравого смысла о чем-либо. Например, если

мы используем понятие птицы, сразу приходит на ум совокупность фактов: она летает, она питается червями, и т.д. Хотя эти факты не всегда верны и выводы, основанные на этих фактах, не всегда логичны, но эти структурированные наборы предположений являются частью контекста всего, что мы говорим и думаем. Эти структуры были названы фреймами.

Системы, основанные на знаниях (экспертные системы), и инженерия знаний стали основным направлением исследований ИИ в 1980-х гг.

Экспертная система – это программа, которая отвечает на вопросы или решает проблемы в определенной области знаний, используя логические правила, полученные из знаний экспертов. Одной из первых успешных экспертных систем была программа MYCIN [Shortliffe, Buchanan,1975]. Она позволяла выявлять бактерии, вызывающие тяжелые инфекции, такие как бактериемия и менингит, и рекомендовать антибиотики в дозировке, адаптированной к массе тела пациента. Программа была написана на языке LISP, содержала базу из 600 правил и весьма простую машину вывода. Для описания неопределенности знаний разработчики MYCIN впервые использовали вместо классической байесовской статистики так называемые «факторы уверенности», что сделало систему гораздо более компактной и удобной для использования.

Экспертные системы работали в небольшой области конкретных знаний (тем самым избегая проблемы «знаний здравого смысла»), а их простая архитектура позволяла относительно легко создавать и модифицировать программы. Кроме того, в них использовались парадигмы, позволяющие отойти от проблемы замкнутости мира, т.е. не требующие строгой логики предикатов для своей реализации – фреймы, сценарии, продукционные системы. Все это активно используется и сейчас.

Подобные программы оказались полезными и вошли в моду, тем самым был достигнут первый коммерческий успех ИИ. Корпорации по всему миру начали разрабатывать и развертывать экспертные системы, и к 1985 г. они потратили более миллиарда долларов на ИИ, причем большую часть – на собственные отделы ИИ.

В России (тогда СССР) развивались работы Д.А. Поспелова и его сотрудников, выполненные еще в 1960–1970-е гг., которые на полтора десятка лет опередили западные аналоги. Они предложили комплекс новых методов построения систем управления, в основе которого лежит идея разработки семиотических (логико-лингвистических) моделей представления объекта управления и описания процедур управления ими. В СССР с помощью этих методов были построены эффективные системы оперативного диспетчерского управления такими сложными объектами, как грузовой морской порт, атомная электростанция, комплекс трубопроводов, и пр.

Однако к концу 1980 гг. в мире наступила «вторая зима ИИ». Первым признаком стал внезапный обвал рынка специализированного оборудования для ИИ в 1987 г. Настольные компьютеры Apple и IBM, неуклонно набирая скорость и мощность, в 1987 г. превзошли по производительности более дорогие машины на LISP. Кроме того, ранние экспертные системы оказались слишком дорогими в обслуживании. Их было трудно обновлять, они не могли учиться, они могли

делать гротескные ошибки при вводе необычных данных. В целом экспертные системы оказались полезными только в нескольких особых случаях.

2.3. Стадия III – Интеллектуальные агенты

Однако отрасль продолжала развиваться, несмотря на критику. Многие исследователи выступали за совершенно новый подход к ИИ – за переход к интеллектуальным агентам.

Теоретическая парадигма для появления интеллектуальных агентов в ИИ состояла в следующем: для проявления настоящего интеллекта машине необходимо иметь тело – ей нужно воспринимать, двигаться, выживать и взаимодействовать с миром. Эти сенсомоторные навыки необходимы для навыков более высокого уровня, таких как здравый смысл, а абстрактное мышление на самом деле – наименее интересный или важный человеческий навык, т.е. интеллект строится «снизу вверх». Этот подход возродил идеи кибернетики и теории управления, которые были не популярны с 60-х гг.

С другой стороны, в компьютерной науке понятие интеллектуального агента существовало давно – это программа, самостоятельно выполняющая задание, указанное пользователем компьютера, в течение длительных промежутков времени. Примерами служат компьютерные вирусы, боты, поисковые роботы. Такие агенты могут иметь сложный, зачастую реализуемый нейросетями алгоритм, как, к примеру, у поисковой системы Google (при поиске по видео). «Интеллектуальность» в этом контексте понимается как возможность обратной связи в соответствии, например, с результатами анализа поисковых запросов и их выдачей. В операционных системах семейства UNIX интеллектуальный агент, действующий в пределах одного компьютера или локальной сети, обычно называется демоном, в семействе Windows – службой (сервисом).

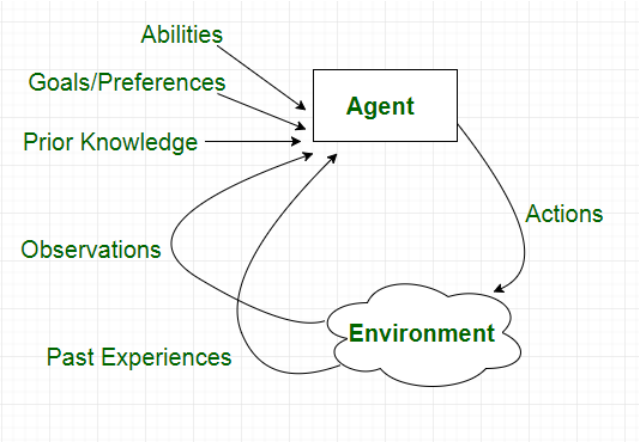


Рис. 2.4. Структура интеллектуального агента

В ИИ под термином «интеллектуальный агент» понимаются сущности, получающие информацию через систему сенсоров о состоянии управляемых ими процессов и осуществляющие влияние на них через систему актуаторов, при этом их реакция рациональна в том смысле, что процессы, выполняемые ими, содействуют достижению определённых параметров. Наиболее близким

аналогом в живой природе является инстинктивное поведение насекомых. В ИИ возможны разные классификации агентов, в том числе:

– физический агент – агент, воспринимающий окружающий мир через некоторые сенсоры и действующий с помощью манипуляторов;
– временной агент – агент, использующий изменяющуюся с ходом времени информацию, предлагающий некоторые действия или предоставляющий данные компьютерной программе или человеку и получающий информацию через программный ввод.

Кроме того, всех агентов можно разделить на пять групп по типу обработки воспринимаемой информации.

1. Агенты с простым поведением (reflex agents).

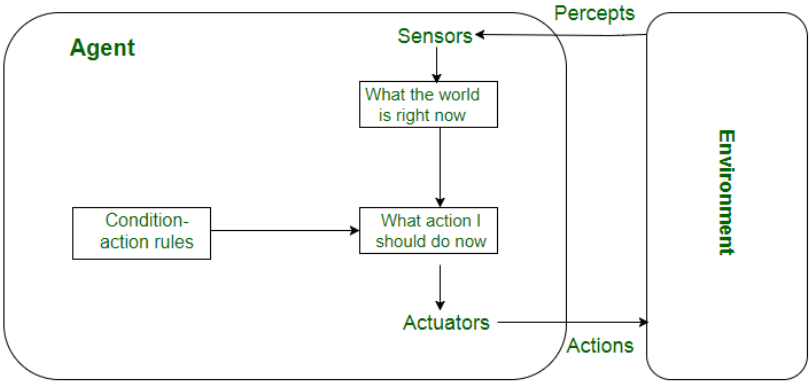


Рис. 2.5. Структура агента с простым поведением

Простые рефлекторные агенты действуют только на основе текущего восприятия, т.е. игнорируют всю историю восприятия. Работа агента основана на правиле «условие – действие» (if-then), которое отображает состояние, т.е. условие, в действие. Если условие истинно, то действие выполняется, иначе нет. Такой агент может успешно работать только тогда, когда среда полностью наблюдаема. Для простых рефлекторных агентов, работающих в частично наблюдаемой среде, часто неизбежны бесконечные циклы; однако их можно избежать, если агент сможет рандомизировать свои действия.

2. Агенты с поведением, основанным на модели (Model-based reflex agents).

Model-based reflex agent (рис. 2.6) находит правило, условие которого соответствует текущей ситуации, и реализует его в своем действии. Агент с поведением на основе модели может работать с частично наблюдаемой средой, используя модель мира. Агент должен отслеживать внутреннее состояние, которое регулируется каждым восприятием и зависит от истории восприятия. Текущее состояние хранится внутри агента, который поддерживает некую структуру, описывающую невидимую часть мира. Для обновления состояния требуется информация о том, как мир развивается независимо от агента и как действия агента влияют на мир.

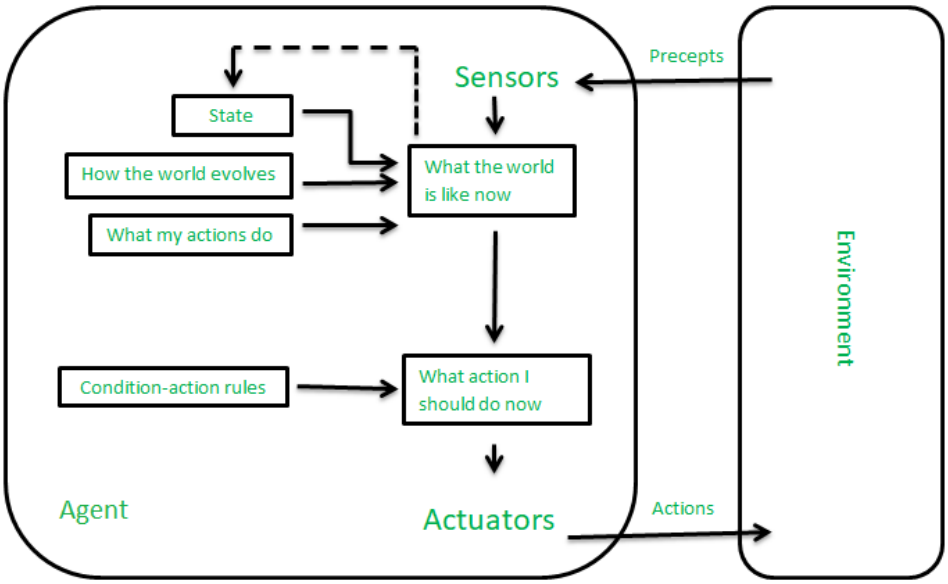


Рис. 2.6. Структура агента с поведением, основанным на модели

3. Целенаправленные агенты (Goal-based agents).

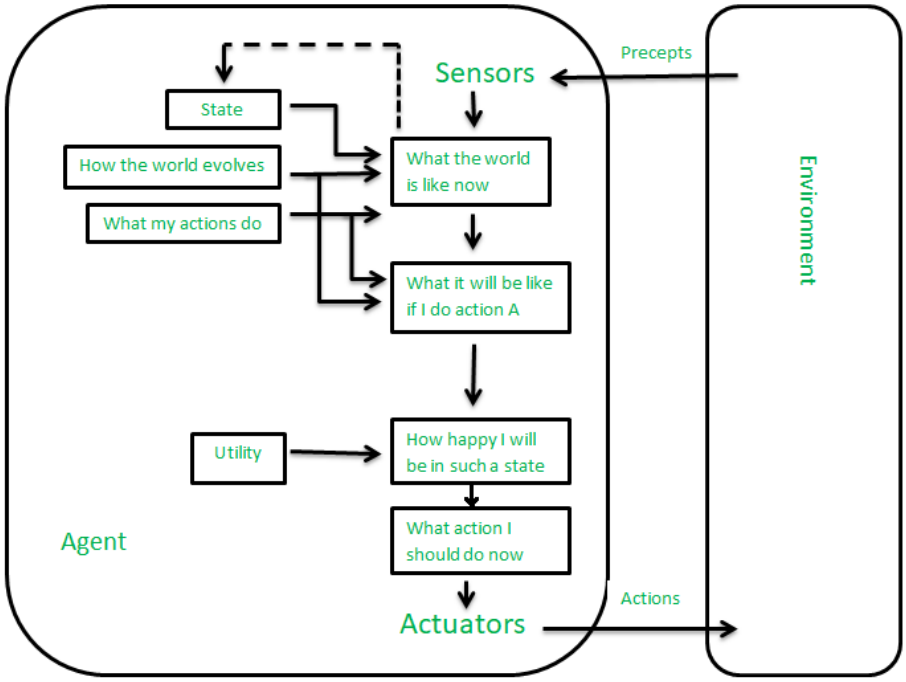


Рис. 2.7. Структура целенаправленного агента

Целенаправленные агенты (рис. 2.7) схожи с предыдущим типом, однако они, помимо прочего, хранят информацию о тех ситуациях, которые для них желательны. Это дает агенту возможность выбрать среди многих путей тот, что приведет к нужной цели. Эти виды агентов принимают решения, основываясь на том, насколько они в настоящее время далеки от своей цели (которая описана в их памяти). Каждое их действие направлено на сокращение своего расстояния от цели. Это позволяет агенту выбирать из множества возможностей ту, которая достигает целевого состояния. Знания, поддерживающие его решения,

представлены в явном виде и могут быть изменены, что делает этих агентов гибкими: поведение целевого агента можно легко изменить.

4. Практичные агенты (Utility-based agents).

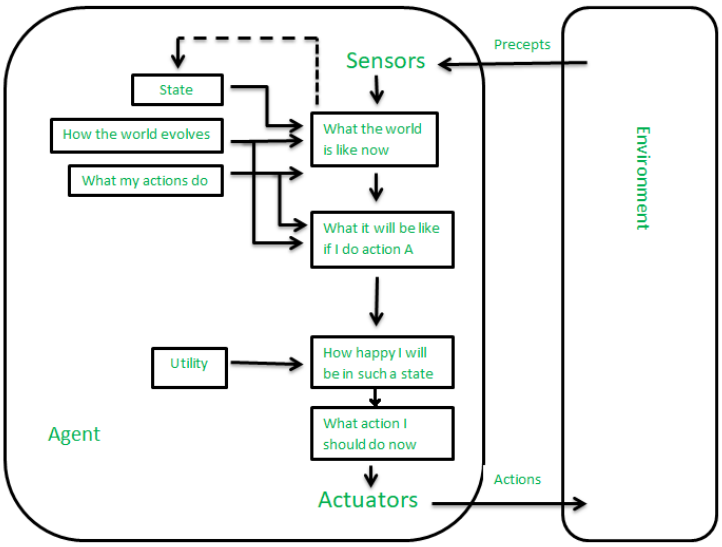


Рис. 2.8. Структура практичного агента

Если целенаправленные агенты различают только состояния, когда цель достигнута и когда не достигнута, то практичные агенты, помимо этого, способны различать, насколько желанно для них текущее состояние. Такая оценка может быть получена с помощью «функции полезности», которая проецирует множество состояний на множество мер полезности состояний. Когда существует несколько возможных альтернатив (рис. 2.8), практичные агенты выбирают действия на основе предпочтения (полезности) для каждого состояния.

Однако иногда достижения желаемой цели недостаточно. Например, чтобы добраться до пункта назначения, можно искать более быструю, безопасную и дешевую поездку. В теории мультиагентных систем даже используется терминология, близкая к человеческим оценкам, например, «воля», «счастье» и пр. В этих терминах полезность описывает, насколько «счастлив» агент. Из-за неопределенности в мире практичный агент выбирает действие, которое максимизирует ожидаемую полезность. Функция полезности отображает состояние в действительное число, которое описывает соответствующую степень счастья.

5. Обучающийся агент (Learning agent).

Обучающийся агент в ИИ – это тип агента, который может учиться на своем прошлом опыте или обладает способностями к обучению. В литературе обучающиеся агенты также называются автономными интеллектуальными агентами (autonomous intelligent agents), что подчеркивает их независимость и способность к обучению и приспособляемость к изменяющимся обстоятельствам.

Обучающийся агент содержит четыре концептуальных компонента (рис. 2.9). Элемент обучения отвечает за внесение улучшений, учась у окружающей среды. Критик описывает, насколько хорошо агент справляется с фиксированным стандартом деятельности (performance). Элемент производительности отвечает за выбор внешнего действия. Генератор проблем отвечает за предложение действий, которые приведут к новым и информативным впечатлениям.

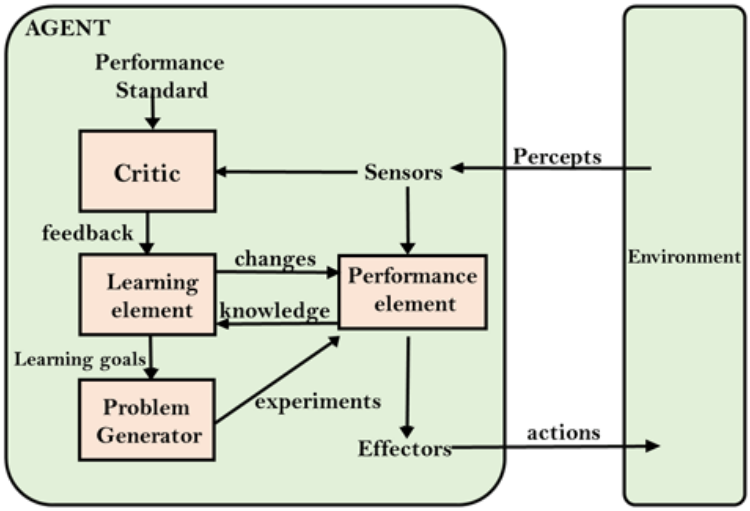


Рис. 2.9. Структура обучающегося агента

Из обучающихся агентов строятся многоагентные системы (МАС) [Wooldridge, 2002]. В таких системах агенты имеют несколько важных характеристик:

- автономность – агенты, хотя бы частично, независимы;
- ограниченность представления – ни у одного из агентов нет представления обо всей системе, или система слишком сложна, чтобы знание о ней имело практическое применение для агента;
- децентрализация – нет агентов, управляющих всей системой;
- агенты могут обмениваться полученными знаниями, используя некоторый специальный язык и подчиняясь установленным правилам «общения» (протоколам) в системе.

В МАС может проявляться самоорганизация и сложное поведение, даже если стратегия поведения каждого агента достаточно проста. Это обстоятельство лежит в основе так называемого роевого интеллекта. В связи с этим подходы к разработке МАС напоминают подходы к управлению в социальных системах. В частности, для программирования таких агентов предложена модель «убеждения, желания и намерения» (belief–desire–intention software model, BDI).

Агент BDI – это особый тип ограниченного рационального программного агента, наделенного определенными ментальными установками, а именно, убеждениями, желаниями и намерениями (BDI).

Убеждения агента представляют собой его информационное состояние, другими словами, его представления о мире (включая себя и других агентов). Убеждения также могут включать правила вывода, позволяющие посредством логических цепочек приводить к новым убеждениям. Использование термина «убеждение», а не «знание», предполагает, что представления агента не обязательно могут быть правдой (и фактически могут измениться в будущем). Убеждения хранятся в базе данных (иногда называемой базой убеждений).

Желания представляют собой мотивационное состояние агента. Они представляют цели или ситуации, которые агент хотел бы выполнить или вызвать, например, найти лучшую цену, пойти на вечеринку или стать богатым.

Цель – это желание, которое активно преследуется агентом. Использование термина «цель» добавляет дополнительное ограничение, заключающееся в том, что набор активных желаний должен быть последовательным. Например, у актора не должно быть одновременных целей пойти на вечеринку и остаться дома, даже если они оба могут быть желательны.

Намерения представляют совещательное состояние агента – то, что агент решил сделать. Намерения – это желания, которым агент в той или иной степени привержен. В реализованных системах это означает, что агент начал выполнение плана.

Планы – это последовательности действий, которые агент может выполнить для достижения одного или нескольких своих намерений. Планы могут включать в себя другие планы: например, план «поехать на машине» может включать в себя план «найти ключи от машины». Планы изначально формируются лишь частично, а детали дополняются по мере их выполнения.

События – это триггеры реактивной активности агента. Событие может обновить убеждения, запустить планы или изменить цели. События могут генерироваться извне и приниматься датчиками или интегрированными системами. Кроме того, события могут генерироваться внутри для запуска несвязанных обновлений или планов действий.

BDI также был дополнен компонентом обязательств, что привело к созданию архитектуры агента, которая включает обязательства, нормы и соглашения агентов, действующих в социальной среде.

Работа агента в МАС происходит в соответствии со следующим алгоритмом:

```
1. initialize-state
2. repeat
      1. options: option-generator (event-queue)
      2. selected-options: deliberate(options)
      3. update-intentions(selected-options)
      4. execute()
      5. get-new-external-events()
      6. drop-unsuccessful-attitudes()
      7. drop-impossible-attitudes()
3. end repeat
```

У модели BDI есть идеологические недостатки. В исходной теоретической модели частично рационального поведения выделяются три типа установок (attitudes), при этом и вера (belief), и желание (desire) – это про-установки (pro-attitudes), связанные с действием, а намерение (intention) выделяется как про-установка, контролирующая поведение. Для их описания пришлось разработать мультимодальную логику, но она не имеет полной аксиоматизации, т.е. не является эффективно вычислимой. В ней не заложено перспективное планирование или упреждающее обсуждение. Это плохо, так как принятые планы могут использовать ограниченные ресурсы, действия могут быть необратимыми,

выполнение задачи может занять больше времени, чем предполагалось при перспективном планировании, а действия могут иметь нежелательные побочные эффекты в случае неудачи.

Тем не менее, МАС и интеллектуальные агенты широко распространены – здесь можно назвать военные приложения, бытовые приложения (автоматизированные пылесосы), транспортные приложения и т.п.

Еще раньше предложения по созданию интеллектуальных агентов были высказаны и в СССР. Д.А. Поспелов в 1969 г. [Поспелов, 1969] предложил гиромат – по существу, интеллектуальный агент, имеющий «сознание» и «самосознание». В процессе своего функционирования гиромат строит в своей памяти модель окружающей среды и синтезирует программу действий в соответствии с заложенными в него целями, сообразуясь с этой моделью. Кроме того, в модели отражен сам гиромат, вся его структура и все известные из опыта функционирования в данной среде взаимоотношения между ним и средой. Это дает гиромату возможность анализировать не только мир, в котором он функционирует, но и свое функционирование в этом мире. «Сознанием» гиромата было названо его свойство отображать внешнюю среду в своей памяти и анализировать закономерности этой среды и результаты своих воздействий на среду, а «самосознанием» гиромата – свойство отображать себя в модели среды и анализировать закономерности воздействия среды на свою структуру и функционирование.

Пространство состояний гиромата описывается мультиграфом с n вершинами (понятиями), дуги которого раскрашены r различными цветами (типами связей). В каждый момент времени в гиромате в виде соответствующих векторов фиксируются текущие состояния среды и гиромата. В процессе накопления жизненного опыта гиромат с помощью модели кратковременных гипотез формирует некоторые устойчивые закономерности, обнаруживаемые им в окружающей среде и в своей структуре. Модели таких гипотез есть в сознании и самосознании гиромата.

В целом можно констатировать, что на рубеже XX и XXI веков произошло серьезное взаимопроникновение математики, ИИ и смежных наук. Исследователи ИИ начали разрабатывать и использовать сложные математические инструменты больше, чем когда-либо в прошлом. Пришло понимание того, что над многими проблемами ИИ уже работали исследователи в смежных областях, таких как математика, электротехника, экономика или исследование операций. Общий математический язык обеспечил как более высокий уровень сотрудничества с более устоявшимися и успешными областями знаний, так и достижение результатов, которые можно было измерить и доказать; ИИ стал более строгой «научной» дисциплиной. В ИИ стали использоваться байесовские сети, скрытые марковские модели, теория информации, стохастическое моделирование и классическая оптимизация. Были также разработаны точные математические описания парадигм «вычислительного интеллекта», таких как нейронные сети и эволюционные алгоритмы.

С другой стороны, появились сильные прикладные результаты: алгоритмы, изначально разработанные исследователями ИИ, стали появляться как части более крупных систем, например, в интеллектуальном анализе данных,

промышленной робототехнике, логистике, распознавании речи, банковском программном обеспечении, медицинской диагностике и поисковых системах.

Поразительно, но эти успехи в области ИИ в 1990-х и начале 2000-х гг. почти или вовсе не получили достойной оценки. Многие из величайших инноваций ИИ были сведены к статусу просто еще одного элемента в наборе инструментов информатики. Многие исследователи ИИ в 1990-х гг. сознательно называли область своей работы другими именами, такими как информатика, системы, основанные на знаниях, когнитивные системы или вычислительный интеллект. Несбывшиеся обещания «зимы ИИ» продолжали преследовать исследования ИИ вплоть до середины 2000-х годов.

2.4. Стадия IV – Глубокое обучение, большие данные, сильный ИИ

2.4.1. Глубокое обучение

Нейронные сети появились еще на заре ИИ. Первые публикации по многослойным персептронам относятся к началу 1960 гг. [Rosenblatt, 1961; Ивахненко, 1962]. Однако серьезный прикладной интерес к нейронным сетям связывается с прогрессом в области «железа», т.е. аппаратного обеспечения компьютерной техники.

Нейронные сети как классификаторы. Формальный нейрон (рис. 2.2) имеет n входов $x_1,\ldots,x_n$. В адаптивном сумматоре рассчитывается их взвешенная сумма,

$$s = \sum_{i=1}^{n} x_i \cdot w_i ,$$

подается на нелинейный преобразователь F, и на выходе нейрона получается

$$y = F(s).$$

Функция F нелинейного преобразователя называется активационной функцией нейрона. Исторически первой была модель, в которой в качестве активационной функции использовалась ступенчатая функция или функция единичного скачка:

$$F(s) = \begin{cases} 0, & s < 0, \\ 1, & s \geq 0. \end{cases}$$

По аналогии с биологическим нейроном, когда суммарное воздействие на входе превысит критическое значение, генерируется импульс 1, иначе нейрон остается в состоянии покоя, то есть выдается 0.

Существует множество других функций активации (рис. 2.10). Одной из наиболее распространенных является логистическая функция (сигмоида), которая в общем случае моделирует кривую роста вероятности f некоего события по мере изменения управляющих параметров x и α:

$$f(x) = \frac{1}{1+e^{-\alpha x}} .$$

Ценность сигмоиды как функции активации заключается в том, что она имеет простое выражение для производной:

$$f'(x) = \alpha \cdot f(x) \cdot (1 - f(x)),$$

что упрощает вычисление обратного распространения ошибки. Однако сейчас в качестве функции активации более популярна ReLU и ее модификации.

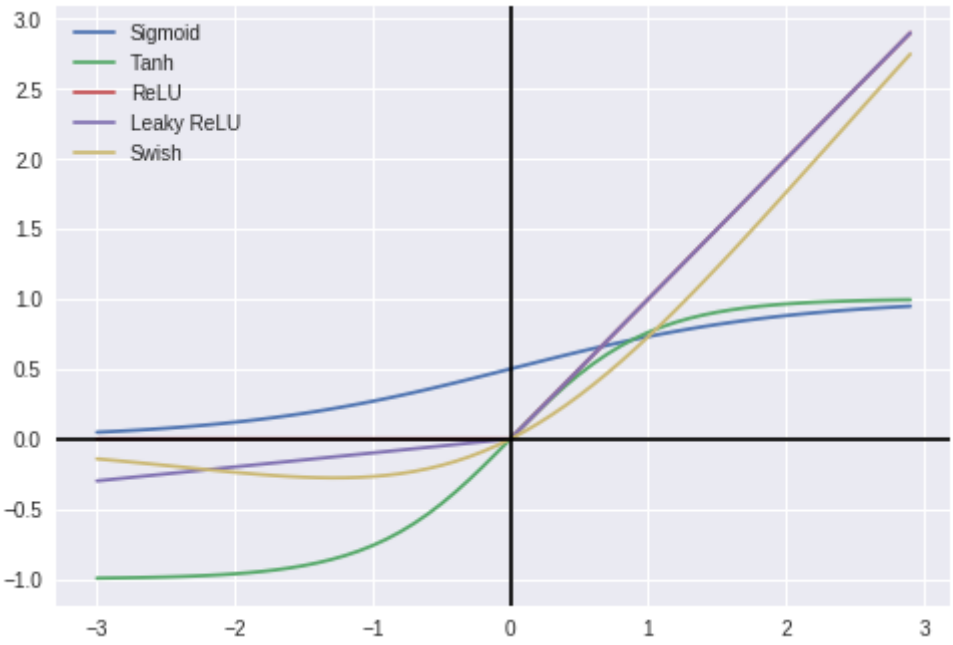
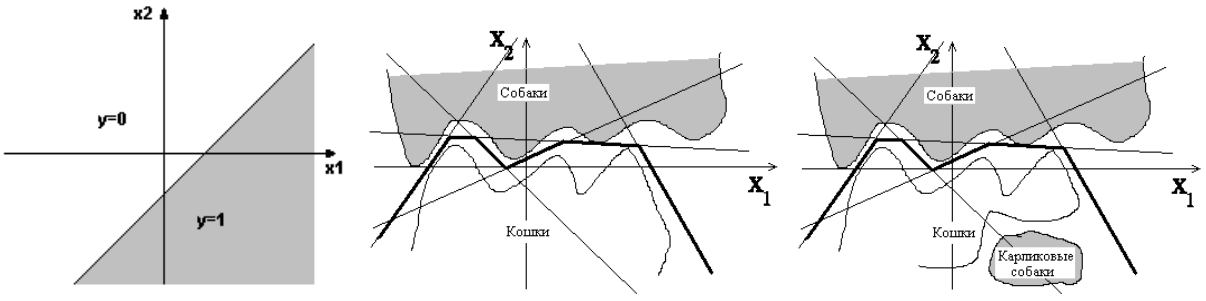


Рис. 2.10. Различные функции активации

Один формальный нейрон в самом простом случае, когда его функцией активации является ступенчатая функция, формирует на выходе 1, если $S = \sum_{i=1}^{n} w_i x_i + w_0 \geq 0$, и 0, если $S < 0$. Таким образом, он разбивает пространство входов на две части с помощью некоторой гиперплоскости. Если система имеет всего два входа, то это пространство двухмерно, и нейрон будет разбивать его с помощью прямой линии. Это разбиение определяется весовыми коэффициентами w_i нейрона.



а б в

Рис. 2.11. Нейронная сеть в построении разделяющей поверхности

Таким образом, один нейрон строит линейную регрессию в пространстве признаков (рис. 2.11, а).

Но не всегда пространство признаков разделимо одной линией. Пусть необходимо разделить собак и кошек по двум признакам – длина хвоста x_1 и длина туловища x_2, а они линейно неразделимы (рис. 2.11, б). Если составить из N таких нейронов слой, то получится разбиение многомерного пространства входов N гиперплоскостями. Если теперь выходы первого слоя нейронов использовать в качестве входов для нейронов второго слоя, то нетрудно убедиться, что каждая комбинация нулей и единиц на выходе второго слоя может соответствовать некоему объединению, пересечению и инверсии областей, на которые пространство входов разбивалось первым слоем нейронов. Двухслойная сеть, таким образом, может выделять в пространстве входов произвольные выпуклые односвязные области.

В случае еще более сложной задачи, когда требуется различать многосвязные области произвольной формы (рис. 2.11, в), всегда достаточно трехслойной сети. Говоря более строго, двухслойная сеть нейронов реализует все логические функции бинарных переменных:

И: $x1 \wedge x2 = [x1 + x2 - 3/2 > 0]$,

ИЛИ: $x1 \vee x2 = [x1 + x2 - 1/2 > 0]$,

НЕ: $-x1 = [-x1 + 1/2 > 0]$,

кроме исключающего ИЛИ,

XOR $x1 \oplus x2 = [x1 \oplus x2]$,

для которого нужна трехслойная сеть.

Рассмотрим конкретный пример – выделение изображений с кошками из большого набора изображений [Козлов, 2018]. В качестве тренировочного набора данных используем CIFAR-10 (рис. 2.12).

Общая схема решения задачи представлена на рис. 2.13. Изображение с кошкой представим в виде одномерного массива пикселей x размером $1\times(32\times32\times3) = 1\times3072$, а классификатор – в виде матрицы $\omega = [\omega_0, \ldots, \omega_9]$ размером 3072×10 (соответственно числу классов в датасете); столбец матрицы, соответствующий классу кошек, обозначим ω_0. Если веса ω подобраны удачно, то перемножение $x\cdot\omega$ дает оптимальный угол наклона гиперплоскости, которая отделяет кошек от не-кошек в пиксельном пространстве. Суммирование с матрицей b дает дополнительное линейное смещение этой плоскости. Аналогичные выражения будут справедливы для всех классов.

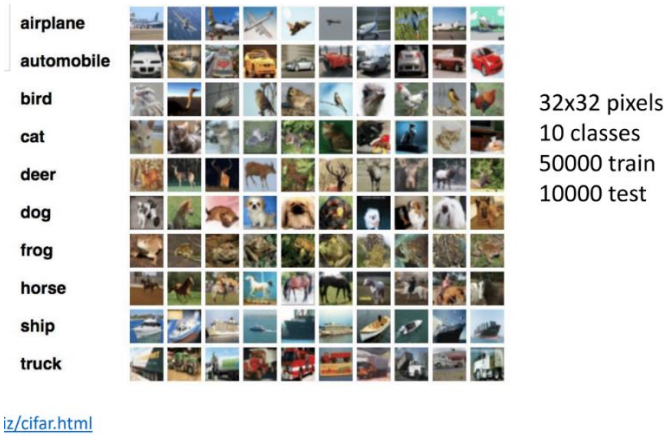


Рис. 2.12. Тренировочный набор данных CIFAR-10

Таким образом, на выходе получится матрица $x \cdot \omega + b$ размером 1×10, в которой максимальный элемент соответствует классу изображения, а отрицательное значение – насколько данный элемент НЕ соответствует классу.

Построение пары матриц ($x \cdot \omega + b$) является базовой операцией, выполняемой отдельным слоем любой нейронной сети. Например, в случае глубоких сетей члены матрицы последнего слоя – это коэффициенты при сформированных сетью признаках самого высокого уровня.

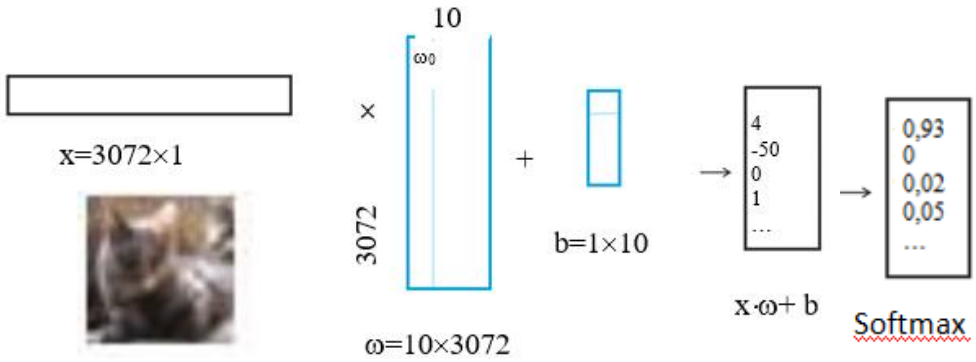


Рис. 2.13. Нейронная сеть как классификатор

Однако для вычисления оптимальных весов ω невыгодно максимизировать непосредственно матрицу $x \cdot \omega + b$, так как при небольшом изменении ее коэффициентов (например, от 4 к 4,01 или даже 5) положение максимума изменится очень слабо. Поэтому на выходе нейронной сети используется функция softmax. В предположении, что значения (scores) компонентов матрицы $x \cdot \omega + b$ – это вероятности соответствующих классов c, причем они распределены по нормальному закону, можно перейти от абсолютных значений матрицы $x \cdot \omega + b$, т.е. от абсолютных расстояний объекта до разделяющих плоскостей, к вероятностям сдвига. Для этого вводится дополнительный слой, реализующий функцию softmax:

$$\text{soft max}(\vec{v}) = \frac{\exp(v_i)}{\sum_k \exp(v_i)},$$

который каждое значение матрицы умножает на соответствующий множитель. Например, для scores = 4, показанных на рис. 2.13, он равен $\frac{e^4}{e^4 + e^{-50} + e^0 + e^1}$. В результате даже небольшие изменения scores (от 4 к 4,01) находят отражение в весах, т.е. градиент будет выражен. Кроме того, переход от абсолютных весов к вероятностям идеологически соответствует принципу максимального правдоподобия, что позволяет обоснованно ввести функцию потерь как оценку соотношения между полученными значениями на софтмаксе и ожидаемым выходным значением:

$$-\ln p(\text{data}) = -\sum \ln \text{p}(c=y_j \,|\, x_j) = -\sum \ln \frac{\exp\left(\left(\overrightarrow{w_y}\right)_j \left(\overrightarrow{x_j}\right) + \left(b_y\right)_j\right)}{\sum_i \exp\left(\left(\overrightarrow{w_y}\right)_j \left(\overrightarrow{x_j}\right) + \left(b_y\right)_j\right)},$$

где $p(\text{data}) = \bigcap_j p\left(c = y_j \middle| x_j\right)$ – вероятность получить все картинки из датасета при конкретных значениях весов матрицы ω.

Очевидно, что самыми «правильными» будут такие значения весов, которые максимизируют эту вероятность. Для их поиска используется метод обучения нейронных сетей, называемый «обратным распространением» (backpropagation), также известный как обратный режим автоматического дифференцирования. Его предложил финский математик Сеппо Линнайнмаа в своей диссертации в 1970 г. [Linnainmaa, 1970], а в 1986 г. метод был «переоткрыт» и популяризирован [Rumelhart, 1986].

Метод состоит в вычислении обратного распространения ошибки, которое направлено на минимизацию функции потерь путем корректировки весов и смещений сети. Для этого:

- вычисляется суммарная ошибка (total_error) как разность между ожидаемым значением *y* (из обучающего набора) и полученным значением *y* (посчитанным на этапе прямого распространения ошибки), проходящих через функцию потерь (cost function).
- вычисляется частная производная ошибки по каждому весу (эти частные производные отражают вклад каждого веса в общую ошибку (total_loss)).
- эти производные умножаются на число η, называемое скоростью обучения (learning rate).
- полученный результат вычитается из соответствующих весов.

В результате получатся следующие обновленные веса:

$$\omega_1 = \omega_1 - (\eta * \partial(\text{err}) / \partial(\omega_1))$$
$$\omega_2 = \omega_2 - (\eta * \partial(\text{err}) / \partial(\omega_2))$$
$$\ldots\ldots$$

Как легко видеть, в полученные выражения входят значения производных функции активации, откуда понятна важность выбора вычислительно эффективной функции активации. В качестве функции потерь также выбираются достаточно простые формы, в том числе квадратичная (среднеквадратичное отклонение), кросс-энтропийная, расстояние Кульбака-Лейблера (прирост информации).

Глубокие нейронные сети. Если между наборами линейных слоев вставить нелинейные преобразования, то выразительная сила нейронной сети кардинально улучшится (рис. 2.14). Конструкция «наборы линейных слоев + нелинейные слои между ними + выходной слой софтмакс для перевода коэффициентов в вероятности» – это уже глубокие НС (ГНС).

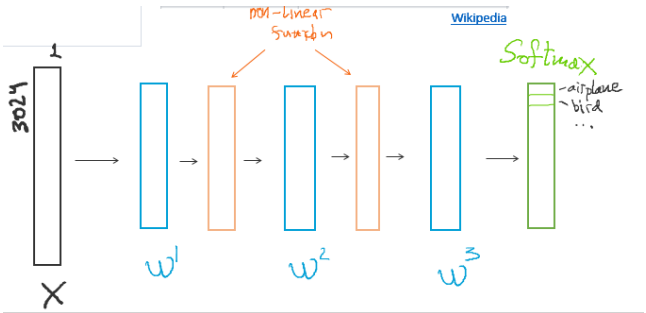
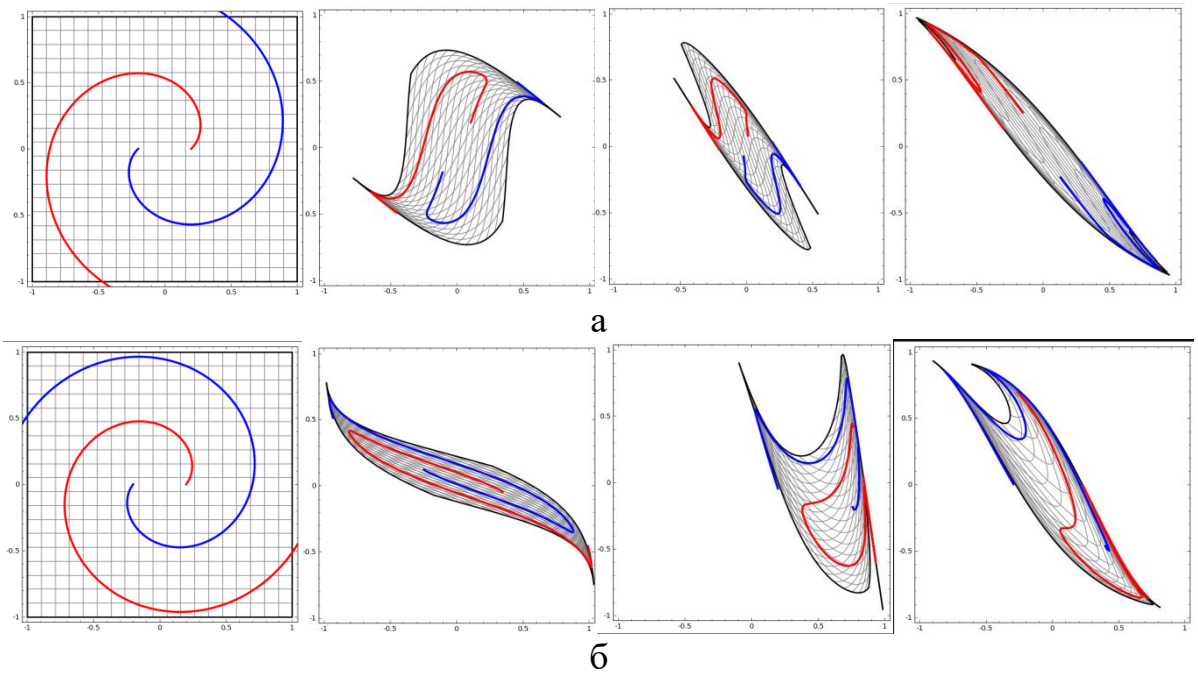


Рис. 2.14. Схема глубокой НС

Грубо говоря, разделительные поверхности (рис. 2.11) в этом случае отрисовываются не гиперплоскостями, а гиперповерхностями, что позволяет добиться большей точности.

Последовательность преобразований межслойного признакового пространства в процессе обучения сети проиллюстрирована на рис. 2.15 [Косолапов, 2018].



а

б

Рис. 2.15. Нейронная сеть преобразует признаковое пространство

Задача сети – классифицировать две спирали, используя четыре скрытых слоя, где каждый слой – это аффинное преобразование + поточечное применение монотонной функции активации. На рисунках показана последовательность решения этой задачи. Видно, что нейронная сеть не строит разделяющую поверхность сложной формы, а в ходе обучения пытается произвести топологическое преобразование признакового пространства таким образом, чтобы в результате между двумя спиралями можно было провести линейную границу. В первом случае ей это удается, а во втором случае, когда спирали «запутаны» сильнее, точность разделения уже не 100% (сети не хватило размерности, т.е. количества обучаемых нейронов).

Однако простое увеличение числа слоев в НС приведет к тому, что число нейронов, т.е. весов или степеней свободы, запредельно возрастет, что позволит сети «заучивать» все тренировочные данные и неминуемо приведет к переобучению. Противовесом этому послужили сверточные нейронные сети (СНС) (рис. 2.16).

СНС состоит из разных видов слоев: сверточные (convolutional) слои, субдискретизирующие (subsampling) слои и слои «обычной» нейронной сети – персептрона. Первые два типа слоев, чередуясь между собой, формируют входной вектор признаков для многослойного персептрона.

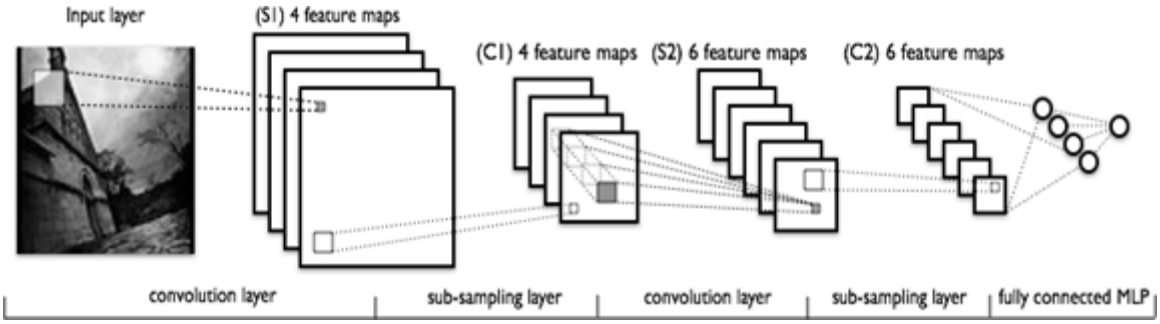


Рис. 2.16. Схема сверточной нейронной сети LeNet

Сверточный слой (convolutional layer) представляет из себя набор карт (feature maps). У каждой карты есть синаптическое ядро (в разных источниках его называют также сканирующее ядро или фильтр) – фильтр или окно, которое скользит по всей области предыдущей карты. Размер ядра обычно берут в пределах от 3×3 до 7×7.

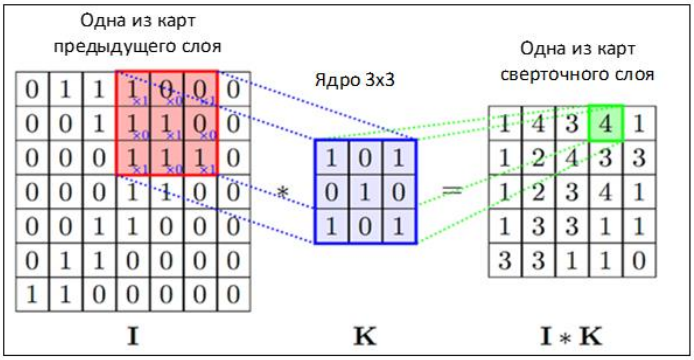


Рис. 2.17. Схема работы сверточного слоя

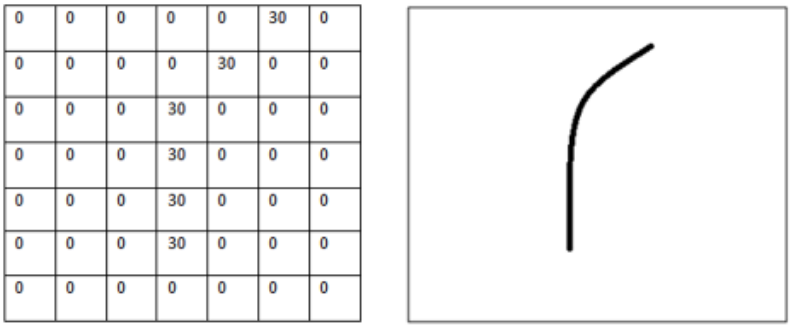
Ядро производит операцию свертки, которая часто используется для обработки изображений:

$$f * g = \sum_{1}^{k} f\left[m-k, n-l\right] * g\left[k, l\right],$$

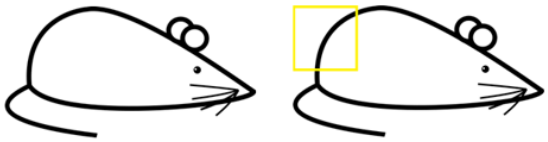
где f – исходная матрица изображения, g – ядро свертки. Неформально эту операцию можно описать следующим образом: окном размера ядра g проходим с заданным шагом (обычно 1) все изображение f, на каждом шаге поэлементно умножаем содержимое окна на ядро g, результат суммируем и записываем в матрицу результатов, как на рисунке. Используя разные фильтры, можно получить разные типы границ и разные варианты обработки изображений.

В СНС каждый фильтр можно рассматривать как идентификатор отдельного свойства – прямые границы, простые цвета, участки кривых и пр. В результате формируется фильтр, выделяющий некоторый характерный участок изображения (рис. 2.18).

Задавая разные коэффициенты матрицы, можно выделять различные участки изображения (рис. 2.19).

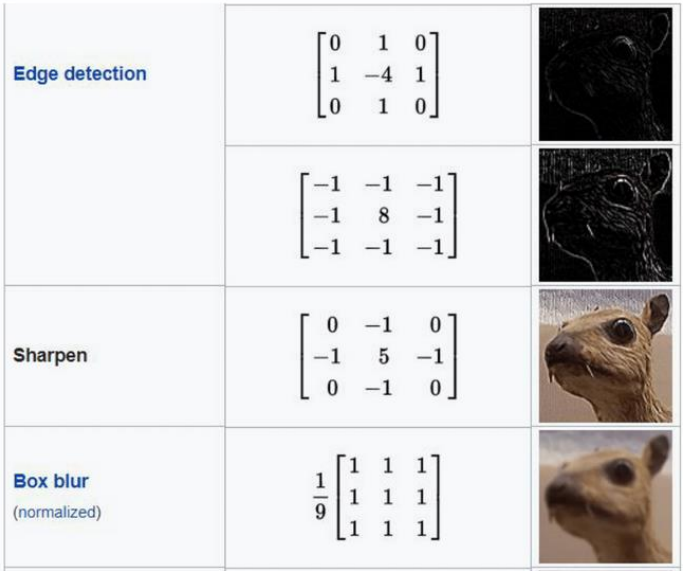


а



б

Рис. 2.18. Пример работы фильтра: для зоны изображения, выделенной желтым квадратом, фильтр сформирует максимальный сигнал



а б

Рис. 2.19. Фильтрация в сверточном слое: а – исходное изображение, б – различные матрицы фильтрации

Субдискретизирующий слой (pooling layer) уменьшает размерность карт предыдущего слоя, используя тот факт, что изображения обладают свойством локальной скоррелированности пикселей – соседние пиксели, как правило, не сильно отличаются друг от друга. Таким образом, если из них сформировать какой-либо агрегат, то потери информации будут незначительными. Операция субдискретизации (рис. 2.20) также является операцией свертки. Важно, что введение этой операции сохраняет возможность вычисления градиента в обратном ходе, так как вместо всех разных нейронов будет работать выбранный максимальный нейрон.

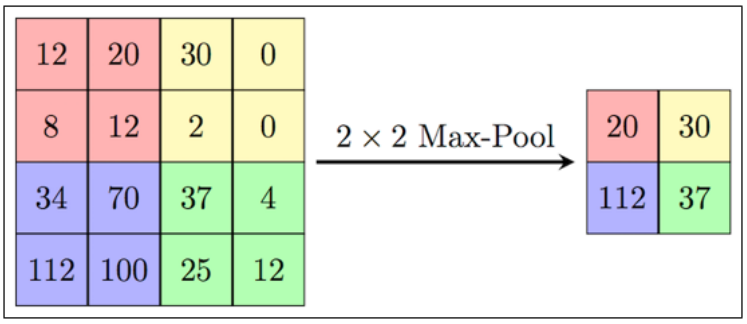


Рис. 2.20. Схема работы субдискретизирующего слоя

Таким образом, каждая последующая сборка «сверточный слой + субдискретизирующий слой» выделяет все более высокоуровневые (обобщенные) признаки изображения, которые в конце поступают на полносвязные слои и на softmax. Сборки таких слоев можно ставить друг за другом или в другом разумном порядке. При этом получаются разные архитектуры (топологии) СНС.

Первая реально работающая архитектура СНС – сеть LeNet [Lecun, 1998] (рис. 2.16), которая распознавала рукописные почтовые индексы на почтовых отправлениях. Сеть имела 60 тысяч параметров, ее основные «строительные блоки» – свертки 5×5 со сдвигом 1 и пулинг 2×2 со сдвигом 2.

Появление графических процессов (GPU) позволило значительно увеличить количество обучаемых параметров. Следующая модель СНС, AlexNet [Krizhevsky, 2012], уже содержала 60 миллионов параметров, и для обучения такой модели потребовались два графических ускорителя (рис. 2.21).

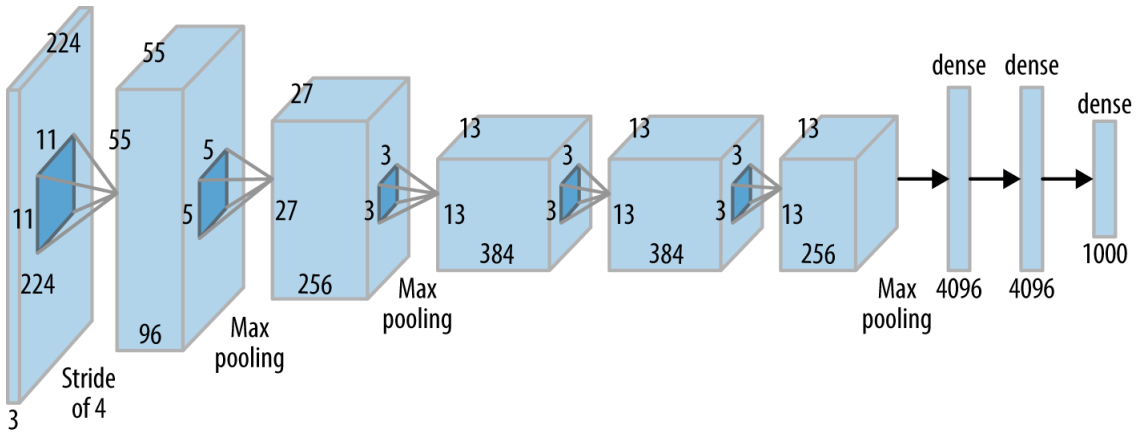


Рис. 2.21. Архитектура сети AlexNet

Однако произвольное увеличение ширины (количество нейронов в слоях) и глубины (количество слоев) сети имеет ряд недостатков. Во-первых, увеличение количества параметров способствует переобучению, а увеличение количества слоев добавляет еще и проблему затухания градиента. Во-вторых, увеличение количества сверток в слое приводит к квадратичному увеличению вычислений в этом слое. Если же новые параметры модели используются неэффективно, например, многие из них становятся близки к нулю, тогда вычислительные мощности тратятся впустую.

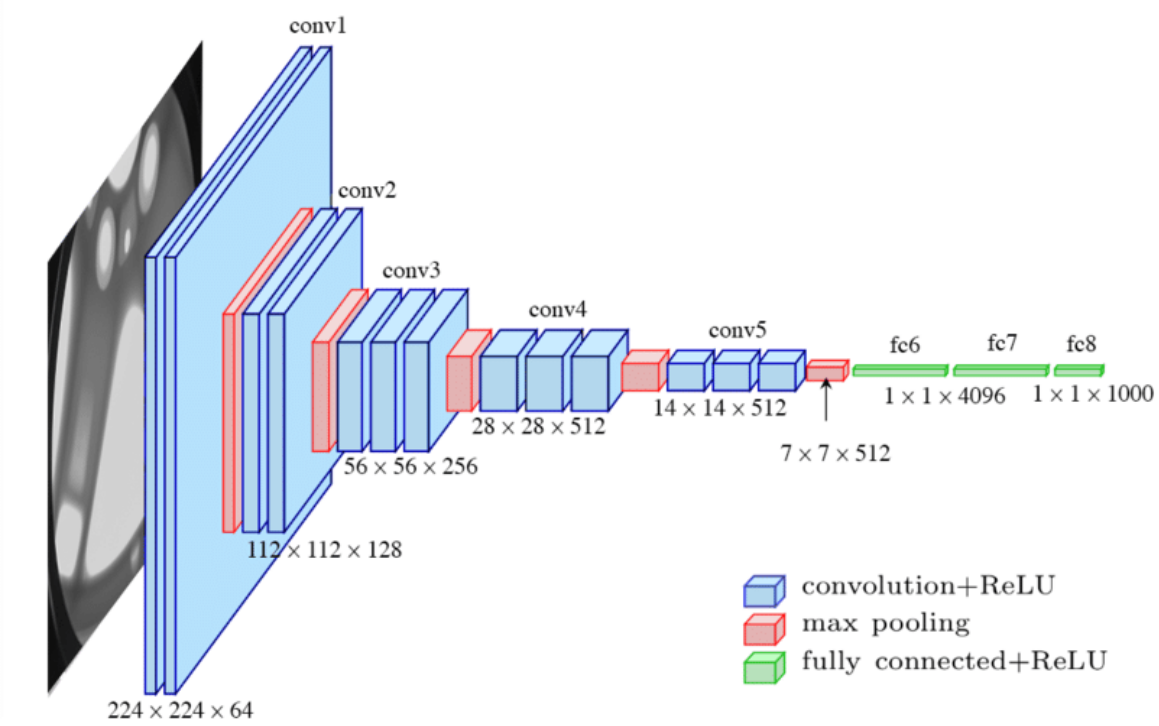


Рис. 2.22. Архитектура VGG-16

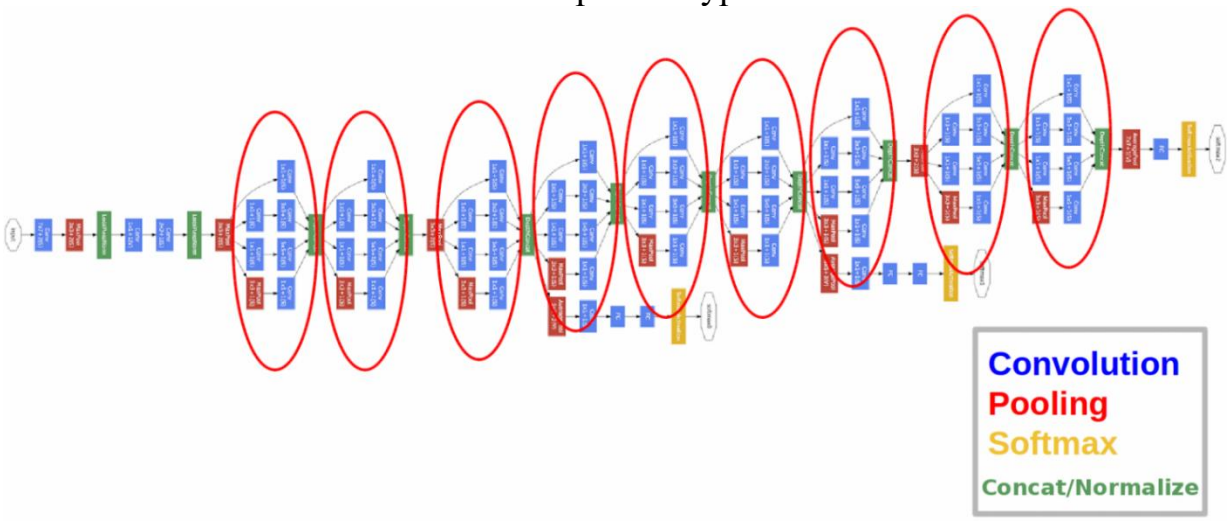


Рис. 2.23. Архитектура GoogLeNet. Кружками выделены Inception blocks

Для уменьшения числа обучаемых параметров было предложено использовать стек сверточных слоев или конкатенацию нескольких параллельно работающих сверток в отдельном блоке (Inception block), что привело к созданию архитектур VGG [Simonyan, 2015] (рис. 2.22) и GoogLeNet [Szegedy, 2014] (рис. 2.23) соответственно.

Еще один способ уменьшения числа настраиваемых параметров – использовать так называемое остаточное обучение (residual learning): сеть собирается из фрагментов (рис. 2.24), каждый из которых обучается к уже известному входу (x) достраивать изменение $F(x)$. Это сильно упрощает количество варьируемых параметров в матрице, так как резкие изменения на реальных обрабатываемых изображениях встречаются достаточно редко. Этот принцип был реализован в таких архитектурах, как ResNet [He, 2016] (рис. 2.25) и UNet [Ronneberger, 2015] (рис. 2.26).

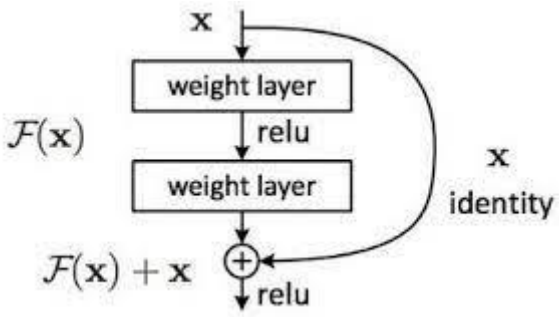


Рис. 2.24. Блок остаточного обучения

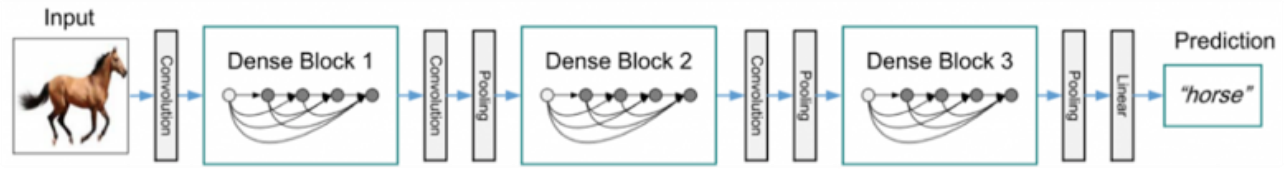


Рис. 2.25. Схема архитектуры ResNet

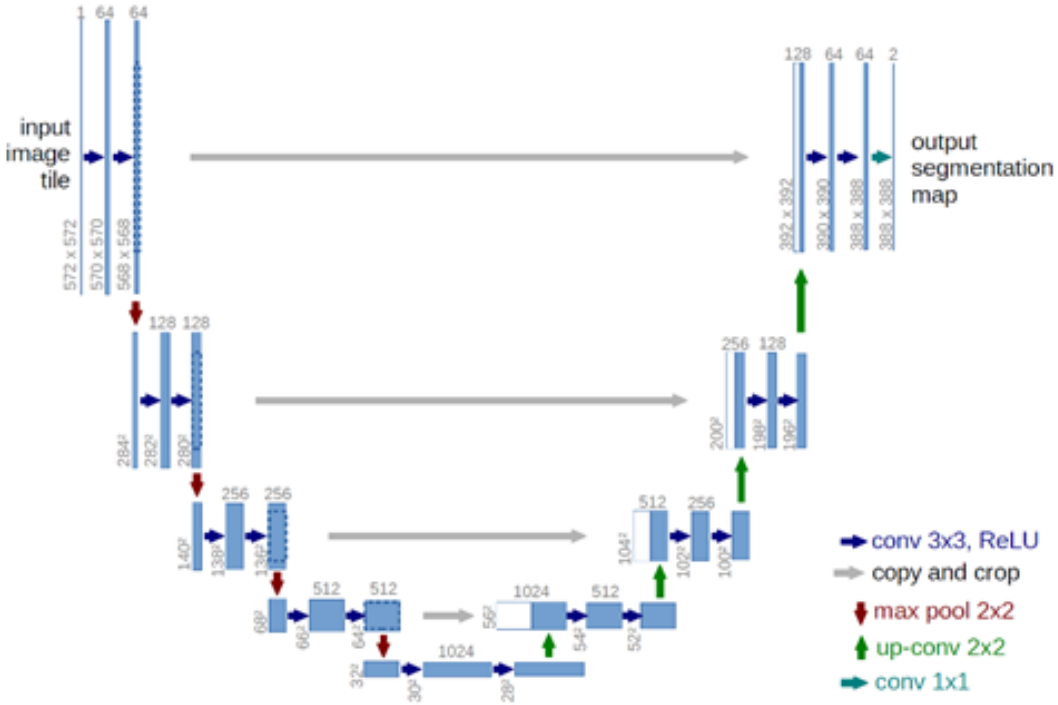


Рис. 2.26. Схема архитектуры U-Net

U-Net – это архитектура свёрточной нейронной сети, принципиально создававшаяся для сегментации изображений (первоначально – для биомедицинских изображений). Архитектура сети представляет собой последовательность слоёв свёртка+пулинг, которые сначала уменьшают пространственное разрешение картинки (прямой проход – слева), а потом увеличивают его (обратный проход – справа). При этом для разворачивания изображения в обратном ходе добавляются тождественные проходы (short-cut connections), т.е. создается дополнительная связь между блоками с одинаковым разрешением на входной и выходной ветвях. В результате на прямом проходе сеть выделяет все более высокоуровневые признаки, и к концу прямого прохода формируется макроструктура

изображения, а на обратном проходе восстанавливается микроструктура изображения из активаций, записанных на прямом проходе.

Архитектура U-Net оказалась исключительно важной для медицины, где размеры обучающих наборов данных принципиально ограничены сравнительно небольшим количеством больных с конкретным заболеванием. Здесь вообще нет полносвязных слоев, в которых сосредоточено большинство параметров сети; кроме того, сеть ориентирована на обработку контекстно-богатых медицинских изображений, в которых практически не встречаются «пустые» пиксели. Это позволяет успешно обучать сеть на очень маленьком тренировочном датасете (в исходной статье он содержал всего 30 изображений).

Внимание в глубоких НС. Внимание играет жизненно важную роль в восприятии [Corbetta, 2002]. Механизмы внимания в сочетании с памятью обеспечивают способность человека концентрироваться на каком-то одном компоненте информационного потока и игнорировать другие. Это позволяет людям выделять существенную информацию из зашумленных данных (например, поддерживать разговор с конкретным собеседником в шуме толпы), целенаправленно обдумывать одну идею на фоне потока сознания, запоминать одно, но важное событие из всех одновременно происходящих, и т.д.

Реализация механизмов внимания средствами глубокого обучения впервые была представлена в статье [Bahdanau, 2015]. В основе лежит идея, аналогичная поисковым системам. Например, когда вы ищете в Интернете видео, поисковая система сопоставит ваш запрос (текст в строке поиска) с набором ключей (название видео, описание и т. д.), связанных с видеороликами в своей базе данных, а затем представит вам наиболее подходящие значения (конкретные видеоролики). Аналогично, функцию внимания для нейронных сетей можно описать как отображение запроса и набора пар ключ-значение на выходные данные, где запрос, ключи, значения и выходные данные являются векторами. Для формирования этих векторов нужно предварительно перевести элементы потока в такое признаковое пространство, где определимо расстояние между ними, т.е. вычислить эмбеддинг каждого элемента [Mikolov T. et al. Distributed Representations of Words and Phrases and their Compositionality. arXiv:1310.4546 [cs.CL]]. С этой целью используется комбинация «энкодер-декодер» (рис. 2.27), а механизм внимания (рис. 2.28) включается между ними.

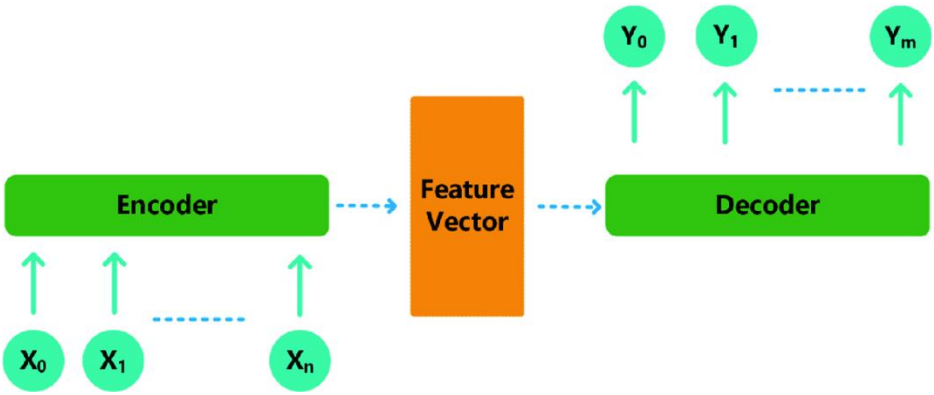


Рис. 2.27. Схема преобразования «энкодер-декодер»

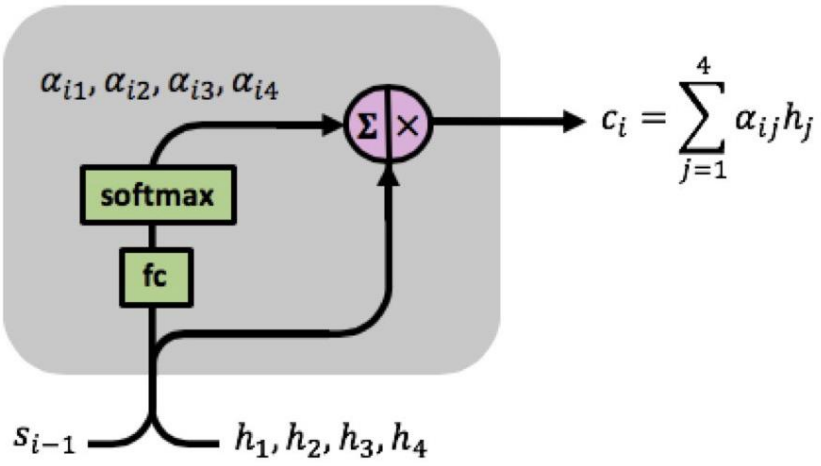


Рис. 2.28. Механизм внимания

На вход механизма внимания подаются h_1, h_2, h_3, h_4 – сформированные энкодером эмбеддинги элементов входной последовательности x_1, x_2, x_3, x_4, а также состояния декодера s_0, s_1, s_2, s_3. Выходом механизма внимания является последовательность так называемых контекстных векторов c_1, c_2, c_3, c_4:

$$c_i = \sum_j a_{ij} h_j .$$

Для получения весов внимания a_{ij} используется связка однослойной полносвязной НС *fc* и функции softmax. На каждом шаге *i* на вход *fc* подается конкатенация векторов $[s_{i-1}, h_i]$ и вычисляется

$$e_{ij} = fc\,[s_{i-1}, h_i].$$

Значения e_{ij} посредством функции softmax приводятся к интервалу [0,1]:

$$a_{ij} = \frac{\exp e_{ij}}{\sum_k \exp e_{ik}}.$$

Посмотрим, например, как работает механизм внимания при обработке предложения **`I like fresh fruits.`**

Делаются эмбеддинги для всех слов предложения.

Берется вектор первого слова и вектор второго слова, подаются на однослойную сеть с одним выходом, которая выдает степень их похожести (скалярная величина). Эта скалярная величина умножается на вектор второго слова, получая его некоторую «ослабленную» на величину похожести копию. Делая то же самое для всех оставшихся слов предложения, получаются их «ослабленные» (взвешенные) копии, которые выражают степень их похожести на первое слово.

Все эти взвешенные вектора складываются друг с другом, давая один результирующий вектор (размерностью в один эмбеддинг):

```
output = like*weight(I,like) + fresh*weight(I,fresh) +
+ fruits*weight(I, fruits).
```

Таким образом, веса внимания a_{ij} отражают важность h_j по отношению к предыдущему скрытому состоянию s_{i-1} при определении следующего состояния s_i и генерации y_i. Больший вес внимания a_{ij} заставляет декодер фокусироваться на

входных значениях x_j (представленных выходными значениями энкодера h_j) при прогнозировании выходных значений y_i.

Предложены различные модификации механизма внимания.

Self-attention (рис. 2.29) для каждого элемента последовательности формирует взвешенную комбинацию эмбеддингов всех других элементов в той же последовательности (включая те, которые появляются в ней позже).

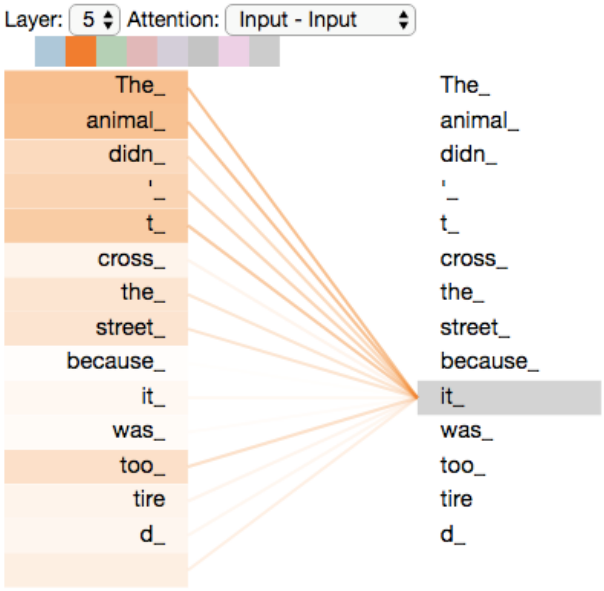


Рис. 2.29. Механизм self-attention при разборе предложения [Alammar, 2018]

Multi-head attention [Vaswani, 2017] использует несколько одиночных механизмов внимания, работающих параллельно (рис. 2.30, б). Это дает возможность одновременно рассматривать связь конкретного слова с другими словами в различных аспектах (например, в грамматическом и в смысловом). Обратим внимание на то, что для получения матрицы весов внимания a_{ij} здесь используется не обучаемая связка *fc* – softmax, а просто скалярное произведение векторов $s_j^{\mathrm{T}} \cdot h_i$ (рис. 2.30, а), что сильно упрощает вычисления.

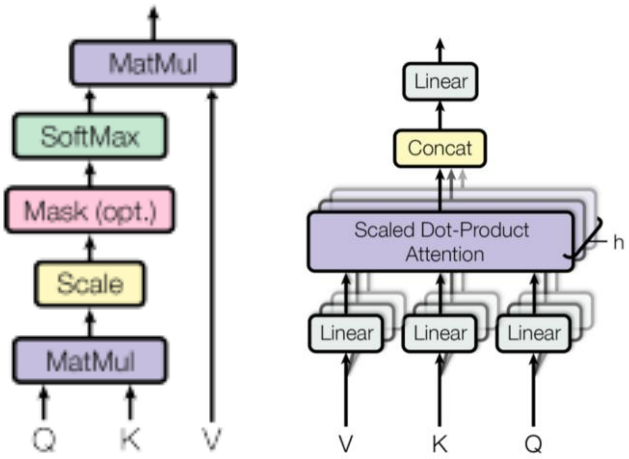


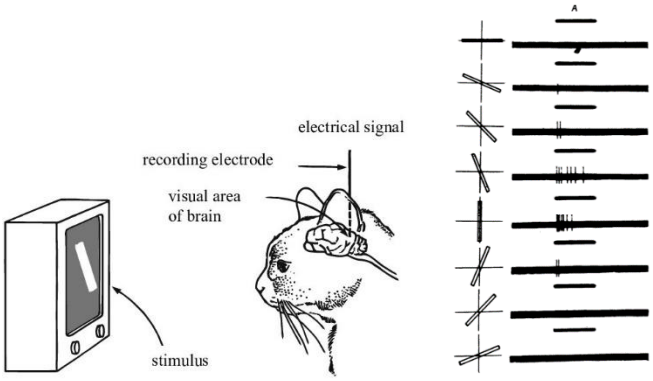
Рис. 2.30. а - Scaled Dot-Product Attention, б - Multi-head attention: Q (query) - вектор входа в декодере, K (key) и V (value) - финальные эмбеддинги энкодера

Концепция внимания может использоваться для разных модальностей, а также для их сопоставления. Например, в работе [Xu, 2015] механизм внимания применяется к изображению для создания подписи к нему (рис. 2.31). Изображение сначала кодируется сверточной нейронной сетью для извлечения признаков. Затем декодер использует функции свертки для формирования последовательности слов подписи. Визуализация весов внимания наглядно демонстрирует, на какие области изображения обращает внимание модель, чтобы вывести определенное слово.

Рис. 2.31. Исходное изображение и последовательность формирования подписи к нему (A woman is throwing a frisbee in a park)

Взаимосвязь глубокого обучения и функционирования реального мозга. Многие идеи и технологические решения, применяемые в глубоком обучении, имеют свои аналогии в функционировании мозга животных и человека. Показательный пример такой аналогии – механизм зрения высших животных и человека.

Первые эксперименты [Hubel, 1959] были проведены в 1959 г. на кошках. Эксперимент состоял в следующем: найти, какие именно нейроны зрительных отделов коры головного мозга кошки (рис. 2.32, а) реагируют на разные типы изображения. Результат оказался удивительным: на каждый простой тип изображения (например, отрезки с учетом ориентации и позиции) реагирует свой нейрон (рис. 2.32, б). Нейроны, представленные на этих рисунках, теперь называются простыми нейронами (simple cells).



а б

Рис. 2.32. Эксперимент Хьюбела и Визеля 1959 г.:
а – схема эксперимента; б – предпочтительная реакция конкретного нейрона на различные ориентации предъявляемого отрезка

Современный взгляд на механизм зрения человека значительно более детализирован (рис. 2.33). По мере прохождения визуальной информации через визуальную иерархию – от сетчатки через зрительный нерв и латеральное коленчатое тело (lateral geniculate nucleus, LGN) в зрительные зоны коры (cortex) – происходит все более сложная обработка. Палочки и колбочки сетчатки (retinal rodes and cones) обеспечивают спектральную фильтрацию, а нейроны LGN – пространственную и временную фильтрацию входного изображения.

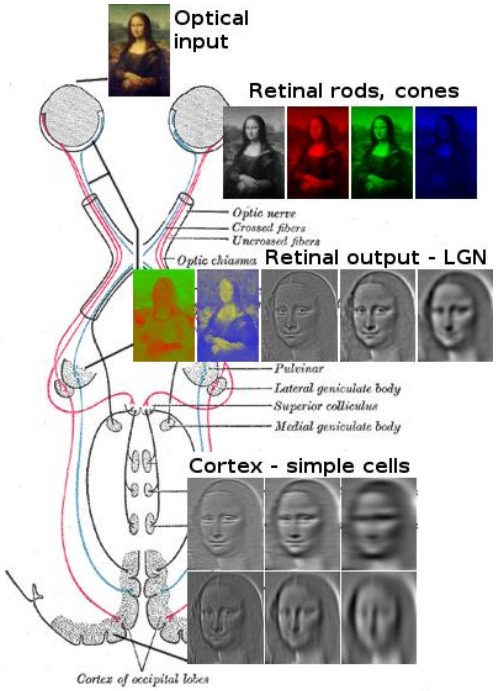
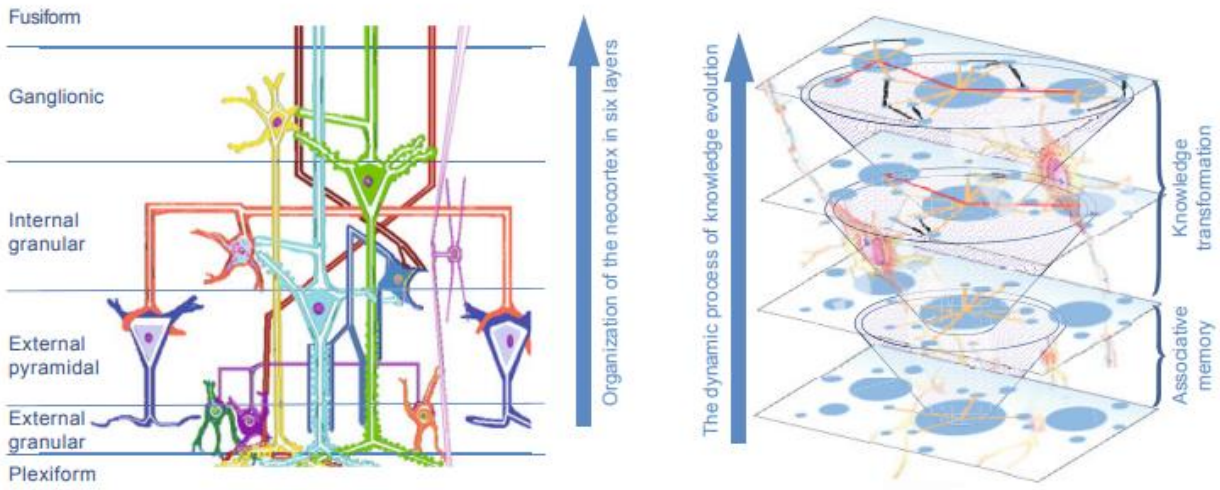


Рис. 2.33. Схема обработки визуальной информации в зрительной системе

В зрительной коре решается несколько взаимосвязанных задач: имеется зона, создающая восходящую карту заметности (salience map) поля зрения в зависимости от внимания к определенной зоне изображения; простые нейроны (simple cells) выделяют широкий спектр зрительных примитивов, в том числе полосы определенной ориентации, края и углы, информацию о цвете и движении и др.; имеются зоны, избирательно реагирующие на законченный объект (например, человеческое лицо), на собственное смещение относительно поля зрения и

на смещение объектов интереса в нем. Зрительная зона коры также имеет прямые связи с зонами, управляющими движением глаз и рук, что поддерживает исполнительную часть процесса обработки зрительной информации.

Таким образом, согласно представлениям когнитивистики [Li, 2002; Zhaoping, 2019; Jessell, 2000], зрение человека рассматривается как совокупность этапов кодирования, выбора и декодирования, управляемых вниманием, причем решающую роль в этом процессе играет неокортекс – новые области коры головного мозга, которые у низших млекопитающих только намечены, а у человека составляют основную часть коры.



а б

Рис. 2.34. Процесс синергетического действия механизма памяти мозга и механизма передачи знаний: (а) динамичный процесс эволюции знаний; (б) организация неокортекса в шесть слоев

Когнитивные нейробиологи еще в 1990-х гг. пришли к выводу [Zheng, 2017], что неокортекс имеет иерархическую структуру (рис. 2.34, а), а его развитие представляет собою самоорганизующийся процесс, в результате которого формируется стек датчиков и фильтров, оптимально настроенный на свою рабочую среду и решаемые задачи. Рассмотренный выше процесс обработки информации в зрительной системе человека хорошо иллюстрирует этот тезис. Еще одним примером может служить процесс формирования человеческого знания, который происходит параллельно с мыслительной деятельностью на базе ассоциативной памяти (рис. 2.34, б).

Можно констатировать, что современные технологии глубокого обучения успешно реализуют практически все механизмы, характерные для нейробиологии. Подобно неокортексу в мозге, нейронные сети используют иерархию многоуровневых фильтров, в которой каждый слой обрабатывает информацию из предыдущего уровня (или операционной среды), а затем передает свои выходные данные (и, возможно, исходные входные данные) другим слоям; также успешно реализованы механизмы внимания. В результате этого процесса получается самоорганизующийся стек обработчиков, настроенный на решение конкретной задачи.

2.4.2. Большие данные

Большие данные — это совокупность данных, которые не могут быть собраны, управляемы и обработаны с помощью обычных программных инструментов в течение определенного периода времени. Стандарт [ГОСТ Р 59277–2020] определяет большие данные следующим образом.

большие данные (big data): Обширные наборы данных – главным образом, по таким характеристикам данных, как объем, разнообразие, скорость генерации и/или изменчивость – которые требуют использования технологии масштабирования для эффективного хранения, обработки, управления и анализа.

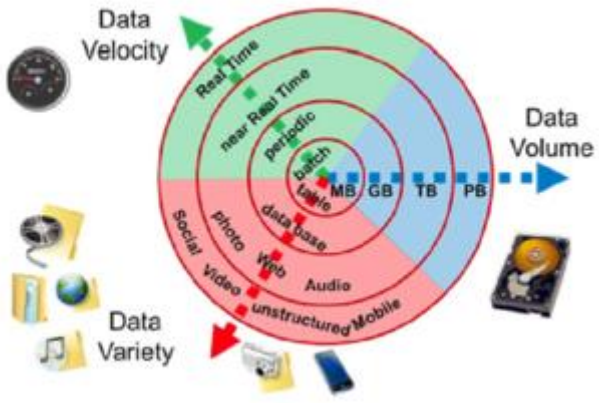


Рис. 2.35. Характеристики «5V» больших данных (предложенные IBM): Volume, Velocity, Variety, Value, Veracity — объем, скорость, разнообразие, ценность, достоверность

Термин был предложен в 1990-х гг. по аналогии с терминами «большая руда», «большая нефть» и пр. Но, как оказалось, это не просто маркетинговое название. Большие данные – это, по существу, подход к обработке данных, принципиально отличный от традиционной бизнес-аналитики: вместо выборочного опроса для анализа используются все данные; обработка происходит в реальном времени; данные в большинстве своем не структурированы; задача – не в том, чтобы полностью использовать датасет, а в том, чтобы выявить с его помощью необходимые закономерности (рис. 2.35).

Для обработки больших данных разработаны специальные технологии: технологии структурирования и хранения данных (OLAP vs NoSQL, XML, JSON, облачные системы хранения); распределенные параллельные вычисления (MapReduce и др); комплексные технологии обработки (Hadoop).

Технологии OLAP развивают идею реляционного хранения данных, используя многомерные кубы данных, что математически соответствует переходу к тензорному описанию данных.

NoSQL – комплексное название для хранилищ данных нереляционного типа. Они делятся на четыре группы:

– Ключ-значение (Key-value data store) (рис. 2.36). Каждое значение данных связывается с уникальным ключом, и хранилище ключей/значений

использует этот ключ для хранения данных с помощью соответствующей функции хеширования. Функция хеширования выбирается для обеспечения равномерного распределения хешированных ключей по хранилищу данных.

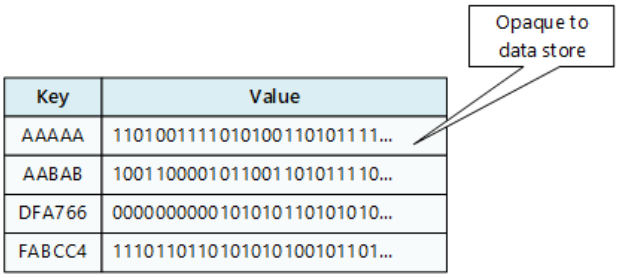


Рис. 2.36. Схема хранения данных «Ключ-значение»

– Ключ-документ (рис. 2.37) – частный случай предыдущей версии. Хранимой сущностью является документ.

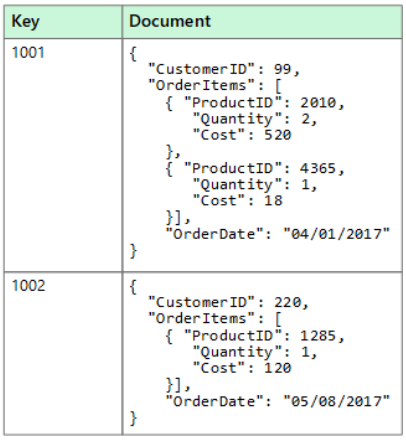


Рис. 2.37. Схема хранения данных «Ключ-документ»

– Графовые базы данных (рис. 2.38). Для хранения сущностей используются узлы, для хранения взаимосвязей между сущностями – ребра. Обход соединений или взаимосвязей в графовых базах данных выполняется очень быстро, поскольку взаимосвязи между узлами не вычисляются во время выполнения запроса, а хранятся в базе данных.

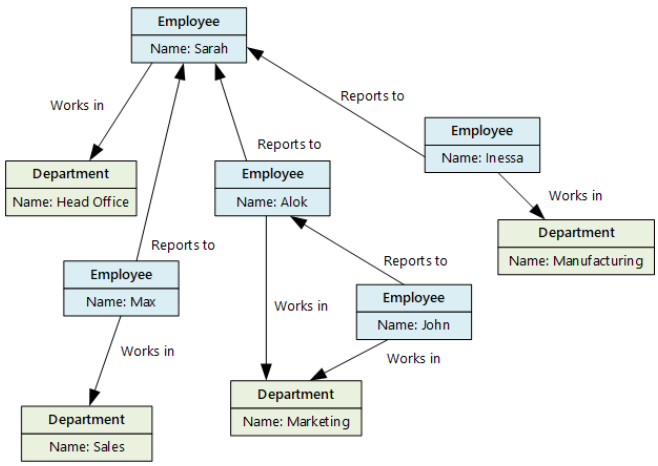


Рис. 2.38. Схема хранения данных в графовой базе

– Базы временных рядов (time series data stores) (рис. 2.39) предназначены для хранения большого числа сравнительно коротких записей. Они позволяют хранить неупорядоченные и запаздывающие данные, проводить автоматическую индексацию точек данных и быстро выполнять запросы, описанные через временные окна.

timestamp	deviceid	value
2017-01-05T08:00:00.123	1	90.0
2017-01-05T08:00:01.225	2	75.0
2017-01-05T08:01:01.525	2	78.0

Рис. 2.39. Схема хранения данных в базе временных рядов

Технология MapReduce [Dean, 2004] – это модель распределённых вычислений от компании Google, используемая в технологиях Big Data для параллельных вычислений над очень большими (до нескольких петабайт) наборами данных в компьютерных кластерах, и фреймворк для вычисления распределенных задач на узлах (node) кластера. Запросы разбиваются и распределяются по параллельным узлам и обрабатываются параллельно (шаг «Map»), затем результаты собираются (шаг «Reduce»). Благодаря этому программисты могут легко и эффективно использовать ресурсы распределённых Big Data систем.

Технология Apache Hadoop [Apache, 2012] – это свободно распространяемый набор утилит, библиотек и фреймворк для разработки и выполнения распределённых программ, работающих на кластерах из сотен и тысяч узлов. Ядро состоит из части хранения, известной как распределенная файловая система Hadoop (HDFS), и части обработки, которая представляет собой модель MapReduce. Hadoop разбивает файлы на большие блоки и распределяет их по узлам в кластере. Затем он передает упакованный код в узлы для параллельной обработки данных. Это позволяет обрабатывать набор данных быстрее и эффективнее, чем в более традиционной архитектуре суперкомпьютера, основанной на параллельной файловой системе, где вычисления и данные распределяются через высокоскоростную сеть.

Как подход больших данных соотносятся с обработкой аналогичной информации в мозгу человека? Компьютерная обработка больших данных обеспечивает движение к автоматической интегральной оценке ситуации, которую легко выполняет мозг человека, т.е., в конечном итоге, к автоматической компьютерной диагностике.

Например, для автоматизированного мониторинга эпилепсии у пациентов ежедневно создается от 5 до 10 ГБ данных. Точно так же одно несжатое изображение томосинтеза молочной железы занимает в среднем 450 МБ данных. Это лишь некоторые из многих примеров, когда компьютерная диагностика использует большие данные. По этой причине большие данные были признаны одной

из ключевых проблем, которые необходимо решить системам компьютерной диагностики, чтобы выйти на новый уровень производительности.

2.4.3. Обучение с подкреплением

Самообучение и самостоятельное принятие решений с первых шагов развития ИИ рассматривалось как один из главных способностей интеллекта. Современный подход к моделированию принятия решений основан на концепции обучения с подкреплением, предложенной в 1986 г. [Sutton, 2015].

Концептуально обучение с подкреплением (Reinforcement learning) – это почти то же самое, что и обучение с учителем, но в роли учителя выступает настоящая или виртуальная среда. Например, при обучении робота его помещают в лабиринт, из которого он сам должен найти выход. В процессе поиска робот получает от внешней среды информацию о том, где выхода нет, таким образом он изучает окружающий мир и учится находить путь к выходу. Наградой за успешно выполненное задание являются набранные в процессе решения баллы, а также возможность взяться за новое задание.

При обучении с подкреплением одновременно преследуются две цели:
– минимизировать ошибки. Машина учится анализировать информацию перед каждым следующим ходом. Например, беспилотный автомобиль во время обучения учится вовремя реагировать на сигнал светофора, остановиться перед пешеходом на переходе, пропустить быстро движущийся автомобиль или спецтранспорт справа. Чтобы достичь лучшего результата, автомобиль обучается в виртуальной модели города со случайными пешеходами и другими участниками дорожного движения;
– получить от выполнения задания максимальную выгоду. Сама выгода при этом должна быть запрограммирована заранее; это может быть максимально быстрое время прохождения маршрута, оптимальное расходование ресурсов предприятия, обслуживание как можно большего количества посетителей и т.д.

Обучение с подкреплением применяется там, где сложно понять, какое действие приводит к какому результату, т.е. нужно соизмерить отсроченную выгоду (цель) с ситуативным принятием решения. Этот вид обучения решает сложную задачу – соотнесения немедленных действий с отсроченной отдачей, которую они производят. Например, если бот обучается игре в Packman (рис. 2.40), то цель обучения – максимизировать набранные очки и при этом избегать “опасности” и, как следствие, проигрыша.

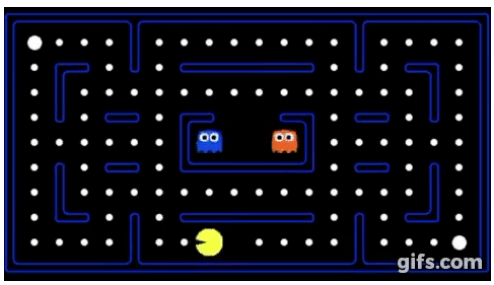


Рис. 2.40. Обучение бота игре в Packman [https://youtu.be/QilHGSYbjDQ]

Методы обучения с подкреплением имеют три общие ключевые идеи. Во-первых, целью всегда является оценка функций ценности (estimation of value functions). Во-вторых, они работают, отслеживая и сохраняя значения (бэкапы) в соответствии с реальными или возможными траекториями состояний. В-третьих, они следуют стратегии обобщенной итерации политики (generalized policy iteration, GPI), что означает, что они поддерживают приблизительную функцию ценности и приблизительную политику и постоянно пытаются улучшить одно на основе другого. В реальной практике могут использоваться разные комбинации этих трех идей.

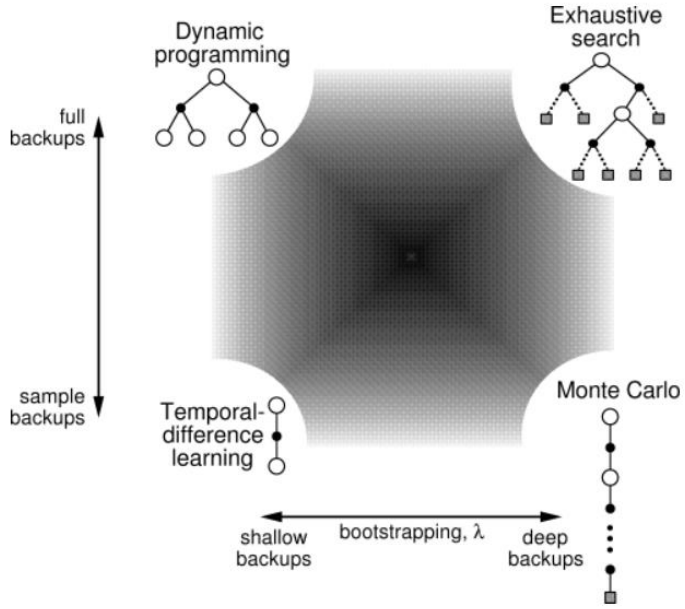


Рис. 2.41. Пространство методов обучения с подкреплением [Sutton, 2015]

На рис. 2.41 показан двумерный срез пространства методов обучения с подкреплением, связанный с типом бэкапа, используемого для улучшения функции ценности.

Вертикальное измерение заключается в том, являются ли они выборочными бэкапами (на основе выборочной траектории) или полными бэкапами (на основе распределения возможных траекторий). Для полных бэкапов, конечно же, требуется модель, в то время как пробные бэкапы могут выполняться как с моделью, так и без нее (еще одно измерение вариаций). Горизонтальная координата соответствует глубине бэкапа, то есть степени начальной загрузки. Вдоль нижнего края пространства находятся методы выборочного бэкапа, начиная от бэкапа только на один шаг назад (TD) и заканчивая резервным копированием Монте-Карло с полным возвратом.

В трех из четырех углов пространства находятся три основных метода оценки значений: динамическое программирование (DP), временна́я разница (TD) и метод Монте-Карло, а в четвертом углу – исчерпывающий поиск (например, полный перебор).

Метод временно́й разницы (TD) – это простейший метод проб и ошибок: делается первоначальная оценка, исследуется пространство поиска и

обновляется предыдущая оценка на основе полученного результата; если в задаче есть какая-то закономерность, то она найдется. Так учатся собаки Павлова (по принципу «звонок – еда») и другие животные.

Метод динамического программирования (DP) может реализовываться двумя способами:

- нисходящее динамическое программирование: задача разбивается на подзадачи меньшего размера, они решаются и затем комбинируются для решения исходной задачи. При этом решения уже решенных подзадач запоминаются.
- восходящее динамическое программирование: все подзадачи, которые впоследствии понадобятся для решения исходной задачи, просчитываются заранее и затем используются для построения решения исходной задачи. Этот способ лучше нисходящего программирования в смысле размера необходимого стека и количества вызова функций, но иногда бывает нелегко заранее выяснить, решение каких подзадач нам потребуется в дальнейшем.

В методе Монте-Карло процесс описывается математической моделью, в которую входит генератор случайных величин. Модель многократно запускается, на основе результатов всех запусков вычисляются вероятностные характеристики рассматриваемого процесса. Например, чтобы вычислить методом Монте-Карло площадь S некоторой плоской фигуры, нужно нарисовать прямоугольник L, в который включена данная фигура, и по нему случайным образом разбросать большое количество точек. Если число точек N достаточно велико, то отношение площади S к площади прямоугольника L будет с заданной точностью равняться отношению количества точек M, попавших внутрь фигуры, к полному количеству точек N.

Между базовыми методами оценки значений находится спектр промежуточных, включая методы, основанные на пошаговых бэкапах и их комбинациях, таких как λ-резервные копии, реализованные с помощью трассировки приемлемости.

Задача обучения с подкреплением обычно может быть структурирована многими различными способами. Некоторые из них отражают естественные аспекты проблемы, такие как существование физических датчиков, а другие являются результатом явных попыток разложить проблему на более простые подзадачи. Независимость или почти независимость одних переменных от других иногда можно использовать для получения более эффективных специальных форм алгоритмов обучения с подкреплением. Иногда даже возможно разбить проблему на несколько независимых подзадач, которые могут решаться отдельными обучающими агентами.

Самая важная особенность, отличающая обучение с подкреплением от других видов обучения, заключается в том, что оно использует обучающую информацию, которая оценивает предпринятые действия, а не инструктирует, давая правильные действия. Именно это создает потребность в активном исследовании, в явном поиске хорошего поведения методом проб и ошибок.

Чисто оценочная обратная связь показывает, насколько хорошим было предпринятое действие, но не то, является ли оно лучшим или худшим из

возможных действий. Оценочная обратная связь лежит в основе методов оптимизации функций, в том числе эволюционных.

С другой стороны, чисто инструктивная обратная связь указывает на правильное действие, которое следует предпринять, независимо от того, какое действие фактически предпринято. Такая обратная связь лежит в основе обучения с учителем, которое включает в себя большую часть классификации образов, искусственных нейронных сетей и системной идентификации.

В чистом виде эти два вида обратной связи совершенно различны: оценочная обратная связь полностью зависит от предпринятого действия, тогда как инструктивная обратная связь не зависит от предпринятого действия. Есть также интересные промежуточные случаи, в которых оценка и обучение сливаются воедино.

Для повышения эффективности обучения с подкреплением можно также использовать методы интервальной оценки. В этом случае для каждого действия оценивается доверительный интервал его ценности. То есть вместо того, чтобы узнать, что ценность действия равна примерно 10, обучаемая система узнает, что она находится между 9 и 11, скажем, с 95%-й уверенностью. Выбранным действием будет действие, доверительный интервал которого имеет самый высокий верхний предел. Это поощряет исследование неопределенными действий, которые имеют шанс в конечном итоге стать лучшим действием.

Главной проблемой обучения с подкреплением является удачный выбор баланса между фазой обучения и фазой использования. Теоретические подходы к этой проблеме на сегодняшний день весьма узки, слабы и не обобщаются на все обучение с подкреплением.

Как отмечает сам Саттон [Sutton, 2015], «…мы предполагаем, что функции ценности, бэкап и GPI являются мощными организационными принципами, потенциально применимыми к любой модели интеллекта».

2.4.4. Сильный, слабый и гибридный ИИ

Определение, данное в стандарте [ГОСТ Р 59277–2020]:

искусственный интеллект (artificial intelligence): Комплекс технологических решений, позволяющий имитировать когнитивные функции человека (включая самообучение, поиск решений без заранее заданного алгоритма и достижение инсайта) и получать при выполнении конкретных практически значимых задач обработки данных результаты, сопоставимые, как минимум, с результатами интеллектуальной деятельности человека.

Это определение соответствует трактовке сильного ИИ (artificial general intelligence, strong AI, full AI, general intelligent action).

В отличие от него, слабый АИ (weak AI, narrow AI) не предназначен для имитации общих когнитивных способностей; скорее, слабый ИИ – это любая программа, предназначенная для решения ровно одной проблемы.

Сопоставляя вышеприведенное определение сильного ИИ [ГОСТ Р 59277-2020], с одной стороны, и определения когнитивных функций в трактовке психологов [Lezak, 1983] и ученых-когнитивистов [Kiely, 2014] (см. раздел 1.2), с

другой стороны, с результатами развития ИИ к настоящему времени (см. разделы 2.4.1–2.4.3), можно убедиться, что многие когнитивные функции человека уже воспроизводятся в ИИ или принципиально моделируются за счет подходов Deep Learning и Big Data.

Проблема достижения инсайта в сильном ИИ решается сейчас, главным образом, через формирование гибридного интеллекта (концепция «человек-в-контуре», human-in-the-loop, HITL) (рис. 2.42).

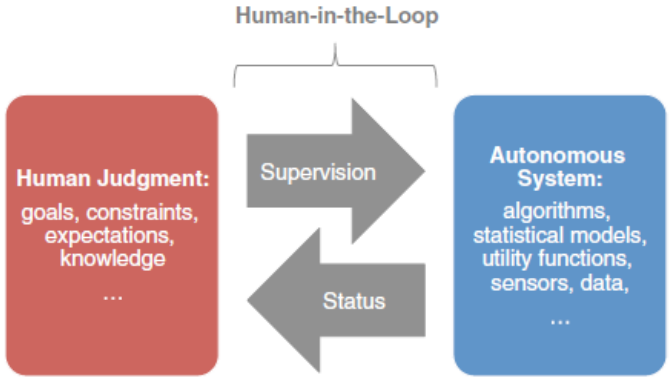


Рис. 2.42. Схема концепции «человек-в-контуре»

В системе HITL человек-оператор является важнейшим компонентом автоматизированного процесса управления, решающим сложные задачи наблюдения, контроля исключений, оптимизации и обслуживания.

Простейшая форма HITL в машинном обучении – это использование людей для разметки данных для обучения алгоритмов машинного обучения. Другим примером HITL является интерактивное машинное обучение, которое может помочь машинам учиться быстрее и эффективнее за счет интерактивной интеграции отзывов пользователей. Этот тип HITL существует уже некоторое время: многие компьютерные приложения учатся на поведении пользователей (например, предсказывая следующее слово, которое пользователь собирается ввести).

В настоящее время появляются более сложные примеры HITL, в которых человек в контуре имеет более подробные сведения о состоянии системы. Например, в системе кризисного консультирования система машинного обучения классифицирует сообщения, отправленные звонящими, и предоставляет визуализацию консультанту-человеку в режиме реального времени. Таким образом, человек и система машинного обучения работают в тандеме, чтобы обеспечить эффективное консультирование.

Подход HITL также успешно применяется к взаимодействию человека и робота. Сюда включаются динамическая адаптация степени автономии, предоставляемой роботам, интерактивное обучение роботов, обучающихся с подкреплением, принятие определенного поведения, а также создание гибких команд «человек–робот».

Подход HITL находит широкое применение в медицине. Примером может служить интеграция клинического диагностического процесса, выполняемого врачами, в медицинскую систему ИИ [Zheng, 2017] (рис. 2.43, 2.44).

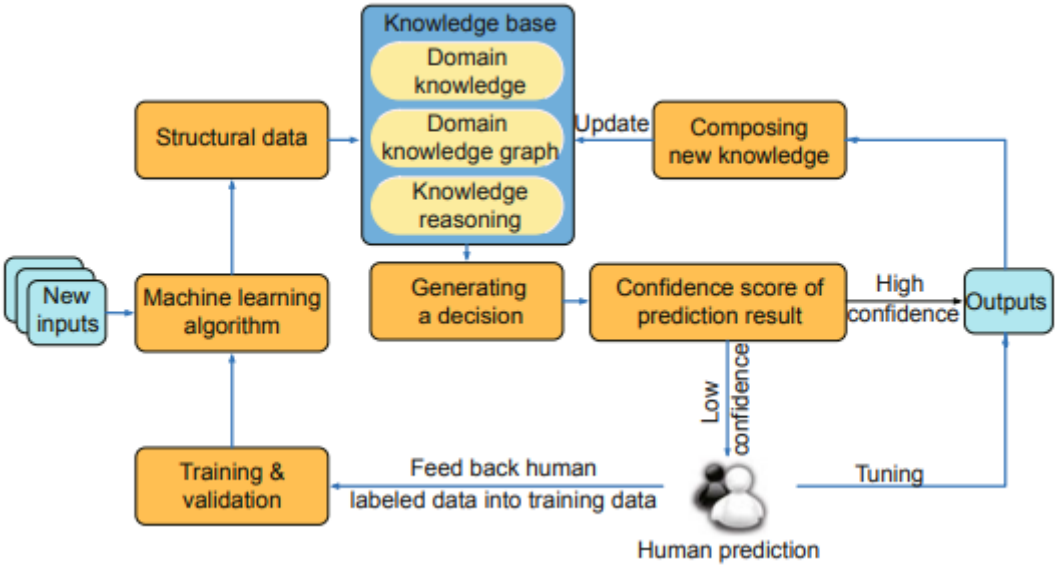


Рис. 2.43. Схема интеграции клинического диагностического процесса в медицинскую систему ИИ

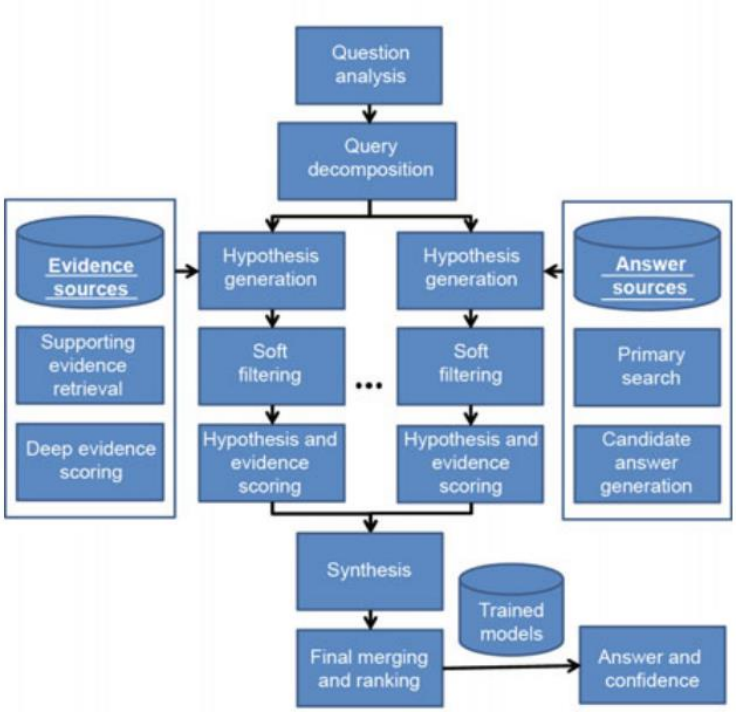


Рис. 2.44. Базовый фреймворк для интеграции клинического диагностического процесса в медицинскую систему ИИ

Человек выполняет две основные функции в системе HITL ИИ:

1. Человек может идентифицировать неправильное поведение с помощью автономной системы и предпринять корректирующие действия. Например, система компьютерного зрения на вооруженном беспилотнике может ошибочно идентифицировать гражданское лицо в качестве участника военных действий, а человек-оператор обеспечит выявление таких случаев и блокирует систему. Ведутся определенные работы, чтобы гарантировать, что ИИ не сможет научиться отключать собственный аварийный выключатель.

2. Человек может быть привлечен как подотчетная сущность в случае неправильного поведения системы. Если полностью автономная система причиняет вред людям, наличие человека в контуре дает уверенность в том, что кто-то возьмет на себя последствия таких ошибок и, таким образом, будет иметь стимул минимизировать эти ошибки. Этот человек может быть человеком в узком цикле управления (например, оператор дрона) или гораздо более медленном цикле (например, программисты в многолетнем цикле разработки автономного транспортного средства). Пока не найден способ наказать алгоритмы за причинение вреда людям, трудно придумать другую альтернативу.

Подведем некоторые итоги существующему положению дел в области сильного ИИ:

- Artificial Narrow Intelligence (ANI, Narrow AI) специализируется в одной области, решает одну проблему.
- Artificial General Intelligence (AGI, Strong AI) способен выполнять большинство задач, на которые способен человек.
- Artificial Super Intelligence (ASI) превосходит возможности интеллекта любого из людей, способен решать сложные задачи моментально.

На сегодняшний день успешно разрабатываются системы AGI, решающие достаточно широкий круг задач: детектирование, распознавание, перевод с одного языка на другой, генерация изображений, генерация текстов, но пока сложно говорить о применимости одной модели для выполнения существенно различающихся друг от друга задач без дополнительной перенастройки.

Вопросы для самопроверки

1. Перечислите стадии развития ИИ.
2. Какие основные недостатки имел планировщик STRIPS?
3. Какие фундаментальные ограничения имели первые системы ИИ (1970-е гг.)?
4. Дайте определение знаний (ГОСТ 33707–2016) и модели знаний (ГОСТ Р 57309–2016).
5. Что понимается в ИИ под термином «интеллектуальный агент»?
6. Каковы характеристики агентов в многоагентных системах?
7. Приведите примеры распространенных функций активации.
8. Какие логические функции могут реализовать двух – и трехслойная сеть нейронов?
9. Для чего в нейронной сети используется слой, реализующий функцию softmax?
10. Почему глубокие НС имеют бóльшую выразительную силу по сравнению с обычными НС?
11. Из каких типов слоев состоит сверточная НС?
12. Опишите архитектуру U-Net. Почему она эффективна в медицинских приложениях?
13. Какие механизмы, характерные для нейробиологии, реализуются современными технологиями глубокого обучения?

14. Перечислите характеристики «5V» больших данных.
15. Какие цели преследуются при обучении с подкреплением?
16. Дайте определение ИИ по ГОСТ Р 59277–2020.
17. Какова роль человека в концепции гибридного интеллекта («человек-в-контуре»)?

3. СТАНДАРТЫ И ТРЕБОВАНИЯ К СИСТЕМАМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

3.1. Обзор стандартов в сфере ИИ

В соответствии с Перспективной программой стандартизации по приоритетному направлению «Искусственный интеллект» на период 2021–2024 гг. [Программа, 2020] в России запланировано принятие 217 государственных стандартов, связанных с ИИ, девять из которых уже представлены в окончательной редакции [Технический, 2022]. Большая часть (шесть) из принятых стандартов относятся к системам ИИ в клинической медицине, что еще раз подчеркивает значении ИИ в этой области деятельности.

Общими для всех областей деятельности являются два стандарта, связанных с классификацией систем ИИ [ГОСТ Р 59277-2020] и со способами обеспечения доверия к системам ИИ [ГОСТ Р 59276-2020], обзор которых представлен ниже.

3.1.1. Классификации систем ИИ

Стандарт [ГОСТ Р 59277–2020] предлагает классифицировать системы ИИ по различным основаниям (рис. 3.1, табл. 3.1). Базовые классы СИИ группируются на основе следующих принципов:

1) по классам и категориям объектов в управлении;
2) по технологиям построения, приобретения и использования знаний;
3) по функциям, которые выполняет СИИ в контуре управления;
4) по методам и технологиям, используемым в СИИ;
5) по методам и средствам взаимодействия СИИ с другими системами и человеком-оператором.

Эти подходы к классификации являются основными. Каждый из них может иметь иерархическую структуру.

Дополнительные классификации могут быть связаны со специальными требованиями к объектам, процессам, контуру управления, архитектуре, ресурсам с учетом окружающей среды (интероперабельность. нормы регулирования, безопасность, действия стандартов, этические требования, надежность, отказоустойчивость, условия внешней среды и т.д.).

Классы можно характеризовать различными дополнительными аспектами или подклассами, например:

- наличием/отсутствием внешнего наблюдения, осуществляемого человеком-оператором либо другой автоматизированной системой;
- степенью понимания системы;
- степенью реактивности/отзывчивости;
- уровнем устойчивости функционирования;
- степенью надежности и безопасности;
- видом аппаратной реализации;
- степенью приспособляемости к внутренним или внешним изменениям;

– способностью оценивать свою собственную работоспособность/пригодность;
– способностью принимать решения и планировать.

Классификация может быть дополнена классами, приведенными в системах стандартизации.

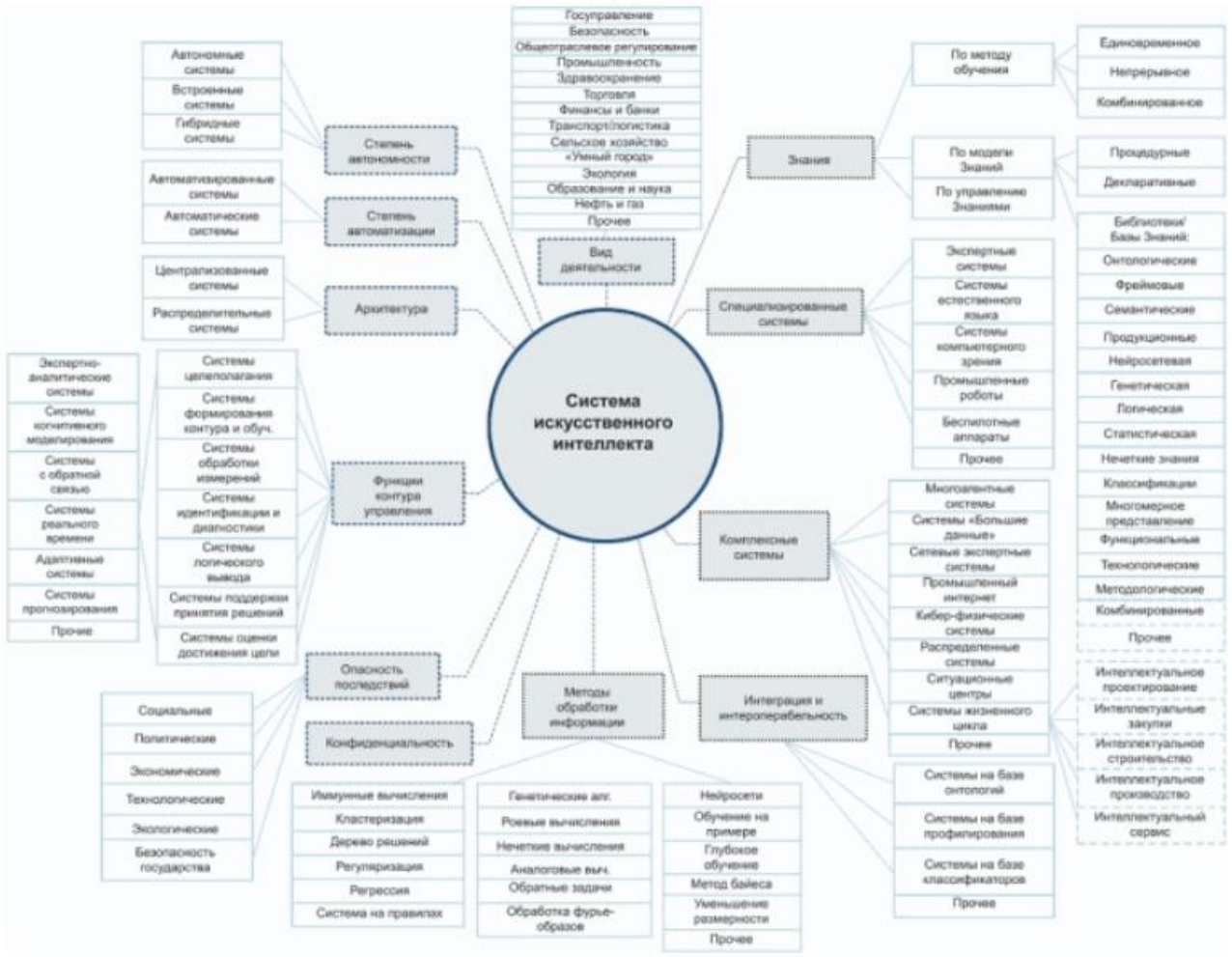


Рис. 3.1. Базовые классы СИИ

Таблица 3.1. Схема классификации систем искусственного интеллекта

Основания для классификации	Классы
1 По степени автономности	1.1 Автономные системы 1.2 Встроенные системы 1.3 Гибридные системы
2 По степени автоматизации	2.1 Автоматизированные системы 2.2 Автоматические системы
3 По архитектурному принципу	3.1 Централизованные системы 3.2 Распределенные системы
4 По видам деятельности	4.1 Государственное управление

	4.2 Безопасность 4.3 Общеотраслевое регулирование 4.4 Промышленность 4.5 Здравоохранение 4.6 Торговля 4.7 Финансы и банки 4.8 Транспорт и логистика 4.9 Сельское хозяйство 4.10 "Умный город" 4.11 Экология 4.12 Образование и наука 4.13 Нефть и газ 4.14 Прочее
5 По функциям контура управления	5.1 Системы с обратной связью 5.2 Системы реального времени 5.3 Адаптивные системы 5.4 Системы формирования цели (Системы целеполагания) 5.5 Системы формирования контура управления и обучения 5.6 Системы обработки измерений 5.7 Системы идентификации и диагностики 5.8 Системы когнитивного моделирования 5.9 Системы логического вывода 5.10 Системы принятия (поддержки) решений 5.11 Экспертно-аналитические системы 5.12 Системы оценки достижения цели 5.13 Ситуационные центры 5.14 Системы прогнозирования 5.15 Прочее
6 По специализации систем	6.1 Экспертные системы (управление знаниями) 6.2 Игровые системы 6.3 Системы естественного языка 6.4 Системы компьютерного зрения 6.5 Промышленные роботы 6.6 Беспилотные аппараты 6.7 Прочее
7 По комплексности и сложности систем	7.1 Многоагентные системы 7.2 Системы "Большие данные" 7.3 Промышленный интернет вещей 7.4 Киберфизические системы

	7.5 Системы жизненного цикла 7.6 Системы сетевой экспертизы 7.7 Распределенные системы управления 7.8 Система распределенных ситуационных центров 7.9 Прочее
8 По методам обработки информации	8.1 Нейросети 8.2 Обучение на примере 8.3 Эволюционные и генетические алгоритмы 8.4 Муравьиные алгоритмы 8.5 Иммунные вычисления 8.6 Глубокое обучение 8.7 Роевые вычисления 8.8 Метод Байеса 8.9 Уменьшение размерности 8.10 Природные вычисления 8.11 Мягкие вычисления 8.12 Кластеризация 8.13 Дерево решений 8.14 Регуляризация 8.15 Аналоговая обработка данных 8.16 Обработка фурье-образов 8.17 Регрессия 8.18 Решение обратных задач 8.19 Система правил 8.20 Прочее
9 По управлению знаниями, моделям и методам обучения	9.1 Процедурные 9.2 Декларативные 9.3 Онтологические 9.4 Семантические 9.5 Продукционные 9.6 Фреймовые 9.7 Нейросетевые 9.8 Генетические 9.9 Логические 9.10 Статистические 9.11 Нечеткие знания 9.12 Классификации 9.13 Многомерное представление (3Д, 4Д) 9.14 Функциональные 9.15 Технологические 9.16 Методологические 9.17 Комбинированное обучение

	9.18 Непрерывное обучение 9.19 Единовременное обучение 9.20 Прочее
10 По методам достижения интеграции и интероперабельности	10.1 Системы с интеграцией на базе онтологий 10.2 Системы на базе профилирования 10.3 Системы, использующие классификаторы 10.4 Прочее
11 По опасности последствий <*>	11.1 Социальная 11.2 Политическая 11.3 Экономическая 11.4 Технологическая 11.5 Техногенная 11.6 Экологическая 11.7 Безопасность государства
12 По конфиденциальности <**>	12.1 Уровень конфиденциальности (0–3)
<*> Классификация в соответствии с категорированием объектов критической информационной инфраструктуры: (1) социальной значимости (здоровье и жизнь людей); политической значимости (причинение ущерба государству); экономической значимости (ущерб субъектам и/или бюджетам); экологической значимости (воздействие на окружающую среду); значимость для обороны/безопасности, правопорядка. <**> Классификация соответствует следующим классам конфиденциальности: (0) Открытая информация; (1) Внутренняя информация; (2) Конфиденциальная информация; (3) Секретная информация.	

3.1.2. Обеспечение доверия к системам ИИ

Обеспечение доверия к системам ИИ в трактовке стандарта РФ. Стандарт [ГОСТ Р 59276–2020] следующим образом определяет доверие к системе ИИ:

доверие к системе искусственного интеллекта (Trustworthiness): Уверенность потребителя и, при необходимости, организаций, ответственных за регулирование вопросов создания и применения систем искусственного интеллекта, и иных заинтересованных сторон в том, что система способна выполнять возложенные на нее задачи с требуемым качеством.

Качество объекта (продукта или услуги) является комплексным показателем, определяющим потребительские свойства объекта **[**ГОСТ Р ИСО 9000–2015**]**. Понятие качества существует в конкретном прикладном контексте и может быть определено лишь для систем ИИ, обеспечивающих решение конкретных практически значимых задач. Приведенное определение качества не распространяется на системы «сильного» или «общего» ИИ, претендующих на

повторение естественных интеллектуальных способностей человека вне зависимости от решаемой прикладной задачи.

Существенные характеристики системы ИИ могут быть присущими или присвоенными, но при оценке качества учитываются только присущие [ГОСТ Р ИСО 9000–2015]. При этом существуют различные классы характеристик, такие как:

а) физические (например, механические, электрические, химические или биологические характеристики):

б) органолептические (например, связанные с запахом, осязанием, вкусом, зрением, слухом):

в) этические:

г) характеристики, связанные со временем;

д) эргономические;

е) функциональные (например, максимальная скорость движения беспилотного транспортного средства, вероятность ошибок первого и второго рода);

ж) информационной безопасности.

Представительный набор существенных характеристик определяется решаемой прикладной задачей и условиями применения системы ИИ. Например, функциональные характеристики для различных систем ИИ могут быть выбраны следующим образом:

- для систем распознавания речи – величина уровня пословной ошибки, фактор реального времени преобразования речи в текст на конкретном сервере обработки данных и другие характеристики;
- для систем дешифрирования аэрокосмических изображений – точность обнаружения объектов, полнота обнаружения объектов, точность локализации объекта, вероятность правильной классификации объекта и другие характеристики;
- для систем беспилотного управления транспортным средством уровней 1–5 – вероятность дорожно-транспортного происшествия при автономном управлении, уровень комфорта управления движением для пассажиров, отношение времени автономного управления к времени вынужденного перехода на ручное управление за все время движения и другие характеристики.

При выборе представительного набора существенных характеристик системы ИИ целесообразно руководствоваться следующими принципами [ГОСТ Р ИСО/МЭК 9126–93]:

- достаточность набора характеристик для принятия решения о возможности использования системы ИИ при решении конкретной прикладной задачи;
- простота и возможность измерения значений характеристик;
- отсутствие перекрытия между используемыми характеристиками;
- соответствие установившимся понятиям и терминологии;
- возможность последующего уточнения и детализации характеристик.

Существенные характеристики и субхарактеристики систем ИИ в зависимости от возможностей их измерения, представления и интерпретации могут быть

определены на различных шкалах, включая шкалу наименований, порядковую шкалу, интервальную шкалу и шкалу отношений.

На каждой стадии и каждом этапе жизненного цикла (ЖЦ) системы ИИ существуют факторы (причины), приводящие к снижению ее качества. Каждый фактор снижения качества связан с возможными отклонениями одной или нескольких существенных характеристик системы ИИ от установленных требований (табл. 3.2).

Таблица 3.2. Факторы снижения качества на стадиях создания и эксплуатации систем искусственного интеллекта

Стадия ЖЦ	Фактор снижения качества системы ИИ
Создание системы ИИ	
Концепция	Недостаточная полнота выбранного набора функциональных характеристик системы ИИ (прикладных характеристик, характеристик безопасности, надежности и других), не позволяющая считать выбранный набор характеристик представительным
Разработка	Недостаточная представительность обучающей выборки, использованной при создании системы ИИ. Смещенность обучающей выборки, способная привести к предвзятости (необъективности) результатов работы системы ИИ. Неоптимальность используемой модели данных. Недостаточный уровень унификации и низкая интероперабельность разрабатываемой системы
Производство	Недостаточная надежность создаваемой системы ИИ. Чрезмерная стоимость владения системой ИИ. Недостаточная понятность, обьяснимость, предсказуемость и др. Недостаточная защищенность информации о модели данных
Эксплуатация системы ИИ	
Применение по назначению	Применение системы ИИ не по назначению. Недостаточная представительность выборки, используемой при тестировании системы ИИ. Недостаточная периодичность тестирования системы ИИ. Отсутствие средств автоматического самотестирования после каждого обучения, дообучения системы ИИ. Недостаточная защищенность информации о функционировании системы ИИ. Недостаточная защищенность информации о модели данных, используемой в системе ИИ. Недостаточная защищенность обрабатываемых персональных данных
Поддержка	Утрата актуальности модели данных
Прекращение применения	Нарушение конфиденциальности персональных данных при выводе системы ИИ из эксплуатации

Факторы снижения качества могут быть связаны с естественными (непреднамеренное снижение качества) или искусственными (преднамеренное снижение качества) причинами. Примерами преднамеренного снижения качества, специфичными для систем ИИ, реализованных на НС, являются:

- на стадии создания системы – наличие преднамеренных искажений в обучающей выборке системы распознавания изображений, приводящих к ошибкам в работе системы распознавания, вызванным специальными, заранее определенными искажениями в исходных данных, включая «состязательные» атаки;
- на стадии эксплуатации системы – отсутствие достоверных и представительных оценок устойчивости системы распознавания изображений к воздействию преднамеренных «состязательных» атак, приводящее к неустойчивой работе системы в процессе ее эксплуатации.

Примерами непреднамеренного снижения качества систем ИИ являются:

- на стадии создания системы ИИ – использование статистически смещенной обучающей выборки, приводящей к появлению «предвзятостей» в результатах работы системы ИИ:
- на стадии эксплуатации системы – нарушение конфиденциальности обрабатываемых данных в условиях, когда уровень конфиденциальности данных существенно и неконтролируемо возрос в процессе эксплуатации системы ИИ вследствие накопления и обобщения информации.

Проверка доверия к системе ИИ обеспечивается подтверждением соответствия представительного набора существенных характеристик системы ИИ требованиям, установленным:

- разработчиком системы ИИ в том случае, если существует возможность подтверждения соответствия этим требованиям любой заинтересованной стороной, а не только самим разработчиком системы ИИ. Такие требования разработчика являются открытыми требованиями, в отличие от внутренних требований, подтверждение соответствия которым может быть выполнено только самим разработчиком системы ИИ;
- потребителем системы ИИ. Подтверждение соответствия потребительским требованиям осуществляется в процессе испытаний системы ИИ;
- организацией, ответственной за регулирование вопросов создания и применения систем ИИ в соответствии с принятыми национальными нормами (регулятором). Данные требования являются опциональными и, как правило, устанавливаются в том случае, если некорректная работа системы ИИ может привести к угрозам безопасности людей, окружающей природной среды, материальных и нематериальных активов. В этом случае подтверждение соответствия требованиям, установленным регулятором, осуществляется в ходе сертификации системы ИИ.

Процедура подтверждения доверия на разных стадиях жизненного цикла систем ИИ включает:

- выбор достаточного (представительного) набора существенных характеристик системы ИИ (разработчиком, потребителем и, при необходимости, организацией, ответственной за регулирование вопросов создания и

применения систем ИИ):

- установление требований к представительному набору существенных характеристик системы ИИ (открытых требований разработчика, потребительских требований и, при необходимости, требований безопасности, установленных организацией, ответственной за регулирование вопросов создания и применения систем ИИ);
- организацию процедур подтверждения качества системы ИИ, т.е. подтверждения соответствия представительного набора существенных характеристик системы ИИ установленным требованиям;
- реализацию мероприятий по обеспечению соответствия представительного набора существенных характеристик системы ИИ установленным требованиям (доведение качества системы ИИ до требуемого уровня за счет устранения причин, приводящих к снижению качества).

Способы обеспечения доверия, направленные на устранение факторов, приводящих к снижению качества систем ИИ, могут быть реализованы на последовательных стадиях ЖЦ разработчиками, потребителями систем ИИ пли третьей стороной (например, органом по сертификации) по инициативе разработчиков или потребителей систем (таблица 3.3).

Способы обеспечения доверия на стадии разработки заключаются в обеспечении соответствия открытым требованиям разработчика системы ИИ. Способы обеспечения доверия на стадии эксплуатации заключаются в обеспечении соответствия требованиям потребителя системы ИИ и требованиям организации, ответственной за регулирование вопросов создания и применения систем ИИ в соответствии с принятыми национальными нормами (опционально).

Таблица 3.3. Способы обеспечения доверия к системам ИИ на различных стадиях жизненного цикла

Фактор снижения качества системы ИИ	Способ обеспечения доверия к системам ИИ
Создание системы ИИ	
Недостаточная полнота выбранного перечня существенных характеристик системы ИИ (прикладных характеристик, характеристик безопасности, надежности и других)	Выбор представительного набора существенных характеристик системы и корректных правил их определения
Недостаточная представительность обучающей выборки, использованной при создании системы ИИ	Формирование представительной обучающей выборки (например, для биометрических систем ИИ) (ГОСТ Р 57194.1)
Статистическая смещенность обучающей выборки, способная привести к предвзятости (необъективности) результатов работы системы	Очистка набора данных различными способами. Статистический анализ наборов исходных данных и оценка их представительности и качества.

	Кросс-валидация выборки, полученной при разметке данных людьми. Непосредственная корректировка модели. Наложение ограничений на допустимую область применения системы ИИ
Неоптимальность используемой модели данных	Разработка оптимальной модели данных
Недостаточная надежность создаваемой системы ИИ. Чрезмерная стоимость владения системой ИИ	Использование рациональной ИТ-инфраструктуры, обеспечивающей надежную реализацию алгоритмов обработки данных при приемлемой стоимости владения системой ИИ
Недостаточная понятность, объяснимость, предсказуемость и др.	Использование интеллектуальных алгоритмов обработки данных, обеспечивающих принятие системой объяснимых, предсказуемых и т.д решений
Недостаточная защищенность информации о модели данных	Принятие эффективных мер по защите информации о модели данных на стадии разработки системы ИИ
Эксплуатация системы ИИ	
Применение системы ИИ не по назначению	Соблюдение допустимой области применения системы ИИ
Недостаточная представительность выборки, используемой при тестировании системы ИИ	Выбор представительной тестовой выборки при подтверждении соответствия системы установленным функциональным требованиям
Недостаточная защищенность информации о модели данных, используемой в системе ИИ. Недостаточная защищенность обрабатываемых персональных данных	Принятие эффективных мер по защите информации на стадии эксплуатации системы ИИ
Утрата актуальности модели данных	Своевременное выявление существенных отклонений в условиях эксплуатации системы ИИ и принятие мер по актуализации модели данных
Нарушение конфиденциальности персональных данных при выводе системы ИИ из эксплуатации	Принятие эффективных мер по защите персональных данных при выводе системы ИИ из эксплуатации

При создании и применении систем ИИ необходимо стремиться к применению всей совокупности способов обеспечения доверия, так как при этом обеспечивается устранение максимального количества факторов, приводящих к снижению качества работы системы. Способы обеспечения доверия на стадиях создания и

эксплуатации дополняют друг друга, что объясняется наличием преимуществ и недостатков, присущих способам различных типов (табл. 3.4).

Таблица 3.4. Особенности способов обеспечения доверия на разных стадиях жизненного цикла систем искусственного интеллекта

Тип характеристик	Преимущества	Недостатки
Способы обеспечения доверия на стадии создания системы ИИ	Позволяют заблаговременно (на этапах проектирования и разработки) гарантировать определенные свойства системы. Гарантируют высокую повторяемость свойств создаваемых систем	Предполагают достаточно полный доступ к процессу создания системы (основная часть требований разработчика должна быть открытой, то есть подлежащей проверке любой заинтересованной стороной), что может быть затруднительно для разработчика системы. Показательны для разработчика, но, как правило, неинформативны для потребителя системы
Способы обеспечения доверия на стадии эксплуатации системы ИИ	Хорошо интерпретируются в терминах потребительских свойств, более понятны потребителю системы	Результаты измерения характеристик не всегда могут быть экстраполированы на реальные условия эксплуатации системы (проблема представительности результатов тестирования)

Обеспечение доверия к системам ИИ в других трактовках. Хотя в мире в целом по терминологии наблюдается большое пересечение [Li, 2021], но более-менее усредненное понимание термина «доверительный АИ» в большой мере соотносят с этическими проблемами. Например, в [Li, 2021] дается следующее определение:

доверительный ИИ (Trustworthy AI, TAI) – термин, используемый для описания ИИ, который является законным, соблюдает этические нормы и технически надежен, причем эти требования должны соблюдаться на всем протяжении жизненного цикла AI продукта.

В [Schwartz, 2021] выделены следующие компоненты TAI.

- Конфиденциальность. Помимо обеспечения полной конфиденциальности пользователей, а также конфиденциальности данных, также необходимы механизмы управления доступом к данным.
- Робастность. Системы ИИ должны быть устойчивыми и безопасными. Они должны быть точными, способными обрабатывать исключения, хорошо работать с течением времени и быть воспроизводимыми. Еще одним важным аспектом является защита от угроз и атак со стороны противника.

- Объяснимость (Explainability, XAI). Понимание является важным аспектом в развитии доверия. Важно понимать, как системы ИИ принимают решения и какие функции были важны для процесса принятия каждого решения.
- Справедливость (Fairness). Системы ИИ должны быть справедливыми, беспристрастными и доступными для всех. Скрытые предубеждения в конвейере обработки системы ИИ могут привести к дискриминации и исключению недостаточно представленных или уязвимых групп.
- Прозрачность. Данные, системы и бизнес-модели, связанные с ИИ, должны быть прозрачными. Люди должны знать, когда они взаимодействуют с системой ИИ. Кроме того, возможности и ограничения системы ИИ должны быть понятны соответствующим заинтересованным сторонам и потенциальным пользователям.
- Оценка рисков. Это новая область с множеством приложений ИИ, и выявить соответствующие риски может быть сложно. В Руководстве по этике Европейского союза для надежного ИИ (The European Union's Ethics Guidelines for Trustworthy AI) представлен список оценок, который поможет компаниям определить риски, связанные с ИИ.
- Установление процедур. Процессы для надежного ИИ должны быть встроены в процессы управления компании. Разработка новых политик соответствия должна включать как технические меры по смягчению последствий, так и человеческий надзор.
- Сотрудничество человека и ИИ. Сотрудничество между различными дисциплинами, с различными группами заинтересованных сторон и экспертами в предметной области, затронутыми системой ИИ, имеет важное значение для определения того, какой тип системы, данных и объяснений полезен или необходим в данном контексте. Дальнейшее взаимодействие между политиками, исследователями и теми, кто внедряет системы с поддержкой ИИ, необходимо для создания надежной среды ИИ.
- Управление данными. В системах ИИ необходимо учитывать полную разработку и реализацию пайплайна. Это включает в себя рассмотрение того, как устанавливаются цели для системы, как обучается модель, какие меры защиты конфиденциальности и безопасности необходимы, какие большие данные используются и каковы последствия для конечного пользователя и общества. Объяснение того, какие обучающие данные и функции были выбраны для системы ИИ и являются ли они подходящими и репрезентативными для населения, может способствовать противодействию распространенным типам предвзятости ИИ.
- Мониторинг и контроль ИИ. Компании несут ответственность за разработку, развертывание и использование технологий и систем ИИ. Эти системы необходимо постоянно оценивать и контролировать, пока они выполняют свои задачи, чтобы гарантировать, что с течением времени не возникнут предубеждения. Использование дополнительных ресурсов (например, Google What-if Tool или IBM AI Fairness 360 Open Source Toolkit) для изучения и проверки моделей

может помочь в тестировании, отслеживании и документировании их разработки, упрощая их проверку и аудит.

- Сотрудничество с третьими сторонами. Помимо систем ИИ, разработанных собственными силами, бывают случаи, когда системы ИИ закупаются у внешних деловых партнеров. В таких случаях все вовлеченные стороны должны взять на себя обязательство обеспечить надежность системы и ее соответствие действующим законам и правилам. Процедуры аудита необходимо расширить, чтобы включить в них потенциальные риски и неблагоприятные последствия для конечных пользователей при разработке ИИ.
- Повышение осведомленности об этичном ИИ. Необходимо повысить осведомленность об этике ИИ всех сотрудников, работающих с ИИ. Риски и потенциальное влияние ИИ, а также способы снижения рисков должны быть разъяснены всем соответствующим заинтересованным сторонам – от руководителей компаний и специалистов по соблюдению законодательства до сотрудников, работающих с клиентами.

По каждому пункту этого огромного списка идет активная работа, предлагаются регламенты, фреймворки и отдельные решения. Мы рассмотрим более подробно объяснимый ИИ (explainable AI, XAI).

3.2. Подходы к построению объяснимого ИИ

3.2.1. Классификации средств объяснимого ИИ

Модели машинного обучения (такие как нейронные сети, машины опорных векторов, ансамбли решающих деревьев) являются прозрачными в том смысле, что все происходящие внутри них вычисления известны, но в то же время они рассматриваются как модели типа «черный ящик». Часто говорят, что модели машинного обучения плохо интерпретируемы. Здесь имеется в виду то, что процесс принятия решения не удается представить в понятной человеку форме, то есть:

- понять, какие признаки или свойства входных данных влияют на ответ;
- разложить алгоритм принятия решения на понятные составные части;
- объяснить смысл промежуточных результатов, если они есть;
- описать в текстовом виде алгоритм принятия решения (возможно, с привлечением схем или графиков).

Достичь полной интерпретируемости в машинном обучении, как правило, не удается, но даже частичная интерпретация может существенно помочь. Обзор способов интерпретации моделей машинного обучения можно найти, например, в [Linardatos, 2021] и [Li, 2021].

Чем же может помочь интерпретация модели?

Во-первых, интерпретировав алгоритм, мы можем открыть для себя что-то новое о свойствах исследуемых данных (например, какие признаки в табличных данных в наибольшей степени влияют на ответ) [Lundberg, 2019].

Во-вторых, интерпретация модели помогает оценить ее качество. Люди обычно имеют предварительные знания о предметной области, которые они

могут использовать, чтобы принять или отвергнуть какой-то прогноз, если они понимают причины, лежащие в его основе. Поэтому, узнав, на что именно обращает внимание модель, какими правилами руководствуется при предсказании, можно оценить правдоподобность этих правил [Ribeiro, 2016].

Средства XAI принято классифицировать с точки зрения объема области применения, методологии и использования [Das, 2020]:

1. Область применения: на чем фокусируется метод XAI – на локальном экземпляре или на понимании модели в целом?
   1.1. Локальный (local) XAI: основное внимание уделяется объяснению отдельных экземпляров x из имеющегося множества данных X. Генерирует одну карту объяснений g на данные $x \in X$.
   1.2. Глобальный (global) XAI: пытается понять модель в целом. Обычно требуется группа экземпляров данных для создания одной или нескольких карт пояснений.
   1.3. Смешанный вариант (both).

Пример. Пусть модель AI предназначена для определения тональности текстового документа. В этом случае локальный XAI может генерировать оценки атрибуции отдельных слов в тексте, а глобальный XAI даст представление о решении модели в целом, что приводит к пониманию атрибуций для массива входных данных.

2. Методология: на что ориентирован алгоритмический подход – на конкретный экземпляр входных данных или параметры объясняемой модели?
   2.1. На основе обратного распространения (backpropagation-based, BB): XAI оценивает градиенты, которые распространяются обратно из выходного слоя во входной слой. Примеры – карты значимости (saliency maps), карты релевантности значимости (saliency relevance maps) и карты активации класса (class activation maps).
   2.2. На основе возмущений (perturbation-based, PB): XAI оценивает действие случайных или направленных изменений конкретного экземпляра входных данных. Примеры – окклюзия, частичное замещение признаков с помощью операций заполнения или генеративных алгоритмов, маскирование, условная выборка.
   2.3. На основе ранжирования значимости признаков (ranging features).
3. Использование: как разработан метод XAI – как интегрированный в модель или как применимый к любой модели в целом?
   3.1. Интегрированный в модель (Intrinsic): объяснимость зависит от архитектуры нейронной сети и, как правило, не может быть перенесена на другие архитектуры.
   3.2. Апостериорный (post-hoc): метод XAI является моделенезависимым, т.е. не связан с архитектурой модели и направлен на объяснение результатов уже обученных нейронных сетей.

Реальные решения в области XAI содержат в себе все три группы классифицирующих признаков, поэтому классификационные схемы получаются очень сложными. Более упрощенная схема классификации, ориентированная на доминирующие подходы XAI в области медицины [Zhang, 2022] (рис. 3.2) содержит

только один классифицирующий признак (Intrinsic – Post-hoc), который признается всем авторами обзоров как базовый. Среди многочисленных классификационных признаков XAI дихотомия «метод XAI интегрирован в модель или является апостериорным, т.е. применим к любой модели в целом» (Intrinsic – Post-hoc) рассматривается всем авторами обзоров как базовая.

Интегрированные (Intrinsic) методы XAI предполагают, что объяснение зависит от архитектуры конкретной модели (например, нейронной сети) и, как правило, не может быть перенесено на другие архитектуры. Такие методы XAI имеют встроенные в модель интерпретируемые элементы, причем интерпретация возможна либо посредством следования строгим аксиомам и(или) правилам, либо посредством детальных объяснений решений.

Апостериорные (Post-hoc) методы XAI рассматривают объясняемую модель ИИ как черный ящик, т.е. не знают внутренних операций модели и не имеют доступ к архитектуре модели и структурам слоев. Поэтому алгоритм XAI будет работать с любой сетевой архитектурой. Это одно из основных преимуществ апостериорных методов XAI. Например, уже обученное и хорошо зарекомендовавшее себя решение нейронной сети можно объяснить, не жертвуя точностью обученной модели.

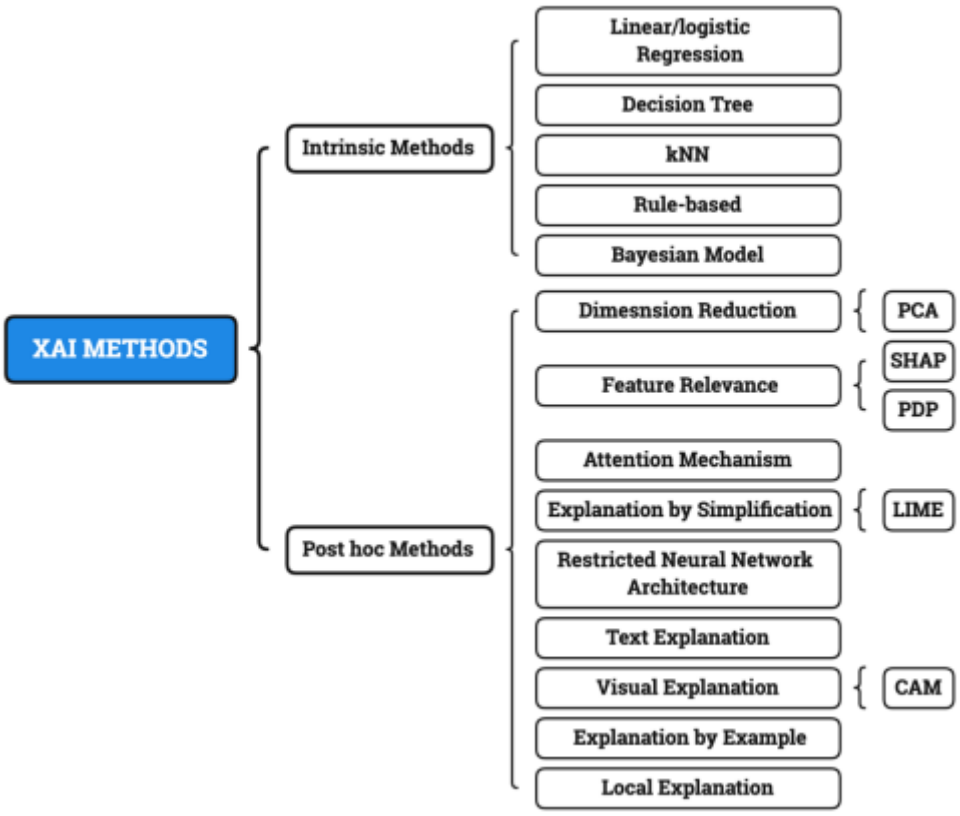


Рис. 3.2. Таксономия методов XAI в медицине с примерами

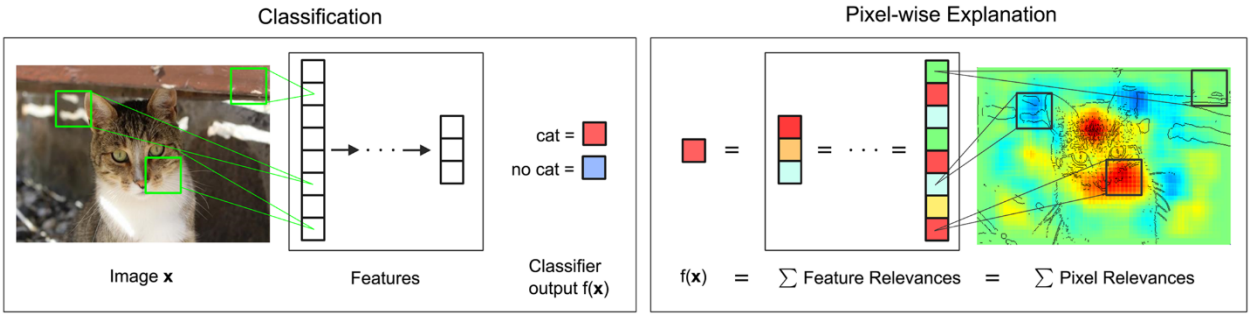
3.2.2. «Зоопарк» средств объяснимого ИИ

Карты значимости (Gradient-based Saliency Maps) в глубоких нейронных сетях [Simonyan, 2013] выделяют те пиксели входного изображения, которые оказали наибольшее влияние на результат, т.е. на отнесение объекта к

конкретному классу. Вычисляя градиенты score-функции в обратном ходе, получают сводку важности пикселей путем изучения положительных градиентов, которые оказали большее влияние на результат. Классификационные признаки метода – local, BB, post-hoc.

Рис. 3.3. Карты значимости классов для конкретных изображений с использованием атрибуции на основе градиента

Послойное обратное распространение релевантности (Layer-wise Relevance BackPropagation, LRP) [Bach, 2015] иллюстрируется на рис. 3.4. При классификации с помощью НС изображение преобразуется в представление вектора признаков, и применяется классификатор для присвоения изображению заданной категории, например «кошка» или «без кошки», при этом вычисление вектора признаков обычно включает использование нескольких промежуточных представлений (рис. 3.4, а). Метод LRP разлагает выходные данные классификации на суммы оценок релевантности функций и пикселей. Окончательные релевантности визуализируют вклад отдельных пикселей в прогноз (рис. 3.4, б). Классификационные признаки метода – both, BB, post-hoc.



а б

Рис. 3.4. Визуализация процесса попиксельной декомпозиции

Локальные интерпретируемые моделенезависимые объяснения (Local Interpretable Model-Agnostic Explanations, LIME) [Ribeiro, 2016]. Чтобы получить представление, понятное людям, LIME пытается найти важность смежных групп пикселей (патчей пикселей) в исходном изображении по отношению к выходному классу. Пример результата показан на рис. 3.5. Здесь тройку лучших

классов составляют «электрогитара» (p = 0.32), «акустическая гитара» (p = 0.24) и «лабрадор» (p = 0.21). Выбирая группу «суперпикселей» из входного изображения, классификатор предоставляет визуальные пояснения к предсказанным меткам наиболее вероятных классов. Классификационные признаки метода – both, perturbation-based, post-hoc.

Рис. 3.5. Локальные объяснения классификации изображений с помощью LIME

Псевдокод LIME представлен ниже.

```
Algorithm 1 LIME algorithm for local explanations
Input: classifier f, input sample x, number of superpixels n,
number of features to pick m
Output: explainable coefficients from the linear model
1: ŷ ← f:predict(x)
2: for i in n do
3:   pᵢ ← Permute(x)          Randomly pick superpixels
4:   obsᵢ ← f:predict(p)
5:   distᵢ ← |ŷ - obsᵢ|
6: end for
7: simscore ← SimilarityScore(dist)
8: x_pick ← Pick(p; simscore;m)
9: L ← LinearModel.fit(p; m; simscore)
10: return L.weights
```

В последние годы было проведено множество исследований, улучшающих и расширяющих алгоритм LIME для решения множества новых задач, в том числе:

- Sound-LIME – расширенный алгоритм LIME для анализа музыкального контента с помощью временной сегментации, а также частотной и частотно-временной сегментации входной мел-спектограммы [Mishra, 2017].
- В работе [Zafar, 2019] авторы использовали алгоритмы агломерационной иерархической кластеризации и K-ближайших соседей (KNN) для замены случайного возмущения, которое используется для построения групп пикселей в исходном алгоритме LIME. Здесь авторы используют иерархическую кластеризацию для группировки обучающих данных в виде кластеров, а KNN используется для поиска ближайших соседей к тестовому экземпляру. Как только KNN выбирает кластер, этот кластер передается как возмущение

входных данных вместо случайного возмущения, как в исходном алгоритме LIME. Тем самым достигается большая стабильность метода по сравнению с традиционным алгоритмом LIME.

- Отбор пикселей, группируемых в суперкластер, может также производиться по другим принципам – учитывая нелинейные отношения между пикселями [Bramhall, 2020] или рассматривая суперпиксель как клику в неорентированном графе [Shi, 2020].
- Предложено также обобщение алгоритма LIME на глобальную задачу, псевдокод которого представлен ниже.

Algorithm 3 LIME Algorithm for Global Explanations
Input: classifier f, input samples $x_1,\ldots,x_n \in X$
Output: explanation matrix after submodular pick
1: Define instances X and budget B
2: for $x \in X$ do
3: $f_{LIME} \leftarrow$ LIME(f; x)
4: end for
5: Select B features from f_{LIME}
Submodular Pick:
6: M $\leftarrow$ GenerateMatrix(X;B)
7: $X_{min} \leftarrow$ GreedyOptimization(M)

Метод LIME нашел широкое применение при интерпретации результатов ИИ в медицине (рис. 3.6).

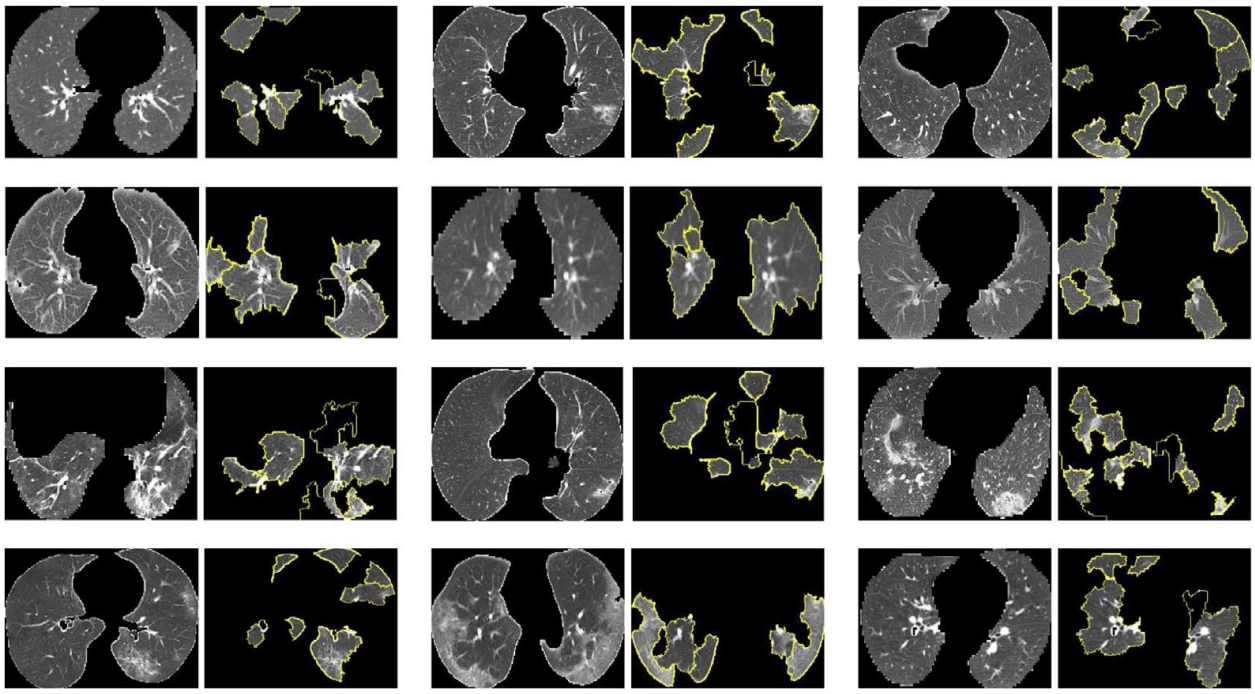


Рис. 3.6. Визуализация наиболее значимых для интерпретации зон КТ-изображений легких, пораженных COVID, методом LIME [Ribeiro, 2016]

Аддитивные объяснения Шепли (SHapley Additive exPlanations, SHAP). Эта группа подходов XAI базируется на методе нахождения справедливого

равновесия в кооперативных играх, за который его автор Lloyd Shapley получил Нобелевскую премию. Справедливость в методе Шепли определяется тем, насколько существенный вклад вносит каждый конкретный игрок в общий результат. Например, пусть имеется предприятие из N сотрудников, $N = \{o, w1, \ldots, wm\}$, где собственник, o, обеспечивает решающий капитал в том смысле, что без него нельзя получить никакой прибыли, а каждый из m рабочих ($w1$, ..., wm) вносит в общую прибыль сумму p. Тогда, согласно методу Шепли, справедливым является следующее распределение прибыли: $mp/2$ собственнику и по $p/2$ каждому рабочему. Расчет базируется на введении вектора Шепли, удовлетворяющего определенному набору аксиом, который эффективно применим в самых различных предметных областях.

В работе [Lundberg, 2017] было предложено использовать подход Шепли для вычисления вклада отдельных признаков (features) входных данных x в выходной прогноз. В методе SHAP признаками данных могут быть отдельные категории в табличных данных или группы суперпикселей в изображениях. Классификационные признаки метода – both, perturbation-based, post-hoc.

Пример применения SHAP в медицинских приложениях ИИ [Chan, 2022] представлен на рис. 3.7. Цвета представляют значения числовых функций: красный для больших значений и синий для меньших. Толщина линий в каждой точке определяется количеством примеров при заданном значении. Отрицательное значение SHAP (распространяющееся влево) указывает на снижение риска смертности, а положительное (распространяющееся вправо) – на повышенный риск смертности.

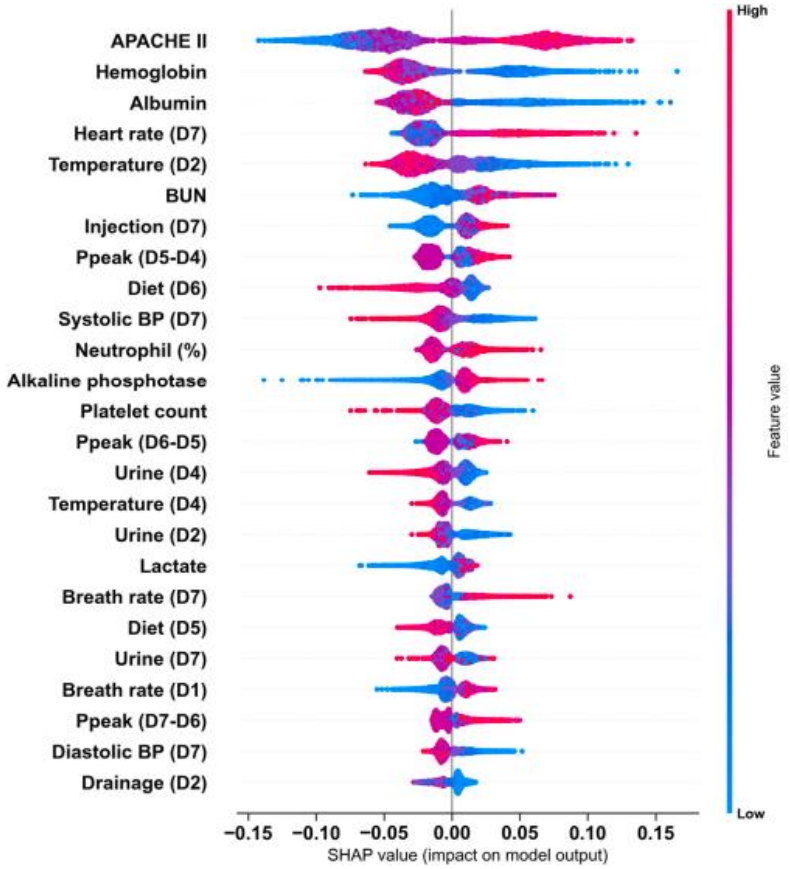


Рис. 3.7. Демонстрация влияния отдельных признаков на прогнозирование смертности методом SHAP

Метод эффективен для сравнительно небольшого набора признаков. Он широко применяется для перечислимого списка признаков (например, для сравнительной оценки важности диагностических признаков пациента, как на рис. 3.6); в то же время при интерпретации изображений (рис. 3.7) (рисунок из [Yang, 2022]) SHAP оказывается слишком грубым по сравнению с LIME (рис. 3.6) и CAM (рис. 3.11)

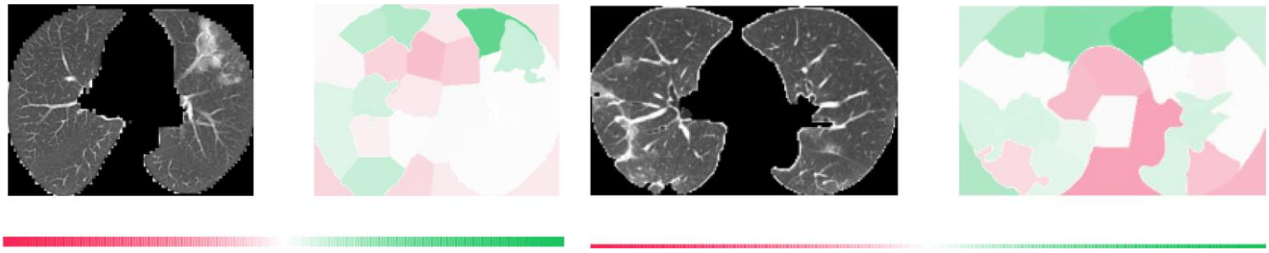


Рис. 3.8. Визуализация наиболее значимых для интерпретации зон КТ-изображений легких, пораженных COVID, методом SHAP

Визуализация признакового пространства методом t-SNE. Выделение наиболее значимых признаков, кроме метода SHAP, может производиться другими способами снижения размерности признакового пространства (dimension reduction). Кроме традиционного анализа главных компонент (PCA), с этой целью в последнее время широко применяется анализ метод t-SNE (t-distributed Stochastic Neighbor Embedding) [van der Maaten, 2008].

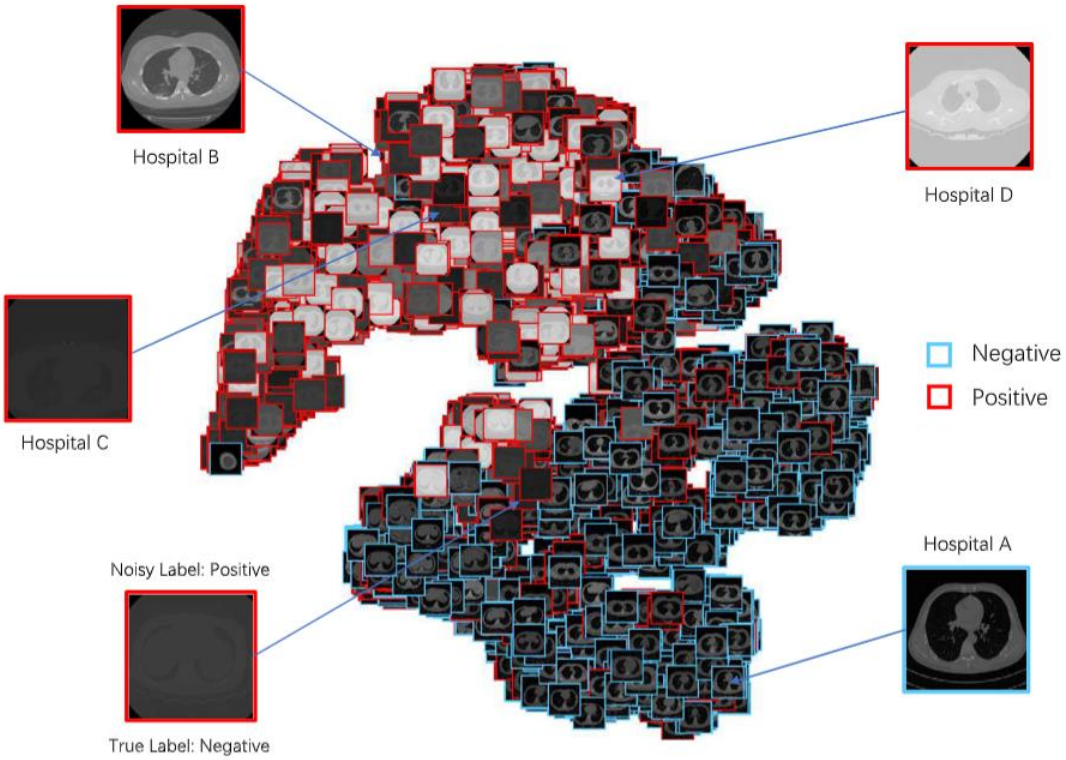


Рис. 3.9. Визуализация методом t-SNE КТ-изображений легких, пораженных COVID

В отличие от линейной кластеризации, метод t-SNE осуществляет вложение данных высокой размерности в пространство низкой размерности, что создает удобную визуализацию: каждый объект высокой размерности моделируется двух- или трёхмерной точкой таким образом, что похожие объекты с большой вероятностью соответствуют близко расположенным точкам, а непохожие объекты – точками, далеко друг от друга отстоящими.

На рис. 3.9 показано распределение классифицированных КТ-изображений легких, пораженных COVID, построенное методом двумерного t-SNE. Исходные изображения взяты из четырех разных больниц (A, B, C и D). На этом рисунке можно найти отличительные визуальные характеристики КТ-изображений из разных больниц. Кроме того, можно заметить, что изображения, соответствующие реальному заболеванию COVID, сгруппированы в одном кластере, а отрицательные изображения – в другом кластере. Более того, в «отрицательном» кластере расположено ложно аннотированное изображение.

Для выделения наиболее значимых элементов входных последовательностей могут также использоваться механизмы внимания (см. раздел 2.4.1). Тем самым можно выявить определенные позиции в последовательности (например, время, номер посещения больного и т.п.), которые наиболее сильно влияют на результат прогнозирования.

Обобщенный метод максимизации активации (Activation maximization) [Erhan, 2010], по существу, представляет собой усреднение карт значимости (Gradient-based Saliency Maps) в глубоких нейронных сетях по целевым классам изображений. Метод позволяет понять, насколько обобщаема конкретная модель ИИ. Классификационные признаки метода – local, backpropagation-based, post-hoc.

На рис. 3.10 показана визуализация моделей целевых классов изображений (гусь, страус и лимузин) для конкретной НС, построенная численно с использованием метода Activation maximization. Визуализация позволяет понять, какие именно признаки изображений целевого класса являются базовыми для классифицирующей системы ИИ.

Рис. 3.10. Визуализация «обобщенной» модели целевого класса методом Activation maximization

Тестирование с помощью векторов активации концептов (Concept Activation Vectors, CAVs) [Kim, 2018]. Метод основан на выделении во входном

датасете понятных человеку различительных концептов. Например, для зебры такой характерный концепт – полосы (рис. 3.11).

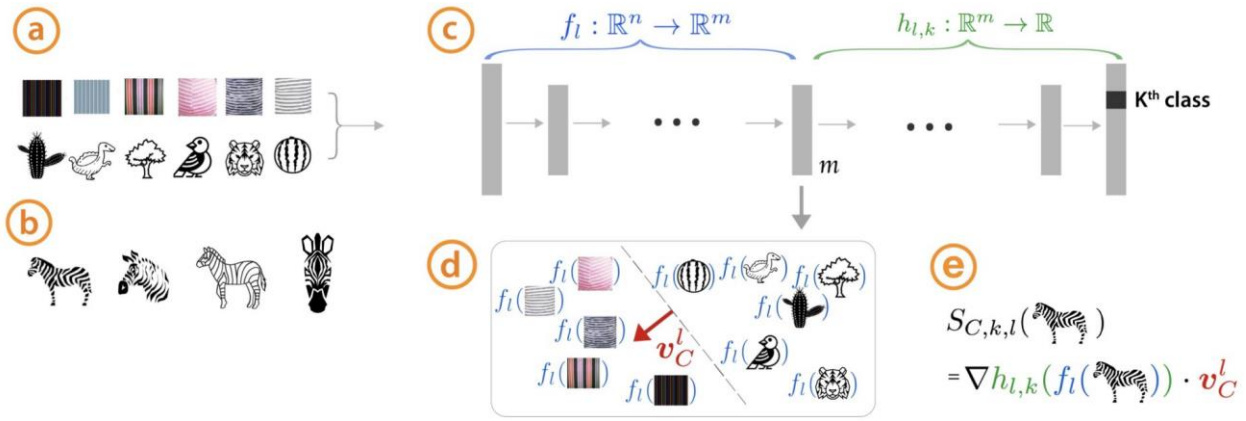


Рис. 3.11. Метод CAV: (a) случайные примеры и их концепты, (б) помеченные примеры из обучающих данных, (c) обученная нейронная сеть, (d) линейная модель, разделяющая активации, извлеченные из определенных слоев в нейронной сети для желаемых примеров и случайных примеров, (e) поиск с использованием производных по направлению

Характерные концепты генерируются либо из входных признаков обучающих данных, либо из предоставленных пользователем данных. Тогда можно оценить чувствительность прогнозов класса *f* к изменениям заданных входных данных путем расчета производных по направлению (аналогично методам на основе градиента) в сторону изменения концепта *C* для определенного слоя *j*. Классификационные признаки метода CAV – global, other, post-hoc.

Методы на основе окклюзии (occlusion-based methods). Целый ряд методов XAI используют следующую гипотезу: если у нас имеется одна и та же группа пикселей информации в одной и той же позиции всех объектов датасета, то модель XAI при оценке относительной важности должна ее игнорировать; следовательно, методы XAI, которые фокусируются на таких группах пикселей, являются плохими.

Группа методов, реализующих эту гипотезу, основана на работе [Zeiler, 2014], в которой объект на входном изображении заменялся на серое пятно (заградительный патч) и оценивалось, насколько эффективной остается при этом работа классификатора. На рис. 3.12 [Zeiler, 2014] показаны различные варианты оценки качества классификатора (нейронной сети) посредством окклюзии.

Пример в первой строке показывает, что самым важным признаком для распознавания собаки является ее морда. Когда она скрыта, активность на карте признаков уменьшается (синяя область в столбце (b)), а вероятность классификации изображения как «померанского шпица» значительно падает (синяя область в столбце (d)), и вместо него классифицируется «теннисный мяч» (e). Во втором примере текст на автомобиле – самый важный признак (b), но классификатор наиболее чувствителен к колесу. Третий пример содержит несколько объектов. Самые важные признаки – лица (b), но классификатор чувствителен к собаке (синяя область в (d)), так как он использует несколько карт признаков.

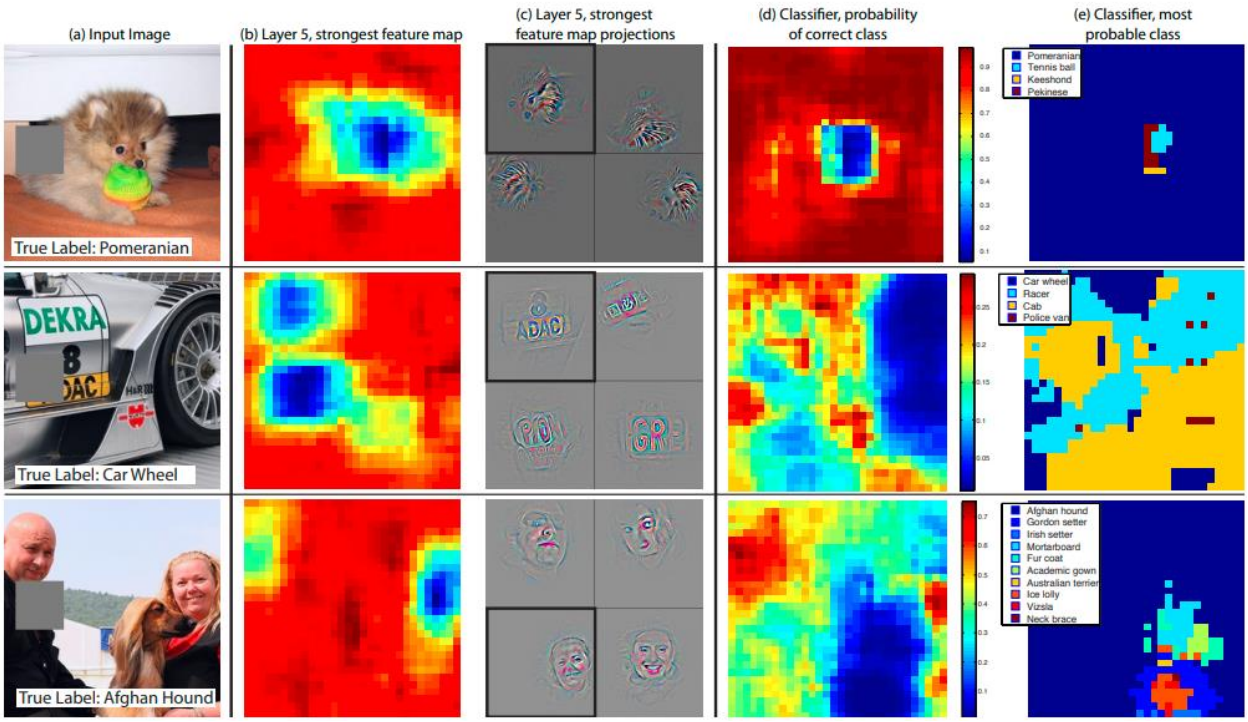


Рис. 3.12. Применение окклюзии для оценки качества классификатора

Работа [Ancona, 2018] иллюстрирует влияние размера заградительных патчей на интерпретируемость модели (рис. 3.13), а также демонстрирует разницу подходов метода окклюзий и градиентных методов (рис. 3.14).

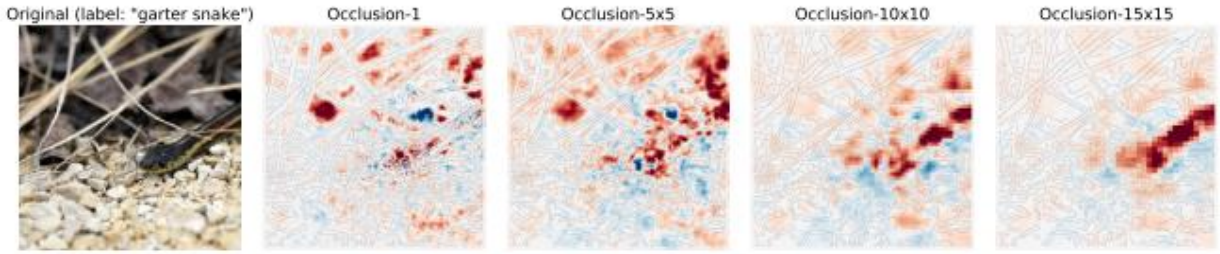


Рис. 3.13. Атрибуции, созданные путем закрытия частей входного изображения прямоугольными серыми пятнами разного размера

Обратите внимание, как размер плашек влияет на результат, фокусируясь на главном объекте только при использовании плашек большего размера.

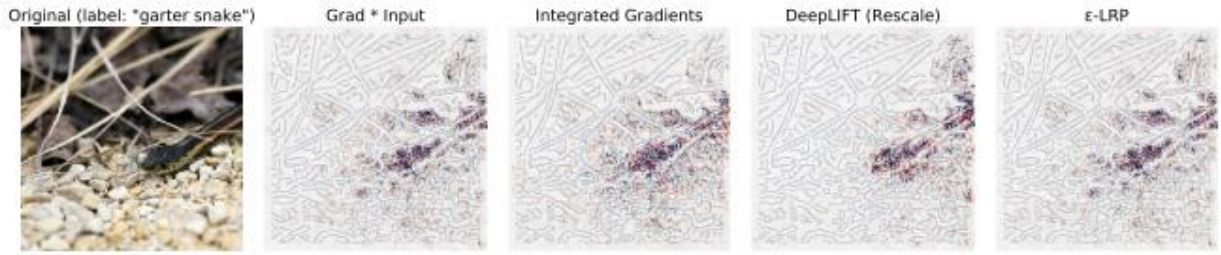


Рис. 3.14. Сравнение разных методов атрибуции при классификации изображений

Все методы на основе градиента выделяют атрибуты, на которые влияет более высокая локальная дисперсия, по сравнению с методами на основе возмущений (рис. 3.12).

Рандомизированная выборка входных данных для объяснения (Randomized Input Sampling for Explanation, RISE) [Petsiuk, 2018] является развитием метода окклюзий. Схема метода RISE показана на рис. 3.15.

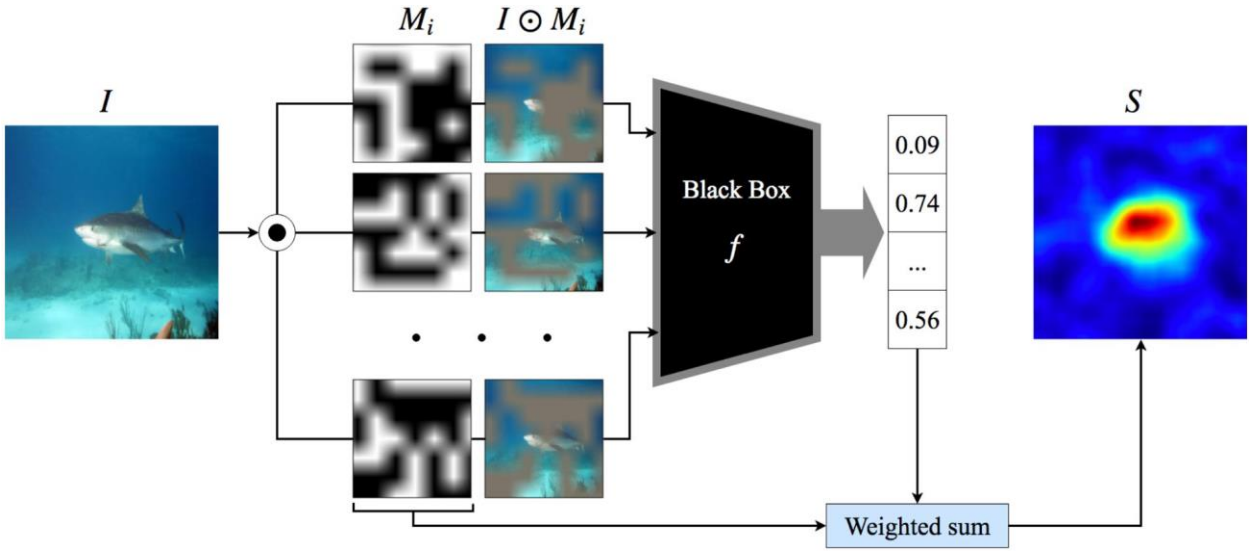


Рис. 3.15. Схема метода RISE

Входное изображение возмущается путем его умножения на рандомизированные маски. Маскированные изображения используются в качестве входных данных, и фиксируются соответствующие им карты значимости. Для нахождения окончательной карты значимости используется средневзвешенное значение масок на основе метода Монте-Карло (блок черного ящика) плюс строится тепловая карта важности индивидуальных прогнозов.

Классификационные признаки метода CAV – local, permutation-based, post-hoc.

Отображение активации классов (Class Activation Mapping, CAM). Значительно более понятными для конечного пользователя (по сравнению с методами на основе возмущений) являются градиентные методы, куда входят уже упомянутые карты значимости (Gradient-based Saliency Maps, рис. 3.3), а также широко распространенная группа методов Class Activation Mapping (CAM) [Zhou, 2016]. Методы на основе CAM фиксируют карты активации каждой группы слоев НС и затем производят их линейную комбинацию, тем самым формируя карту визуального объяснения работы данной модели по данному входному изображению (рис. 3.16).

Отдельные модификации метода CAM в основном связаны с тем, как именно выполняется комбинация отдельных карт активации.

- Grad-CAM [Selvaraju, 2017] определяет коэффициент конкретной карты активации путем усреднения градиентов по всем активационным нейронам на этой карте.
- Grad-CAM++ [Chattopadhay, 2018] является модифицированной версией Grad-CAM, которая фокусируется на положительном влиянии нейронов с учетом производных более высокого порядка.

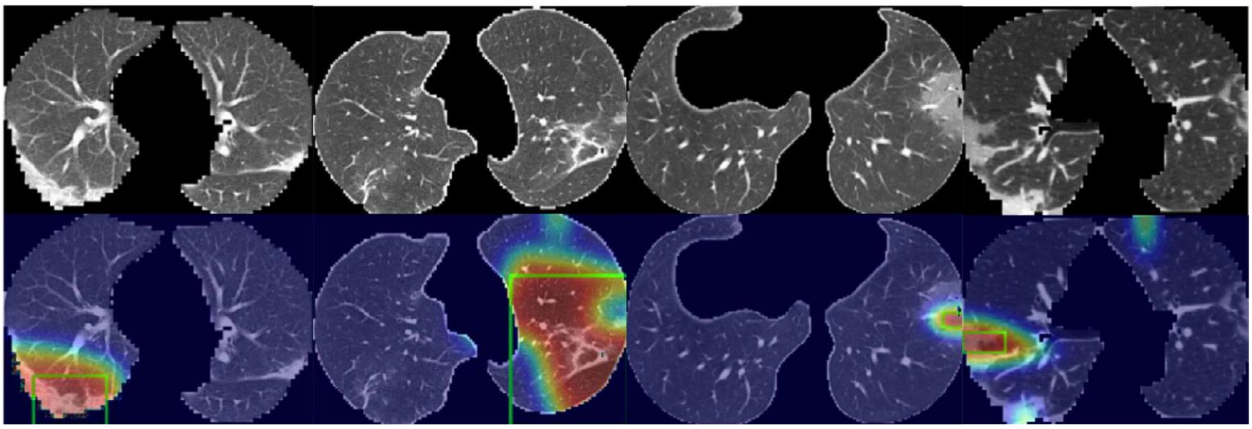


Рис. 3.16. Визуализация наиболее значимых для интерпретации зон КТ-изображений легких, пораженных COVID, методом CAM [Ribeiro, 2016]

Однако градиенты глубоких нейронных сетей имеют тенденцию к уменьшению из-за проблемы насыщения градиента, и использование немодифицированных градиентов приводит к сбою локализации соответствующих областей. Чтобы преодолеть это ограничение, были предложены безградиентные CAM.

- Score-CAM [Wang, 2020] накладывает нормализованные карты активации на входное изображение и делает прогнозы для получения коэффициентов.
- Ablation-CAM [Desai, 2020] определяет коэффициент как долю снижения целевого результата при удалении связанной карты активации. Метод не имеет проблемы насыщения, но требует большого времени выполнения.

Все описанные выше методы определяют свои коэффициенты для соотношения влияния отдельных карт активации эвристическим путем. Однако их также можно определять, используя моделенезависимые наборы аксиом [Lundberg, 2017], в той или иной мере соответствующие принципам Шепли. Это такие методы, как:

- XGrad-CAM [Fu, 2020];
- LIFT-CAM [Jung, 2021].

Сравнение работы различных методов CAM представлено на рис. 3.17.

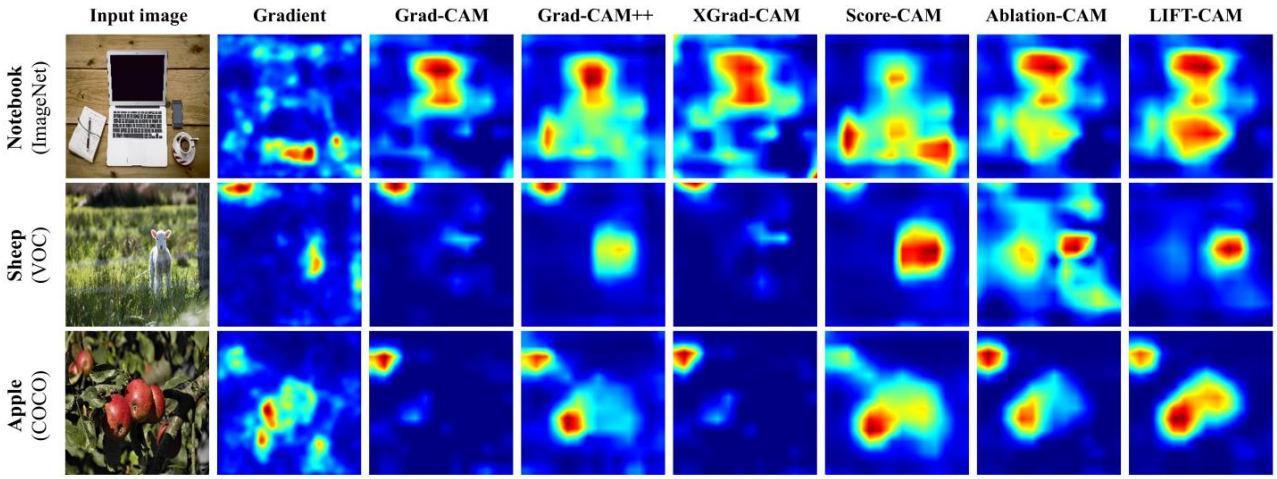


Рис. 3.17. Сравнение различных методов CAM (рисунок из [Jung, 2021])

Список байесовских правил (Bayes Rule List, BRL). Многие алгоритмы XAI, включая LIME и SHAP, используют линейные или древовидные модели для глобальных расширений основных алгоритмов. Деревья и модели на основе правил – неглубокие модели XAI, которые считаются интерпретируемыми в силу своей логико-математической сути. Однако произвольная древовидная модель не может гарантировать полное покрытие правилами всей предметной области, что потенциально снижает точность ее объяснений.

В [Letham, 2015] представлена генеративная модель, формирующая списки байесовских правил (BRL), которая дает апостериорное распределение по возможным спискам решений для улучшения интерпретируемости при сохранении точности. Список правил содержит правила if, else и elseif, обобщенные как антецедент правила IF-THEN, и прогнозы. По мере добавления в список решений большего набора правил IF-THEN модель становится более точной и интерпретируемой.

Выбирая образец списка правил из априорного распределения, итеративно добавляя и редактируя их, BRL пытается оптимизировать правила таким образом, чтобы новое распределение правил соответствовало апостериорному распределению. После оптимизации новые правила можно выбирать из апостериорного распределения (рис. 3.18).

Классификационные признаки метода BRL – global, other, intrinsic.

При большом количестве условий объяснение становится все более сложным для восприятия человеком. Один из способов упростить задачу – выделить часто встречающиеся шаблоны правил и изучить список решений из распределения с помощью байесовских методов. Исследования [Yang, 2017] улучшили масштабируемость BRL за счет уточнения теоретических границ, повторного использования вычислений и адаптации языковых библиотек.

if hemiplegia **and** age >60 **then** *stroke risk* 58.9% (53.8%–63.8%)
else if cerebrovascular disorder **then** *stroke risk* 47.8% (44.8%–50.7%)
else if transient ischaemic attack **then** *stroke risk* 23.8% (19.5%–28.4%)
else if occlusion and stenosis of carotid artery without infarction **then** *stroke risk* 15.8% (12.2%–19.6%)
else if altered state of consciousness **and** age >60 **then** *stroke risk* 16.0% (12.2%–20.2%)
else if age ≤ 70 **then** *stroke risk* 4.6% (3.9%–5.4%)
else *stroke risk* 8.7% (7.9%–9.6%)

Рис. 3.18. Список байесовских правил для определения риска инсульта в течение одного года после установления диагноза фибрилляции предсердий на основании истории болезни пациента. Приведенный риск – среднее значение апостериорного последовательного распределения; в скобках – 95-% доверительный интервал

Обобщенные аддитивные модели (GAM) [Caruana, 2015] хорошо подходят для объектов, которые описываются вектором признаков сравнительно низкой размерности. В этом случае рассчитывается степень влияния конкретного признака на конечный результат:

$$g\left(E\left[y\right]\right)=\beta_0+\sum f_j\left(x_j\right).$$

Классификационные признаки метода GAM – global, other, intrinsic.

Для повышения точности к стандартным GAM можно добавить парные взаимодействия, что приведет к модели под названием GA^{2}M. GA^{2}M сначала строит лучший GAM, а затем обнаруживает и ранжирует все возможные пары взаимодействий в остатках. Затем в модель включаются лучшие k пар (k определяется перекрестной проверкой).

$$g\left(E\left[y\right]\right)=\beta_0+\sum_j f_j\left(x_j\right)+\sum_{i\neq j} f_{i,j}\left(x_j,x_j\right).$$

На рис. 3.19, а, б приведены примеры применения метода GAM к оценке влияния отдельных переменных (мочевина и сахар в крови соответственно) на риск смерти от пневмонии. По вертикальной оси откладывается значение риска, по горизонтальной – величина соответствующей переменной, зеленые линии – погрешности. Для интерпретации попарных взаимодействий их можно визуализировать в виде тепловой карты (рис. 3.19, в).

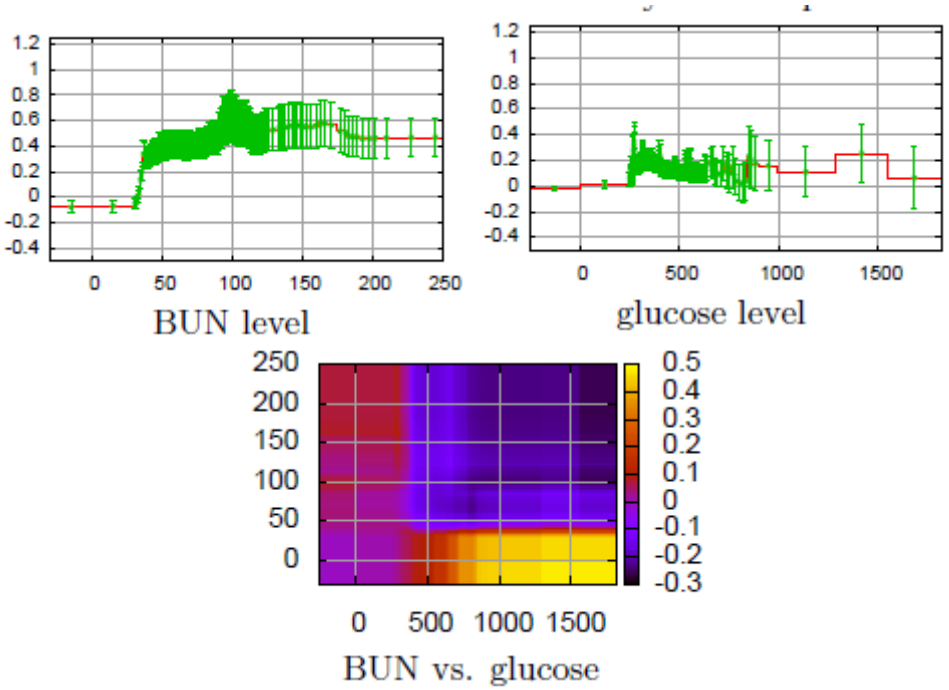


Рис. 3.19. Оценка методом GAM влияния отдельных показателей крови (а – мочевина, б – сахар) и их попарного взаимодействия (в) на риск смерти от пневмонии

В исходной статье [Caruana, 2015] форма получаемых зависимостей аппроксимировалась сплайнами, что приводит к проблеме оптимизации, которая уравновешивает гладкость сплайнов и эмпирическую ошибку. Но чрезмерная регуляризация снижает точность моделей GAM, которые подгоняются с использованием сплайнов. Поэтому многочисленные методы улучшили GAM. Среди них

интересны нейронно-аддитивные модели [Agarwal, 2021], в которых для оценки коэффициентов влияния отдельных признаков используются самостоятельные НС (рис. 3.20).

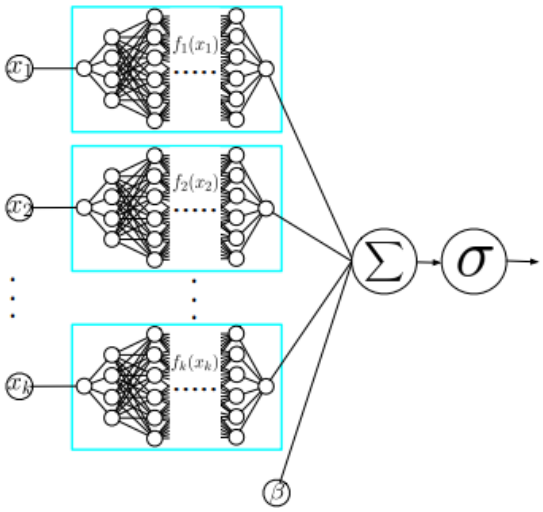


Рис. 3.20. Оценка коэффициентов влияния признаков в методе GAM посредством отдельных нейронных сетей

Существует целый ряд ограничений на использование интегрированных методов XAI, поскольку они требуют тщательной разработки алгоритмов и тонкой настройки для постановки задачи. Тем не менее, пока не превышен разумный предел производительности, встроенная архитектура моделей для XAI может помочь ускорить построение интерпретируемых по своей сути модели для будущих исследований ИИ.

Сети деконволюции (DeConvolution Nets, DCN) [Zeiler, 2014].

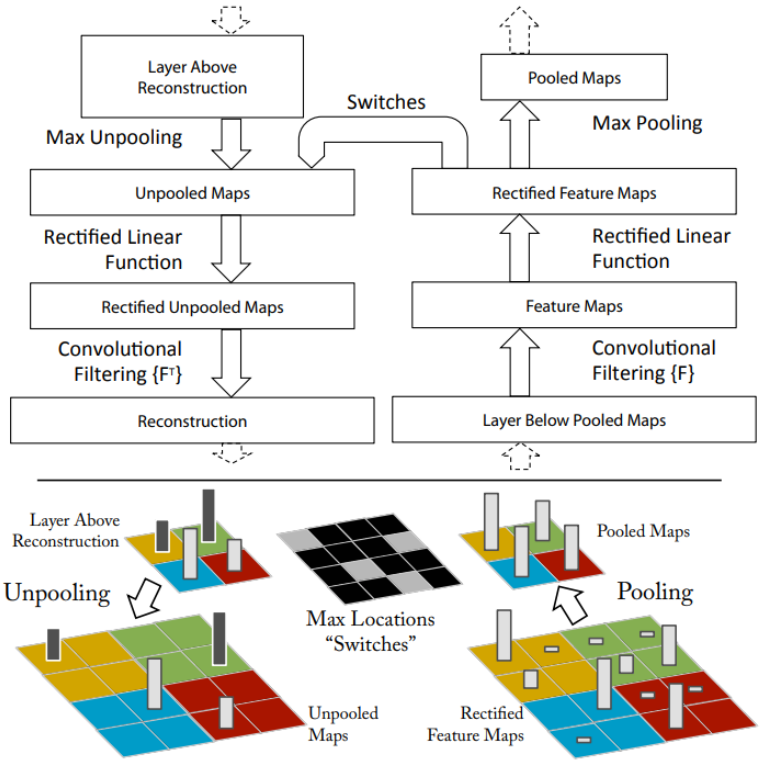


Рис. 3.21. Вверху: слой сети деконволюции (слева), прикрепленный к слою сверточной сети (справа). Внизу: иллюстрация операции распаковки

Сеть деконволюции (рис. 3.21) присоединяется к каждому сверточному слою исследуемой нейронной сети и позволяет проецировать функции активации (выборочно или вместе) обратно в пространство входных пикселей. Тем самым сеть деконволюции реконструирует приблизительную версию объектов сверточной сети из нижележащего слоя. При распаковке в сети деконволюции используются коммутаторы, которые записывают местоположение локального максимума в каждой области объединения (цветные зоны) во время объединения в сверточной сети.

Метод DCN может решать разнообразные задачи, например, выделять элементы изображения, наиболее значимые для его классификации, прослеживать процесс формирования все более обобщенного признака и.т.д. Классификационные признаки метода DCN – local, backpropagation-based, post-hoc.

3.2.3. Сравнение методов объяснимого ИИ

Существующее положение и подходы к оценке средств XAI. Обзоры методов XAI показывают, что эта область все еще не является зрелой, и основным подходом служит оценка с участием человека, т.е. в концепции human-in-the-loop. Количественные и обобщенные схемы сравнительной оценки различных методов XAI, равно как и эффективности каждого отдельного метода, еще предстоит разработать. Объяснимость наиболее интересных и современных алгоритмов машинного обучения и глубокого обучения оставляет желать лучшего (рис. 3.22).

Рис. 3.22. Сравнительная оценка объяснимости и производительности модели ИИ для разных алгоритмов машинного обучения [Yang, 2022]

Однако определенные шаги в этом направлении уже сделаны [Zhou, 2021]. Прежде всего, сформулированы общие требования к бенчмаркингу методов XAI [Doshi-Velez, 2017; Elshawi, 2019; Miller, 2019]. Например, идеальное решение должно иметь как высокую объяснимость, так и высокую производительность (рис. 3.19). Однако существующие линейные модели, модели на основе правил и деревья решений более прозрачны, но в целом имеют более низкую производительность. Напротив, сложные модели, такие как глубокое обучение и ансамбли, демонстрируют более высокую производительность, но меньшие возможности объяснения.

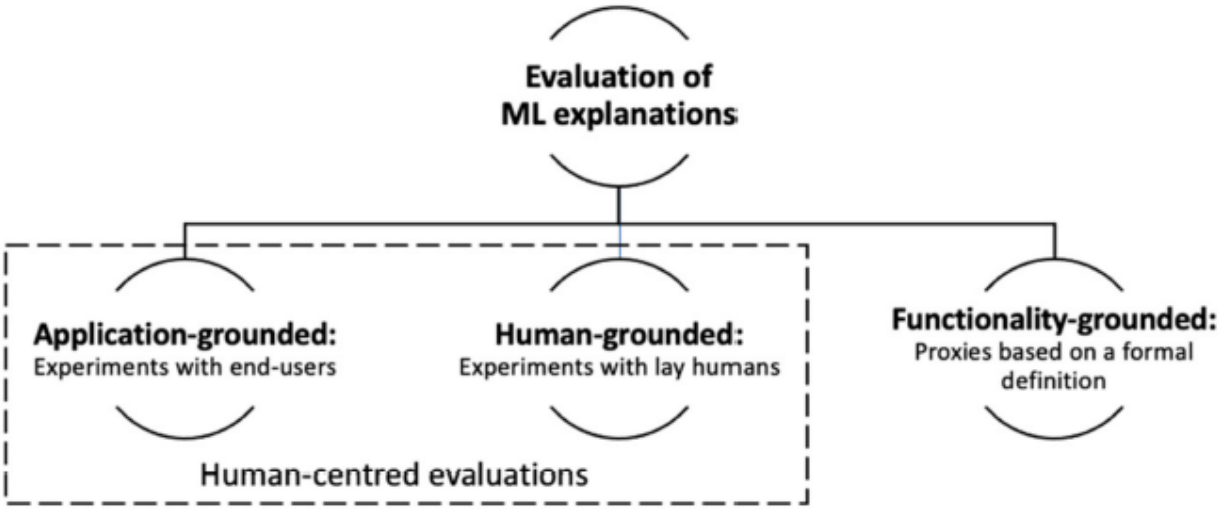


Рисунок 3.23. Таксономия оценки объяснений ИИ

Человеко-центричные оценки средств XAI. Согласно [Holland, 2020], понятность системы машинного обучения определяют два фактора: особенности системы машинного обучения и способность человека к пониманию. В связи с этим в широко цитируемой таксономии Доши-Велеса и Кима [Doshi-Velez, 2017] подходы к оценке средств XAI разделены на три категории (рис. 3.23):

– Оценка приложений (эксперименты с конечными пользователями). Этот вид оценки требует проведения экспериментов с конечным пользователем с использованием объяснения в реальном приложении. Он напрямую проверяет цель, для которой создана система, в реальном приложении, и производительность по отношению к этой цели является убедительным доказательством успеха объяснений. Цель здесь – оценить, насколько хорошо объяснения помогают людям выполнять свои задачи, например, принять решение.
– Человеческая оценка (эксперименты с непрофессионалами) – более простые эксперименты с участием человека, которые поддерживают суть целевого приложения. По сравнению с оценкой приложений эксперименты проводятся не с экспертами в предметной области, а с непрофессионалами, что позволяет использовать большой пул субъектов и снизить затраты на оценку. В идеале этот подход к оценке зависит только от качества объяснения, но не от типов объяснений и точности связанного с ними прогноза.
– Оценка, основанная на функциональности (на формальном определении интерпретируемости). Этот вид оценки не требует экспериментов на людях.

Здесь некоторое формальное определение интерпретируемости служит заместителем для оценки качества объяснения. Примером такой оценки может служить глубина дерева решений.

Преимущества человеко-центричных оценок заключаются в том, что они могут предоставить прямые и убедительные доказательства успеха объяснений. Однако эти оценки обычно дороги и требуют много времени из-за приглашения людей и необходимых разрешений (например, от комитетов по этике исследований на людях), а также дополнительного времени для проведения экспериментов. Самое главное, что эти оценки субъективны.

Субъективные показатели для человеко-центричных оценок в основном базируются на опросниках. Вопросы задаются во время или после выполнения задачи, чтобы получить субъективные ответы пользователя по задачам и объяснениям. Примерами субъективных метрик являются доверие, уверенность и предпочтения пользователей, которые в значительной степени используются в качестве фокуса для оценки объяснимых систем. Например, в [Holzinger, 2020] предложена шкала под названием System Causability Scale, чтобы быстро определить, подходит ли и в какой степени объяснение или сам процесс объяснения для намеченной цели. Шкала эффективна для понимания требований к объяснениям человеко-машинного интерфейса ИИ, которые часто специфичны для конкретной предметной области.

Объективные показатели для человеко-центричных оценок включают человеческие метрики, такие как физиологические и поведенческие показатели людей, во время принятия решений, или метрики, связанные с задачей, такие как продолжительность задачи и производительность задачи. Например, в [Zhou, 2019] показано, что физиологические сигналы, такие как кожно-гальваническая реакция и наполняемость пульса, могут использоваться в качестве индикаторов доверия пользователей к оценке качества объяснений машинного обучения. Авторы [Schmidt, 2019] предположили, что более быстрые и точные решения указывают на интуитивное понимание объяснений. Количественно определяя время отклика в исследованиях пользователей, на основе этих показателей объяснимости они получили метрику доверия.

Оценки, основанные на функциональности, могут обеспечить объективные количественные показатели без экспериментов на людях. Примеры таких оценок [Zhou, 2021] сведены в табл. 3.5. Здесь показатели, основанные на размере модели, в основном используются для моделей типа дерева решений – например, количество правил, длина правил и глубина деревьев.

Количество операций во время выполнения модели – здесь измеряется количество логических и арифметических операций, необходимых для запуска объяснимой модели для заданных данных

Сложность основного эффекта для каждой фичи измеряется количеством параметров, необходимых для аппроксимации накопленного локального эффекта с помощью кусочно-линейной модели. Общая сложность основного эффекта получается путем усреднения основных эффектов, взвешенных по дисперсии всех признаков.

Сила взаимодействия определяется как мера зависимости эффекта функции от значений других функций. Он измеряется путем аппроксимации ошибок между моделью, состоящей из суммы накопленных локальных эффектов, и исходной моделью задачи с взаимодействиями.

Уровень (не)согласия определяется как процент предсказаний, которые совпадают между моделью задачи и объяснениями модели.

Таблица 3.5. Количественные метрики для объяснимости ИИ

Тип объяснения	Количественные метрики
Объяснения на основе модели	Размер модели Количество операций во время выполнения Сложность основного эффекта Сила взаимодействия Уровень несогласованности
Объяснения на основе атрибутов	Монотонность Нечувствительность / чувствительность Эффективная сложность Удаление признаков и переобучение Удаление важных фич Селективность Непрерывность n-чувствительность Взаимная информация
Объяснения на основе примеров	Нерепрезентативность Разнообразие

Оценки на основе возмущений (пертурбаций) входного объекта. Количественные показатели для объяснений, основанных на атрибутах, в целом базируются на идее возмущений (пертурбаций) входного объекта. В качестве примера рассмотрим метод контрастных объяснений (Contrast Explanation Method, CEM) [Dhurandhar, 2018]. Здесь вначале ищутся фичи входного датасета (признаки, пиксели, группы пикселей, фрагменты текста и т.п.), минимально достаточные для обоснования классификации – они называются релевантно позитивными (pertinent positives, PPs). Затем ищется минимальный набор фич, называемых релевантно негативными (pertinent negatives, PN), которые, если их сделать ненулевыми или добавить, изменяют классификацию и, следовательно, должны отсутствовать, чтобы сохранилась исходная классификация.

Такие формы объяснений не только распространены в повседневных социальных взаимодействиях (например, близнецы могут различаться по тому, имеет ли один из них шрам), но также широко используются в медицине и криминологии, где аргументы в пользу PN являются наиболее важным аспектом объяснения.

На основе подхода окклюзии строятся количественные оценки эффективности моделей XAI. Для этого, например, в [Yang, 2019] создается датасет BAM путем копирования групп пикселей, называемых общими функциями (CF),

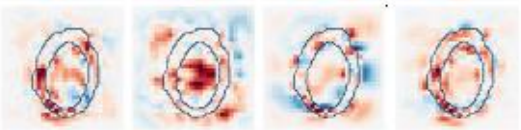
представляющих категории объектов из датасета MSCOCO и вставки их в датасета MiniPlaces.

Для повышения общности подхода в методах атрибуции некоторые авторы вводят специализированные метрики. Так, в [Melis, 2018] авторы описали метрику Faithfullness для оценки корреляции между оценками важности фичи и влиянием производительности каждой фичи на правильное предсказание. Поэтапно удаляя важные фичи и делая прогнозы для отредактированного экземпляра данных, они измеряют эффект важности фичи, а затем сравнивают его с собственным прогнозом релевантности интерпретатора.

В [Luss, 2019] авторы вводят функции монотонных атрибутов и, таким образом, метрику монотонности (Monotonicity), которая измеряет важность или влияние отдельных фич данных на эффективность модели путем постепенного добавления каждой функции. Ожидается, что эффективность модели будет увеличиваться по мере добавления более важных фич.

В таблице 3.6 [Ancona, 2018] представлено сравнение градиентных и пертурбационных методов атрибуции. В таблице слева представлена математическая формулировка пяти методов атрибуции на основе градиента и окклюзии размером 1×1, а справа – примеры атрибуций в наборе данных MNIST с четырьмя CNN, использующими разные функции активации.

Таблица 3.6. Сравнение градиентных и пертурбационных методов атрибуции

Method	**Attribution** $R_i^c(x)$	**Example of attributions on MNIST** ReLU	Tanh	Sigmoid	Softplus
Gradient * Input	$x_i \cdot \frac{\partial S_c(x)}{\partial x_i}$				
Integrated Gradient	$(x_i - \bar{x}_i) \cdot \int_{\alpha=0}^{1} \frac{\partial S_c(\tilde{x})}{\partial(\tilde{x}_i)}\Big\|_{\tilde{x}=\bar{x}+\alpha(x-\bar{x})} d\alpha$				
ϵ-LRP	$x_i \cdot \frac{\partial^g S_c(x)}{\partial x_i}, \quad g = \frac{f(z)}{z}$				
DeepLIFT	$(x_i - \bar{x}_i) \cdot \frac{\partial^g S_c(x)}{\partial x_i}, \quad g = \frac{f(z) - f(\bar{z})}{z - \bar{z}}$				
Occlusion-1	$S_c(x) - S_c(x_{[x_i=0]})$				

3.2.4. Программные средства открытого доступа для объяснимого ИИ

Пакеты OpenSource значительно улучшили воспроизводимость исследований в области глубокого обучения и XAI. Ниже перечислены некоторые программные пакеты для реализации средств XAI, доступные на GitHub.

- Пакет Interpret by InterpretML может использоваться для объяснения моделей черного ящика и в настоящее время поддерживает такие методы, как объяснимое усиление, деревья решений, список правил принятия решений, линейно-логистическую регрессию, ядерные и древовидный SHAP, LIME, анализ чувствительности Морриса и частичную зависимость (explainable boosting, decision trees, decision rule list, linear logistic regression, SHAP kernel explainer, SHAP tree explainer, LIME, Morris sensitivity analysis, and partial dependence).

  Доступен на https://github.com/interpretml/interpret.
- Пакет IML [Molnar, 2018] поддерживается Кристофом Молнаром. Пакет охватывает важность признаков, графики частичной зависимости, графики отдельных условных ожиданий, накопленные локальные эффекты, суррогаты деревьев, LIME и SHAP (feature importance, partial dependence plots, individual conditional expectation plots, accumulated local effects, tree surrogates, LIME, and SHAP.

  Доступен на https://github.com/christophM/iml.
- Пакет DeepExplain поддерживается Марко Анкона [Ancona, 2018]. Пакет поддерживает различные методы на основе градиента, такие как карты заметности, ввод градиента, интегрированные градиенты, DeepLIFT, LRP и т.д., а также методы на основе возмущений, такие как окклюзия, SHAP и т.д. (saliency maps, gradient input, integrated gradients, DeepLIFT, LRP, etc. and perturbation based methods such as occlusion, SHAP).

  Доступен на https://github.com/marcoancona/DeepExplain
- DrWhy от ModelOriented – это пакет с несколькими независимыми от модели и специфическими для модели методами XAI, включая важность функций, при прочих равных условиях, графики частичной зависимости, условную зависимость и т.д. (feature importance, ceteris paribus, partial dependency plots, conditional dependency, etc.)

  Доступен на https://github.com/ModelOriented/DrWhy.

Вопросы для самопроверки

1. По каким основаниям можно классифицировать системы ИИ в соответствии с ГОСТ Р 59277–2020?
2. Как определяется доверие к системе ИИ в соответствии с ГОСТ Р 59276–2020?
3. Как можно классифицировать средства объяснимого ИИ?
4. Каков принцип построения карт значимости (Salience Maps) в глубоких нейронных сетях?

5. Каков принцип построения локальных интерпретируемых моделенезависимых объяснений (LIME) в глубоких нейронных сетях?
6. Каков принцип построения аддитивных объяснений Шепли (SHAP) в глубоких нейронных сетях?
7. Какие методы снижения размерности признакового пространства используются для выделения наиболее значимых признаков в объяснимом ИИ?
8. В чем состоит гипотеза окклюзии в объяснимом ИИ?
9. Каков принцип построения локальных интерпретируемых моделенезависимых объяснений (LIME) в глубоких нейронных сетях?
10. Поясните идею метода отображения активации классов (CAM) при объяснении работы модели ИИ по входному изображению.
11. Какие методы объяснимого ИИ на основе байесовских правил вы знаете?
12. Существует ли общепризнанная схема сравнительной оценки различных методов объяснимого ИИ?
13. Поясните сущность человеко-центричных оценок средств объяснимого ИИ.

4. БАЗОВЫЕ ТЕХНОЛОГИИ ИИ

4.1. Общие сведения

Когнитивные функции, которые, по определению, призван имитировать ИИ, используются человеком в его конструктивной активности, т.е. при решении возникающих перед ним проблем. Согласно представлениям когнитивистики, основным средством решения проблем у человека является многоуровневое и многоаспектное моделирование реальности, схема которого представлена на рис. 4.1 [Введение, 2007].

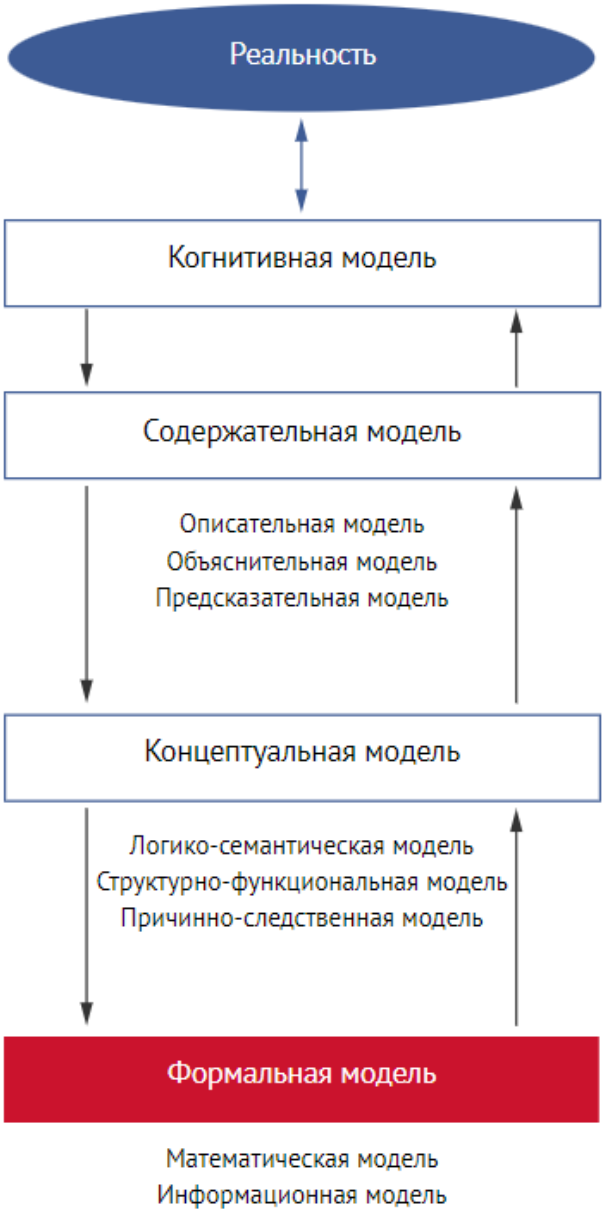


Рис. 4.1. Взаимосвязь моделей при решении задач

При наблюдении за объектом-оригиналом в голове человека формируется некий мысленный образ объекта, который в научной литературе принято

называть когнитивной моделью. Когнитивные модели субъективны, так как, во-первых, они формируются на основе всех предыдущих знаний и опыта, во-вторых, они могут содержать элементы, которые исследователь не может или не хочет сформулировать.

Получить представление о когнитивной модели конкретного человека можно, только описав ее в знаковой форме. Представление когнитивной модели на естественном языке называется содержательной моделью.

По функциональному признаку и целям содержательные модели подразделяются на описательные, объяснительные и прогностические. Описательной моделью можно назвать любое описание объекта. Объяснительная модель позволяет ответить на вопрос, почему что-либо происходит. Наконец, прогностическая модель должна описывать будущее поведение объекта (при этом она не обязана включать в себя объяснительную).

Концептуальной моделью принято называть содержательную модель, при формулировке которой используются понятия и представления предметных областей знания, занимающихся изучением объекта моделирования. В более широком смысле под концептуальной моделью понимают содержательную модель, базирующуюся на определенной концепции или точке зрения.

Выделяют три вида концептуальных моделей: логико-семантические, структурно-функциональные и причинно-следственные. Логико-семантическая модель является описанием объекта в терминах и определениях соответствующих предметных областей знаний, включающим все известные логически непротиворечивые утверждения и факты. Анализ таких моделей осуществляется средствами логики с привлечением знаний, накопленных в соответствующих предметных областях.

При построении структурно-функциональной модели объект обычно рассматривается как целостная система, которую расчленяют на отдельные элементы или подсистемы. Части системы связываются структурными отношениями, описывающими подчиненность, логическую и временную последовательность решения отдельных задач. Для представления подобных моделей удобны различного рода схемы, карты и диаграммы.

Причинно-следственная модель часто используется для объяснения и прогнозирования поведения объекта. Эти модели ориентированы в основном на выявление главных взаимосвязей между составными элементами изучаемого объекта и оценку динамики их взаимодействия.

Формальная модель является представлением концептуальной модели с помощью одного или нескольких формальных языков (например, языков математических теорий, универсального языка моделирования (UML) или алгоритмических языков).

Как показано в разделе 1, для воспроизводства когнитивных функций человека ИИ имеет в своем арсенале технологические решения, позволяющие реализовать в той или иной мере все вышеуказанные модели. Сюда входят как технологические решения, являющиеся общими для большого набора информационных технологий (в том числе информационно-коммуникационная инфраструктура, базы данных, программное обеспечение общего назначения и т.п.), так и

решения, специфичные именно для ИИ. К последним принято относить основные универсальные модели представления и обработки концептов (знаний), в том числе:

- логические модели;
- сетевые модели – семантические сети, фреймы, сценарии, продукционные системы, байесовские сети;
- средства обработки неопределенности – нечеткие модели, модели на основе логики Демпстера-Шефера, модели на основе грубых множеств;
- онтологические модели;
- нейронные сети.

4.2. Логические модели

4.2.1. Логика Аристотеля

Базисом для всех существующих логических языков явилась логика (теория силлогистических выводов) Аристотеля. Она строится на следующих концептах:

- Сущность (обозначаются малыми латинскими буквами) – все то, о чем можно нечто утверждать.
- Классы сущностей (обозначаются большими латинскими буквами) – совокупности, объединенные с помощью общего имени.
- Квантор "Всякий" $\forall$, поставленный рядом с именем класса, показывает, что в высказывании будет утверждаться то, что одновременно истинно для всех сущностей, входящих в этот класс.
- Квантор "Некоторый" (квантор существования) $\exists$, поставленный рядом с именем класса сущностей, показывает, что в высказывании будет утверждаться нечто относительно какого-либо подкласса сущностей, входящих в данный класс.

С помощью кванторов строятся схемы базовых высказываний, используемые в силлогистике:

1. Всякий S есть P (в общем случае схема A "всякий … есть …")
2. Всякий S не есть P (в общем случае схема E "всякий … не есть …")
3. Некоторый S есть P (в общем случае схема I "некоторый … есть …")
4. Некоторый S не есть P (в общем случае схема O "некоторый … не есть …")
5. S есть P
6. S не есть P
7. a есть P
8. a не есть P

Здесь S – субъект высказывания (определяет сущности, о которых что-то утверждается в высказывании), P – предикат высказывания (определяет, что именно говорится). Связки "есть", "не есть" для высказываний 1–6 интерпретируются как отношения включения (невключения) множеств сущностей, а для высказываний 7 и 8 – как отношения принадлежности элемента множеству.

- "|— " – знак выводимости. Запись F|—Q означает, что если все посылки,

входящие в F, выводимы, то заключение Q, также выводимо. Запись |—Q означает, что Q выводимо всегда.

Любой вывод в силлогистике Аристотеля может иметь или не иметь посылок. По числу посылок можно различать выводы ранга 0,1,2 и т.д.

Выводы ранга 0 – это утверждения, для которых посылки не нужны; они называются законами силлогистики. Таких законов в силлогистике три:

- закон тождества (всякая конкретная сущность, входящая в класс s, обладает всеми свойствами элементов этого класса);
- закон противоречия (невозможна такая ситуация, когда конкретные сущности из класса s одновременно входят в класс p и не входят в него);
- закон исключенного третьего (для каждой конкретной сущности a, входящей в класс S, истинно одно из двух утверждений: "a входит в p" или "a не входит в p"; третьего не дано).

Выводы ранга 1 специального названия не имеют.

Выводы ранга 2 называются силлогизмами Аристотеля. В посылках таких выводов фигурируют три класса сущностей: S – меньший термин; P – больший термин; M – средний термин, причем каждая из посылок и заключение представляют собой базовые высказывания силлогистики. Конкретные типы силлогизмов, порожденных с использованием данных структур, называются модусами. Порождение модуса происходит путем выбора из четырех схем A, E, I, O по одной схеме для первой и второй посылок и заключения, что дает 256 различных модусов.

Примеры.

- Модус первой фигуры AAA имеет вид:

Всякий M есть P	Всякая птица имеет перья.
Всякий S есть M	Всякий дятел — птица.
---	---
Всякий S есть P	Всякий дятел имеет перья.

- Модус первой фигуры EJO:

Всякий M не есть P	Всякий дятел красиво не поет.
Некоторые S есть M	Некоторые птицы - дятлы.
---	---
Некоторые S не есть P	Некоторые птицы красиво не поют.

- Модус четвертой фигуры IAO:

Некоторые Pесть M.	Некоторые люди - негры.
Всякий M есть S.	Всякий негр - темнокожий
---	---
Некоторые S не есть P.	Некоторые темнокожие - не люди.

Как видно из примеров, имеются правильные и неправильные силлогизмы (модусы). Из 256 возможных силлогизмов только 24 являются правильными, а остальные могут привести к ошибочному выводу. Правильные модусы образуют ядро теории дедуктивных выводов, в котором от правильных посылок всегда гарантируется переход к правильному заключению.

Кардинальный сдвиг в анализе стандартных рассуждений произошел в тот

период, когда для создания логической теории был применен метод построения формальных систем с помощью специальных символических языков.

Использование логик различных типов позволяет создавать разные логические модели. В частности, были созданы две мощные формальные системы, которые впервые позволили автоматизировать рассуждения, опирающиеся на схему дедуктивного вывода: исчисление высказываний (ИВ) (синоним слова "высказывание" – пропозиция) и исчисление предикатов (ИП).

4.2.2. Исчисление высказываний

Под высказыванием понимается утверждение, относительно которого в любой момент можно сказать, является ли оно истинным или ложным, или, по крайней мере, предполагать, что ему может быть приписана такая интерпретация. Примеры высказываний:

Река Волга впадает в Каспийское море (истинно).

Все жители Земли имеют рост более двух метров (ложно).

В Африке находятся более десяти еще не известных захоронений фараонов (можно приписать некую интерпретацию).

Как любая формальная теория, ИВ определяется как кортеж:

$$M = \langle T, P, A, \Pi \rangle,$$

где T – алфавит системы; P – синтаксические правила; A – система аксиом; П – семантические правила.

Алфавит ИВ включает переменные высказываний (пропозициональные буквы) A, B, C…; знаки логических связок {$\wedge$ или & – «и», $\vee$– «или», $\rightarrow$ – «если…то…», $\neg$ или черта над формулой – «не»}; скобки (,).

Формулы ИВ определяются следующими правилами:

- переменное высказывание есть формула;
- если Ψ и Ω – формулы, то $(\Psi\vee\Omega)$,$(\Psi\&\Omega)$, $(\Psi\rightarrow\Omega)$ и $\neg\Psi$ – формулы;
- других формул нет.

В ИВ возможны различные системы аксиом, порождающие одно и то же множество формул. Одна из них (10 штук) непосредственно использует все логические связки. Другая использует только две связки $\neg$ и $\rightarrow$; операции & – «и», $\vee$ – «или» рассматриваются только как сокращения, а именно: $A\vee B = \neg A\rightarrow B$; $A\&B = \neg(A\rightarrow\neg B)$. Тогда аксиом всего три:

$A\rightarrow(B\rightarrow A)$;

$(A\rightarrow(B\rightarrow C) \rightarrow(A\rightarrow B) \rightarrow (A\rightarrow C)$;

$(\neg A\rightarrow\neg B) \rightarrow((\neg A\rightarrow B) \rightarrow A)$.

В ИВ определены два правила вывода.

- Правило подстановки: если Ψ – выводимая формула, содержащая букву A (обозначим этот факт $\Psi(A)$), то выводима формула $\Psi(\Omega)$, которая получается из Ψ заменой всех вхождений A на произвольную формулу, т.е.

Ω; $\Psi(A)$

----------;
Ψ(Ω)

- Правило заключения (modus ponens): если Ψ и Ψ→Ω – выводимые формулы, то Ω выводима, т.е.

Ψ; Ψ→Ω
----------.
Ω

Из символов и исходных аксиом выводятся вторичные аксиомы, которые называются теоремами, а также разнообразные высказывания. Высказывания могут представлять собою либо ложь, либо истину.

В математической логике показывается, что ИВ выполняет задачу порождения общелогических закономерностей – тождественно-истинностных высказываний. С точки зрения теории алгоритмов множество всех истинных высказываний ИВ перечислимо и разрешимо.

Область применения ИВ – анализ и синтез конечных автоматов, которые, как известно, являются логической основой компьютеров.

Однако существенным недостатком аппарата ИВ является ограниченность выразительных средств, что не позволяет описывать логические задачи и дедуктивные рассуждения всех типов, в частности, силлогистические умозаключения. Естественным развитием аппарата ИВ является исчисление предикатов, разработанное в рамках логики предикатов.

4.2.3. Исчисление предикатов

Определения. Понятие «предикат» в ИП не совпадает с тем же понятием в логике Аристотеля. В ИП предикат – это обобщение понятия «высказывание». Неформально говоря, предикат – это высказывание, в которое можно подставлять аргументы. Если аргумент один, то предикат *P,* называемый одноместным, выражает свойство аргумента a и обозначается $P(a)$. Если аргументов два, то предикат $R(a,b) = aRb$, называемый двухместным, выражает отношение между аргументами a и b. Можно также ввести многоместные предикаты, выражающие отношение между соответствующим количеством аргументов.

В ИП истинностные значения ставятся в соответствие не только высказываниям в целом, но и отдельным предметам и группам предметов.

Обычно (но не всегда) стараются строить фразы логического языка в виде элементарных высказываний (простых синтагм, предикатных форм) вида aRb. Тогда описание объекта представляет собою совокупность синтагм, а совокупность описаний – информационный массив, относящийся к данной предметной области.

Дадим теперь формальные определения.

Предикатом $P(x_1,..., x_n)$ называется функция $P:M^n \rightarrow \{0,1\}$, т.е. функция, принимающая значение "0" или "1", аргументы которой пробегают значения из произвольного множества M. В ИП, как и в логике Аристотеля, используется два квантора: квантор всеобщности и квантор существования. Первый обозначается

как ∀, а запись $\forall xP(x)$ эквивалентна утверждению: "Для всех x из области его определения имеет место $P(x)$". Второй квантор обозначается как ∃, а запись $\exists xP(x)$ эквивалентна утверждению: "Найдется, по крайней мере, один x в области определения x, такой, что истинен $P(x)$".

Алфавит ИП состоит из следующих компонентов:

- предметные переменные $x_1, x_2,..., x_n$, принимающие значения из некоторой предметной области;
- предметные константы $a_1, a_2, ..., a_m$;
- предикатные буквы (константы) $P_1, P_2, ..., P_K$;
- функциональные буквы (константы) $f_1, f_2,..., f_q$;
- знаки логических связок. $\vee$, $\&$, $\neg$, $\rightarrow$;
- кванторы ∀, ∃;
- скобки (,).

Понятие формулы ИП определяется в два этапа.

Этап 1 – определение терма. Предметные переменные и константы являются термами. Если f – функциональная буква, а $t_1, ..., t_n$ -термы, то $f(t_1, ..., t_n)$ – терм.

Этап 2 – определение формулы. Если P – предикатная буква, а $t_1, ..., t_n$ – термы, то $P(t_1, ..., t_n)$ – формула. Формулу данного вида называют часто атомарной формулой или атомом. Если A и B – формулы, то формулами являются $\neg A$, $A \rightarrow B$, $A \vee B$, $A \& B$. Если A – формула, а x – предикатная переменная, входящая в A, то $\forall xA(x)$ и $\exists xA(x)$ – формулы. Выражение является формулой в том и только в том случае, если это следует из данных правил.

Переменные, находящиеся в сфере действия кванторов, называются связанными, остальные переменные – свободными. Атом или отрицание атома иногда называют литералом.

Формулы A и B ИП называются равносильными (эквивалентными), если является общезначимой формула

$(A \rightarrow B) \& (B \rightarrow A)$.

Этот факт отмечается как $A = B$ или $B = A$.

Между высказываниями силлогистики и формулами ИП существует соответствие:

A– Всякое s есть p $\Leftrightarrow$ $\forall x\big(s(x)\big) \rightarrow p(x)$.

E – Всякое s не есть p $\Leftrightarrow$ $\forall x\big(s(x)\big) \rightarrow \neg p(x)$.

I – Некоторые s есть p $\Leftrightarrow$ $\exists x\big(s(x)\big) \rightarrow p(x)$.

O – Некоторые s не есть p $\Leftrightarrow$ $\exists x\big(s(x)\big) \rightarrow \neg p(x)$. .

Исследование выводимости 24 модусов, верных в силлогистике Аристотеля, в ИП привело к следующему результату. Если предполагать, что все классы сущностей не пусты, то приведенная выше замена силлогистических выражений формулами ЛП будет полностью справедлива. При допущении пустых классов сущностей оказываются невыводимыми все модусы силлогистики, в которых вывод носит частный характер, а обе посылки носят общий характер.

Аксиомы ИП включают в себя две группы – аксиомы исчисления

высказываний, а также предикатные аксиомы:

$\forall x F(x) \rightarrow F(y)$;

$F(y) \rightarrow \exists\, x F(x)$.

Здесь $F(x)$ – любая формула, содержащая свободные вхождения x, причем ни одно из них не находится в области действия квантора по y; формула $F(y)$ получена из заменой всех свободных вхождений x на y.

Правила вывода ИП делятся на три группы:

- правило заключения (modus ponens): если Ψ и $\Psi \rightarrow \Omega$ – выводимые формулы, то Ω выводима, т.е.

  Ψ; $\Psi \rightarrow \Omega$

  Ω

  – то же, что и в исчислении высказываний;
- правило обобщения ($\forall$-введения):

  $F \rightarrow G(x)$

  ----------,

  $F \rightarrow \forall x G(x)$

  где $G(x)$ содержит свободные вхождения x, а F их не содержит.
- правило $\exists$-введения:

  $G(x) \rightarrow F$

  ----------,

  $\exists\, x G(x) \rightarrow F$

  где $G(x)$ содержит свободные вхождения x, а F их не содержит.

Аналогично ИВ, набор аксиом и правил вывода для ИП можно уменьшить, заменяя квантор существования $\exists$ на выражение $\neg\forall A\neg$. Эта процедура называется сколемизацией и используется в методе резолюций.

Построенное таким образом ИП называется ИП первого порядка. В исчислениях второго порядка возможны кванторы по предикатам, т.е. выражения вида $\forall P(P(x))$. Однако приложения таких исчислений встречаются достаточно редко, и прикладное значение имеет в основном ИП первого порядка.

Дедуктивный вывод на знаниях – основное применение ИП. В логических моделях решаемая проблема записывается в виде утверждений ИП, цель – в виде утверждения, справедливость которого следует установить или опровергнуть на основании аксиом и правил вывода.

Доказательством теоремы называется поиск ответа на вопрос: следует ли логически формула B из заданного множества формул (в частности, аксиом) $\Sigma = \{A_1, A_2, \ldots, A_n\}$. Доказательство демонстрирует, что некоторая формула B является теоремой на заданном множестве аксиом Σ, т.е. результатом, логически выводимым из аксиом. Из практических соображений удобнее доказывать противоречивость отрицания B.

Доказано (теорема Черча), что не существует эффективной (т.е. выполнимой за конечное число шагов) разрешающей процедуры для исчисления предикатов первого порядка, позволяющей узнать по данной формуле, является она теоремой или нет. Поэтому методы поиска доказательств подтверждают, что

теорема общезначима, если она таковой является. Для необщезначимых формул алгоритмы доказательства работают бесконечно долго. Принимая во внимание результаты Черча и Тьюринга, это лучшее, что можно ожидать от процедур доказательства теорем.

Основной метод поиска доказательства в ИП – метод резолюций. Любую логическую формулу можно представить в виде множества дизъюнктов (можно просто сказать, что дизъюнкт – это совокупность литералов, неявно соединенных дизъюнкцией. Основная идея метода резолюций состоит в проверке, содержит ли множество дизъюнктов S пустой дизъюнкт □. Дизъюнкт называется пустым, если он не содержит никаких литер. Так как пустой дизъюнкт □ не содержит литер, которые могли бы быть истинными при любых интерпретациях, то он всегда ложен. Следовательно, если *S* содержит □, то *S* противоречиво (невыполнимо). Новые дизъюнкты порождаются на основании силлогизма modus ponens следующим образом: если в любых двух дизъюнктах имеется контрарная пара литер (P и ¬P), то эта пара вычеркивается, а из оставшихся частей формируется новый дизъюнкт – резольвента. *Пример:* из двух дизъюнктов P и ¬P∨Q можно сформировать их резольвенту Q.

Метод резолюций обладает полнотой. В силу неразрешимости логики предикатов первого порядка для выполнимого множества дизъюнктов метод будет работать бесконечно долго. Он порождает экспоненциально растущее число резольвент, большинство из которых оказывается ненужным. Есть целый ряд модификаций метода резолюций, которые основаны на использовании специфики предметной области.

Формализация текстов с использованием формул логики предикатов. С использованием формул ИП можно формализовать тексты. При описании текста формула строится вместе со своей интерпретацией.

Пример 1. И.А. Крылов: "А вы, друзья, как ни садитесь, все ж в музыканты не годитесь!". Обозначим через $P(x,y)$ предикат, который связывает между собой способ рассаживания участников квартета к качество исполняемой ими музыки. Предикат $P(x,y)$ становится истинным лишь тогда, когда найдено такое взаимное расположение зверей в квартете, что качество музыки позволяет назвать исполнителей музыкантами. При этих условиях цитате из басни "Квартет" соответствует формула $\forall x \neg P(x,y)$.

Пример 2. Представим на языке ИП фразу «Деталь № 1244 обрабатывается на станке»:

$(a\rho I) \wedge (adb)$,

Здесь использованы обозначения: a – деталь, b– станок, ρ – предикат «иметь имя», d – предикат «процесс обработки», $\wedge$ – « и» (знак коньюнкции).

Пример 3. Представим на языке ИП ситуацию «Между роботом и складом находится яма, слева от которой расположен экскаватор»:

$P1(a, b) \wedge P2(a, b, c)$.

Здесь $P1(x, y)$ – предикат «быть слева от», $P2(x, y, z)$ – предикат «быть между», a – экскаватор, b – яма, c – склад.

Пример 4 – система предикатов:

«Авиационный двигатель с системой турбонаддува, в котором использована двухконтурная схема. Двигатель установлен в хвостовой части фюзеляжа. Хвостовое оперение фюзеляжа предусматривает защиту от флаттера».

Обозначения: Х – хвостовое оперение, Ф – фюзеляж, Д – двигатель, А – авиационный, З – защита, ДС – двухконтурная схема, С – система, Т – турбонаддув, ФЛ – флаттер, R_1 – реализует, R_2 – включает, R_3 – контактирует.

Последовательность простых синтагм:

Д R_3Ф Х R_1З Д R_2С С R_1ДС

Ф R_2Х З R_1ФЛ С R_1Т Д R_1А

Дополнение синтагм на основе формальных преобразований:

ARB $\Rightarrow$ BRA

Ф R_3Д А R_1Д Т R_1С ДС R_1С и др.

ARB $\Rightarrow$ BR^{-1}A

С R_2Д Х R_2Ф и др.

AR_2B $\cap$ BR_2C $\Rightarrow$ AR_1C

Д R_1Т Д R_1ДС и др.

Пример 5. «Часто, когда давление выше нормы, температура также выше нормы». Введем обозначения: $К_1$ – квантификатор «часто», h – давление, t – температура, m – модификатор «выше нормы», r_{48} – отношение «причина–следствие», r_{49} – отношение «способствовать». Тогда приведенное описание может быть записано в виде

$К_1(mh)\ r_{49}\ (mt)$

или

$К_1(mh)\ r_{48}\ (mt)$,

причем второй вариант скорее означает «Всегда при повышении давления выше нормы температура также превышает норму».

Отметим ограничения проблемно-ориентированного языка на ИП по сравнению с естественным языком.

Во-первых, во всех допустимых формально-логических преобразованиях простых высказываний конструируемого языка должны отсутствовать замкнутые циклы. Отсутствие циклов порождения синтагм должно проверяться в каждой разработке. Для их устранения необходимо изменять разработанную описательную и терминологическую базу соответствующей предметной области, что не всегда возможно.

Во-вторых, при переводе выражений с естественного языка на язык предикатов невозможно выразить признаки, характеризующие бинарные отношения (например, квантификаторы – часто, много, хорошо и т.д.), а также отношения сложнее бинарных (например, «Выход за расчетные параметры придает системе неустойчивость функционирования», «ориентированы в различных направлениях», «визуально не различаются»). Как правило, тонкие оттенки смысла не передаются, и язык на основе предикатов организуется в виде простых синтагм вида ARB. В результате информация становится более определенной, чем на естественном языке, но ситуация описывается как более жесткая по сравнению с действительностью. Например, фраза «Несмотря на разнообразие и сложность законов управления, их реализация осуществляется простыми техническими

средствами на основе типовых логических элементов» на языке предикатов сводится к типовой синтагме A*R*B, где A – закон управления, B – логические элементы, *R* – отношение «реализует».

Подчеркнем, что приведенные «переводы» не являются точно поставленными математическими вопросами: не существует способа доказать (в математическом смысле), что перевод адекватен.

4.2.4. Логические системы с изменяющимися отношениями

Как отмечено в разделе 4.1.1, Аристотель предложил 24 модуса силлогизмов, которые дают всегда верные результаты. Остальные модусы четырех возможных фигур Аристотеля не являются в строгом смысле истинными.

В современных интеллектуальных системах все чаще происходит переход к другим логикам, более соответствующим «естественным» механизмам рассуждений человека. Основным приемом здесь является построение моделей с изменяющейся логикой отношений, причем такой, что ее правила учитывают семантику отношений. Некоторые примеры таких логических структур представлены ниже.

Немонотонные модальные логики. В классической логике действует принцип монотонности: если некоторое утверждение выводится в данной системе, то никакие дополнительные сведения не могут изменить этот факт. В открытых системах новые сведения могут изменить ситуацию, и сделанный ранее вывод может стать неверным.

Приведем некоторые примеры реализации немонотонных рассуждений.

- введение модального оператора unless (если не) поддерживает вывод на основе предположения о ложности аргумента. Например, пусть в множестве логических выражений присутствует следующий фрагмент:

  $p(X)$ unless $q(X) \rightarrow r(X)$,

  $p(Z)$,

  $r(W) \rightarrow s(W)$.

Если в ходе последующего вывода обнаружится, что $q(X)$ истинно, то $r(X)$ и $s(X)$ отменяются.

- введение модального оператора M ("не противоречит") образует логику Макдермотта-Дойла. Выражение

  $\forall X$ good_student$(X) \wedge M$ study_hard$(X) \rightarrow$graduates(X)

означает: "Для любого X, где X – хороший студент, если факт, что X прилежно учится, не противоречит остальной информации, то X закончит институт". Основной проблемой здесь является точное определение значения "не противоречит остальной информации", которое может вызвать неразрешимость вывода. В этом направлении предложен ряд решений, например: если значение peter однажды связано с предикатом study_hard, то система в дальнейшем не допустит связывания его с противоположным предикатом not(study_hard).

- множественное наследование, когда наследуются не все из возможных свойств.

- рассуждение на минимальных моделях может быть основано либо на предположении замкнутости мира, либо на ограничениях.

В первом случае используются только предикаты, необходимые для решения. Например, если мы хотим определить, является ли студент членом группы, то просматриваем базу данных (список этой группы). Если студент явно туда не помещен, он не является членом группы. Такую логику, в частности, поддерживает система PROLOG.

Во втором случае для решения задачи используются только существенные для нее предикаты. Для этого вводятся "метапредикаты", ограничивающие возможные интерпретации конкретных предикатов:

$\forall X\ bird(X) \wedge not(abnormal((X)) \rightarrow flies(X)$

– любая не ненормальная птица летает. Метапредикат задает, что означает abnormal – что это пингвин, что у нее перебито крыло или что она мертва.

Для защиты логической целостности заключений системы вывода используются специальные системы поддержки истинности. При любом изменении предположений в базе знаний такая система пересматривает заключения в свете новых предположений с той или иной глубиной возврата.

Правдоподобные (абдуктивный, индуктивный) выводы. Абдукция (лат. abductio – отведение) – силлогизм, у которого большая посылка достоверна, а меньшая – только вероятна. Т.е. логическая абдукция представляет собою процесс синтеза фактов из общих правил и результата.

Индукция (лат. inductio – наведение) – метод получения общего знания о классе объектов на основании исследования отдельных представителей этого класса. Наблюдаемое в опыте многократное повторение какого-то явления при отсутствии исключений внушает уверенность в его универсальности и естественным образом приводит к индуктивному обобщению — предположению, что именно так будет обстоять дело во всех сходных случаях.

Есть различные виды индукции. Полная (совершенная) индукция имеет место тогда, когда в опыте рассмотрены все возможные случаи. Здесь индуктивное обобщение тривиально. Его можно представить схемой дедуктивного умозаключения (сюда, в частности, относится математическая индукция).

На практике, как правило, число всех случаев практически необозримо, а теоретическое доказательство для бесконечного числа этих случаев невозможно. Т.е. обобщения делаются на основе исследования не всех случаев, а только некоторых. Такие обобщения называются неполной индукцией. Неполная индукция уже не является логически обоснованным рассуждением.

Идея правдоподобного вывода – ввести в логические рассуждения весовые коэффициенты ("весьма вероятно", "возможно", "маловероятно" и т.д.). Эти коэффициенты не основаны на статистике, а являются эвристиками, выведенными из опыта рассуждений о предметной области.

В качестве примера рассмотрим неточный вывод на основе фактора уверенности. При формировании базы правил с каждым правилом сопоставляется определенное значение фактора уверенности CF, $0<CF<1$. При использовании конкретной продукции учитываются значения CF, связанные с каждым условием предпосылки, а затем умножаются на CF ядра продукции, что дает CF

заключения. Заметим, что именно такая логика была использована при построении популярной экспертной системы MYCIN.

Псевдофизические логики широко использовались в работах Д.А. Поспелова (см. раздел 1.3). Правила вывода в этом случае отражают свойства восприятия человеком окружающего мира, т.е. они описывают не объективный физический мир, а субъективное представление человека о мире. Это, например, временная логика, отражающая закономерности, присущие человеку при восприятии времени и рассуждений о нем, логика пространство – действие, и т.д.

Типовая структура псевдофизической логики показана на рис. 4.2. Сначала из исходного описания ситуации выделяется некоторая структура фактов и событий (временная, пространственная и т.д.), в которой определяются базовые сущности (явления, события, процессы, факты) и отношения между ними из рассматриваемой группы отношений. В модели представлений отражены основные закономерности восприятия соответствующей структуры. Модель вывода содержит правила, с помощью которых происходит пополнение описания ситуации.

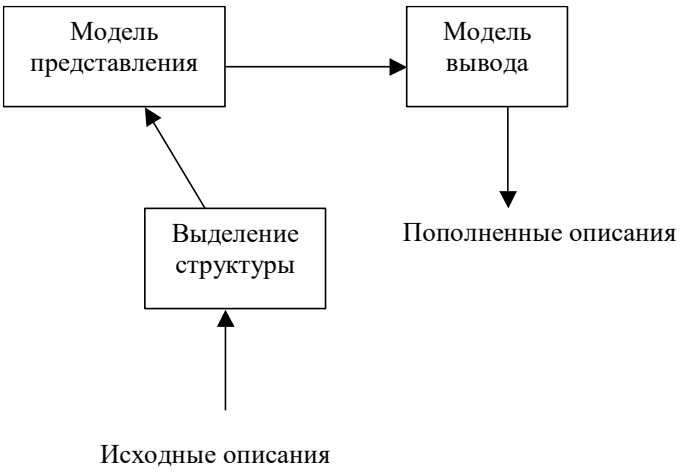


Рис. 4.2. Структура псевдофизической логики

Отметим некоторые особенности псевдофизических логик:

- изменению (масштабированию) подвергаются отношения, сами же объекты остаются неизменными;
- используются различные шкалы – метрические, порядковые, нечеткие и т.д. При шкалировании учитываются особенности человеческого восприятия и опыта. Например, шкала времени часто задается как порядковая, а не как количественная;
- в качестве аксиом используются некоторые утверждения, вытекающие из практики жизни человека, например: «Далеко ли магазин?» – «Минут 15–20». Это позволяет организовать связи между соответствующими шкалами в компьютерной модели.

Псевдофизические логики – один из сильных инструментов пополнения знаний в базах знаний.

4.3. Сетевые модели

Сетевые модели представляют собою сеть, вершины которой отождествляются с некоторыми понятиями, а дуги – с отношениями между понятиями. Это

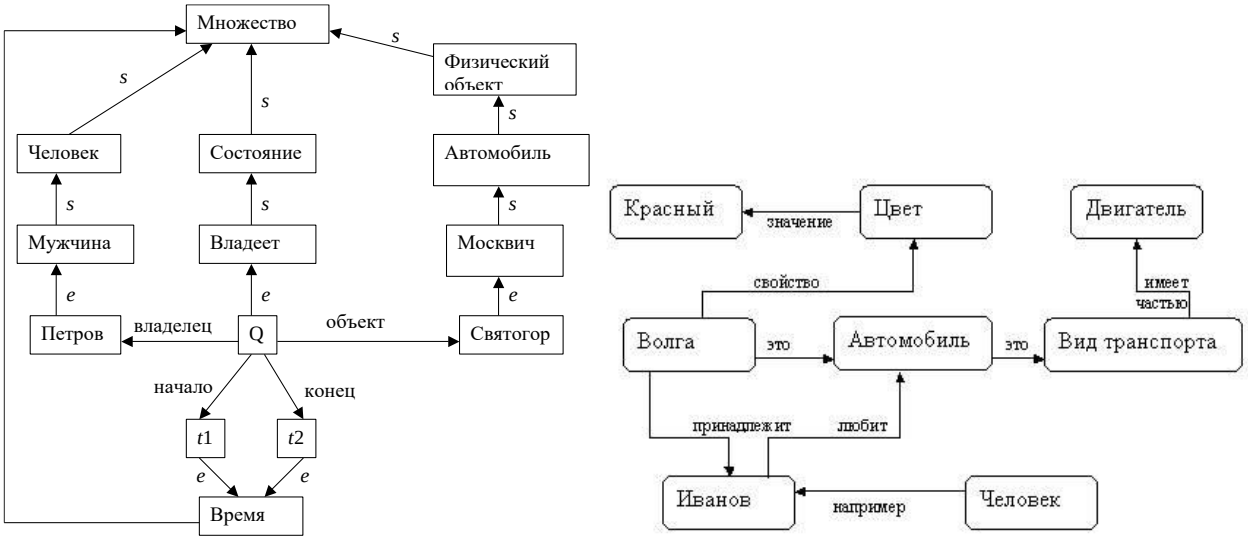
позволяет использовать для работы с ними аппарат теории графов. Вершины могут иметь собственную внутреннюю структуру.

Сетевые модели, в принципе, обладают теми же свойствами, что и исчисление предикатов, но более удобны для визуального представления сложных структур. Это достигается путем выделения в явной форме всех отношений, образующих информационную структуру предметной области, с описанием их семантики.

4.3.1. Семантические сети

Семантическая сеть – направленный граф с помеченными вершинами и дугами. Вершинам ставятся в соответствие понятия, а дугам – семантические отношения между ними. Метки вершин имеют ссылочный характер и представляют собою некоторые имена. Понятиями обычно выступают абстрактные или конкретные объекты, а отношения – это связи типа "это" ("is a"), "имеет частью" ("has part"), "принадлежит", "любит" и т.д.

Примеры семантических сетей представлены на рис. 4.3.



а б

Рис. 4.3. Примеры семантических сетей

Отношения в семантических сетях стремятся типизировать, например:

- связи типа "часть – целое" ("класс-подкласс", "класс – элемент класса", "элемент – множество" и т.п.);
- свойство – значение;
- функциональные связи (определяемые обычно глаголами "производит", "влияет"...);
- количественные (больше, меньше, равно...);
- пространственные (далеко от, близко от, за, под, над...);
- временные (раньше, позже, в течение...);
- атрибутивные связи (иметь свойство, иметь значение...);
- логические связи (и, или, не) и др.

В одной и той же сети можно использовать одновременно несколько типов отношений.

Вывод на семантических сетях. Особенность семантической сети состоит в ее целостности, т.е. в ней невозможно разделить экстенсиональную и интенсиональную часть (базу знаний и механизм выводов, если говорить традиционными для программирования словами).

Обычно интерпретации семантической сети определяется с помощью использующих ее процедур. Наиболее типичная процедура – способ сопоставления частей сетевой структуры. Он основан на построении подсети, соответствующей вопросу, и сопоставлении ее с основной сетью.

С каждой семантической сетью связывается совокупность дизъюнктов вида $B_1\&B_2\&\ldots\&B_m\rightarrow A_1\vee\ldots\vee A_n$ или, что то же самое, $\neg B1\vee\neg B2\vee\ldots\vee B_m\rightarrow A_1\vee\ldots\vee A_n$ (так как $\neg A\vee B$ равносильно $A\rightarrow B$). При этом вводятся два вида вершин: предикатные (соответствуют буквам, входящим в дизъюнкты) и дизъюнктные. Предикатные вершины связываются между собою только через дизъюнктные вершины.

Пример 1: для дизъюнкта g: $B_1\&B_2\rightarrow A_1\vee A_2$ сеть имеет вид рис. 4.4.

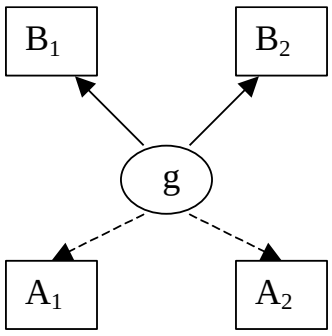


Рис. 4.4. Сеть для дизъюнкта (пример 1)

Проблема поиска решения в базе знаний типа семантической сети сводится к задаче поиска фрагмента сети, соответствующего некоторой подсети, соответствующей поставленному вопросу. При этом, как и в методе резолюций, используется процедура опровержения. Тогда задача дедуктивного вывода на сети формулируется следующим образом. Имеется сеть, с которой сопоставлен набор дизъюнктов; на вход сети подается ситуация, представленная совокупностью фактов $A_1,\ldots A_n$, и вопрос (также в виде дизъюнктов). Решение задачи состоит в постепенном переконфигурировании сети (методом удаления контрарных пар литер) так, чтобы получить противоречия в сети или пустые участки. Есть разные модификации решений, например, использующие идею пересечения путей на сетях или наложения одного фрагмента сети на другой. В основном они аналогичны идеям вывода на логических моделях.

Пример 2: пусть сеть имеет вид рис. 4.5.

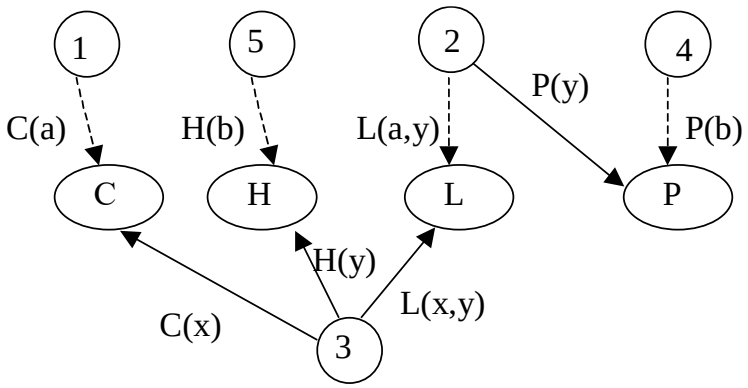


Рис. 4.5. Сеть для набора дизъюнктов (пример 2)

Применяя оператор удаления к вершине C, переходим к конфигурации рис. 4.6. Затем, применяя оператор удаления к вершинам H и L, получаем противоречие в вершине P (рис. 4.7, а) или пустую сеть (рис. 4.7, б).

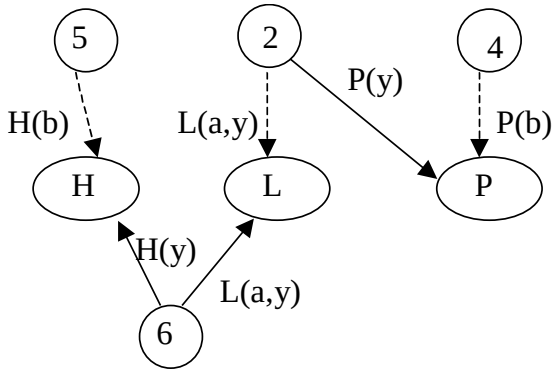


Рис. 4.6. Сеть для набора дизъюнктов (пример 2) – удаление вершины C

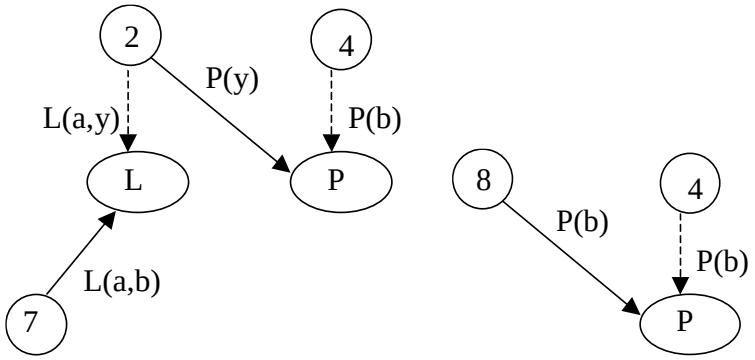


Рис. 4.7. Сеть для набора дизъюнктов – удаление вершины H (а) и L (б)

Проблемы при выводе на сетях аналогичны выводу на логических моделях – комбинаторный рост числа наложений или пересечений и негибкость методов при переходе с одной предметной области на другую.

Представление текстов на естественном языке в виде семантической сети. Семантическая сеть в этом случае рассматривается как множество понятий (слов и словосочетаний), связанных между собой. В нее включаются наиболее часто встречающиеся слова текста, которые несут основную смысловую нагрузку. Для каждого понятия формируется набор ассоциативных (смысловых) связей, т.е. список других понятий, в сочетании с которыми оно встречалось в предложениях текста.

Продолжая уже известный пример из раздела 4.1.3, представим в виде семантической сети нижеприведенный текст (рис. 4.8):

«Авиационный двигатель с системой турбонаддува, в котором использована двухконтурная схема. Двигатель установлен в хвостовой части фюзеляжа. Хвостовое оперение фюзеляжа предусматривает защиту от флаттера».

Обозначения: Х – хвостовое оперение, Ф – фюзеляж, Д – двигатель, А – авиационный, З – защита, ДС – двухконтурная схема, С – система, Т – турбонаддув, ФЛ – флаттер, R_1 – реализует, R_2 – включает, R_3 – контактирует.

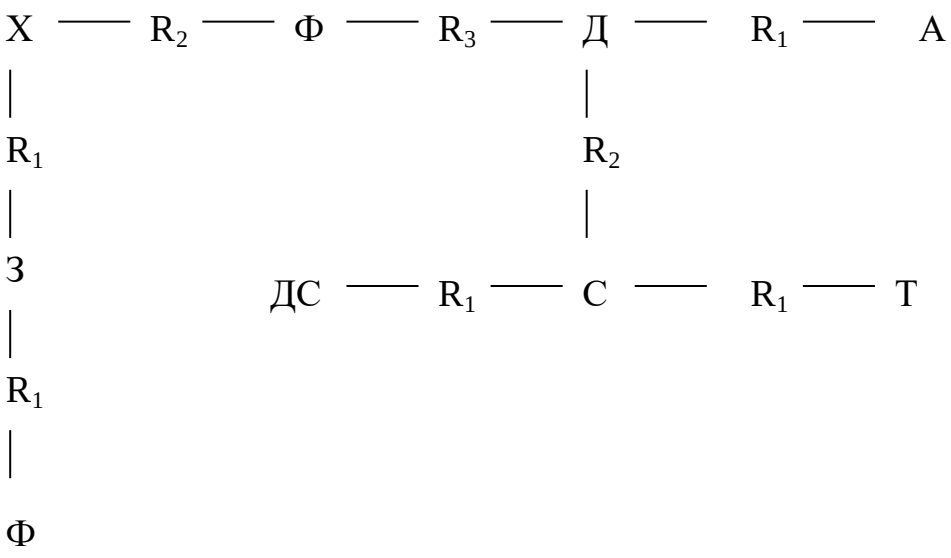


Рис. 4.8. Перевод предложения на семантический язык

Для реализации семантических сетей существуют специальные сетевые языки. Известны экспертные системы, использующие семантические сети в качестве языка представления знаний – PROSPECTOR, CASNBT, TORUS.

4.3.2. Сценарии

Сценарии используются в системах представления знаний для описания известных стандартных (стереотипных) ситуаций реального мира. Организация знаний в виде сценария позволяет:

- восстанавливать информацию, пропущенную в описании ситуации,
- предсказывать появление новых фактов, которых можно ожидать в этой ситуации,
- устанавливать место ситуации в более общем контексте.

Поэтому с содержательной точки зрения сценарий – формализованное описание стандартной последовательности взаимосвязанных фактов, определяющих типичную ситуацию предметной области.

С формальной точки зрения сценарии – вид сетевых моделей, в которых вершинам соответствуют факты, а дугам – отношения специального вида. Эти отношения обладают следующим свойством: если между вершинами *x* и *y* существует множество путей $\pi 1, \pi 2, \ldots, \pi n$ и наличествуют оба факта *a* и *b*, отвечающие вершинам *x* и *y*, то имеет место, по крайней мере, совокупность фактов,

соответствующих вершинам на одном из путей, соединяющих *x* и *y*. Примерами могут служить отношения следующих типов:

- причина – следствие (каузальные сценарии),
- часть – подчасть (дерево целей),
- цель – подцель (классификации),

и т.д. Пример сценария, построенного на отношениях «часть–целое», приведен на рис. 4.9.

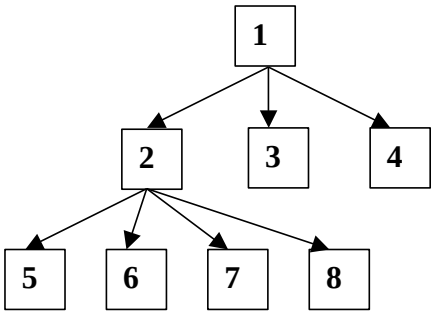


Рис. 4.9. Сценарий типа «часть – целое»: 1 – пылесос, 2 – верхняя часть, 3 – уплотнительное кольцо с фильтром, 4 – нижняя часть, 5 – электродвигатель с вентилятором, 6 – выключатель, 7 – рассеиватель, 8 – шнур

На основании сценариев возможно пополнение знаний о ситуации – обогащение входной информации сведениями, хранящимися в памяти экспертной системы (например, определение цепи событий через одно ключевое событие).

4.3.3. Фреймы

Фрейм – структура данных для представления концептуального (абстрактного) объекта или ситуации. В психологии и философии известно понятие абстрактного образа. Например, слово "комната" вызывает у слушающих следующий образ: жилое помещение с четырьмя стенами, полом, потолком, окнами и дверью, площадью 6-30 м^2. Из этого описания ничего нельзя убрать (например, убрав окна, мы получим уже чулан, а не комнату), но в нем есть "дырки", или "слоты", – это незаполненные значения некоторых атрибутов (количество окон, цвет стен, высота потолка. покрытие пола и др.).

В ИТ фрейм – формализованная модель для отображения такого образа.

Каждый фрейм описывает один объект, а конкретные его свойства и факты, относящиеся к нему, описываются в структурных элементах фрейма – слотах. В общем случае фрейм имеет следующий вид:

```
(Имя фрейма:
        Имя слота 1 (значение слота 1);
        Имя слота 1 (значение слота 1);
        . . .
        Имя слота N (значение слота N)).
```

Значением слота может быть практически что угодно – числа, формулы, тексты на естественном языке, программы, правила вывода, ссылки на другие слоты данного фрейма или других фреймов, а также набор слотов более низкого

уровня, что позволяет организовать многократное вложение. Показано, что между фреймовым и сценарным представлением информации имеется взаимно однозначное соответствие.

Фреймы используются для построения проблемно-ориентированных структур данных. В них фрейм представляется в виде ориентированного графа с помеченными вершинами и дугами. Пример сети фреймов показан на рис. 4.10.

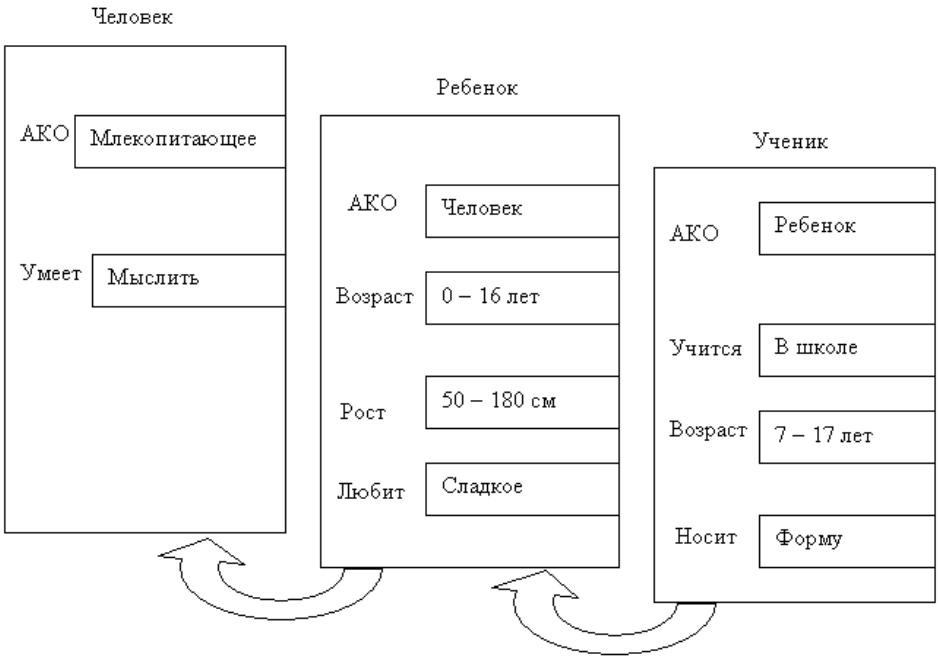


Рис. 4.10. Сеть фреймов

Различают фреймы-образцы, или прототипы, хранящиеся в базе знаний, и фреймы-экземпляры, которые создаются для отображения реальных ситуаций на основе поступающих данных.

Модель фрейма является достаточно универсальной, поскольку позволяет отобразить многообразие знаний о мире через:

- фреймы-структуры для обозначения объектов и понятий (заем, вексель);
- фреймы-роли (менеджер, кассир, клиент);
- фреймы-сценарии (банкротство, собрание акционеров, именины);
- фреймы-ситуации (тревога, авария, рабочий режим устройства) и др.

Важнейшим свойством теории фреймов является заимствованное из теории семантических сетей наследование свойств. И во фреймах, и в семантических сетях наследование происходит по АКО-связям (A-Kind-Of = это). Слот АКО указывает на фрейм более высокого уровня иерархии, откуда неявно наследуются, т.е. переносятся, значения аналогичных слотов.

Например, в сети фреймов на рис. 4.10 понятие "ученик" наследует свойства фреймов "ребенок" и "человек", которые находятся на более высоком уровне иерархии. На вопрос: "Любят ли ученики сладкое?" следует ответ: "Да", так как ним свойством обладают все дети, что указано во фрейме "ребенок". Наследование свойств может быть частичным. Так, возраст для учеников не наследуется из фрейма "ребенок", поскольку указан явно в своем собственном фрейме.

Механизм управления выводом для фреймового представления может быть

организован следующим образом. Связи между данным фреймом и другими фреймами задаются при помощи специального слота, значением которого является присоединенная процедура – специфичная процедура вывода в этом фрейме. При осуществлении вывода вначале запускается одна из присоединенных процедур некоторого фрейма. Затем оценивается возвращаемое значение, и в зависимости от него последовательно запускаются присоединенные процедуры других фреймов. В ходе этого процесса происходит генерация и уничтожение слотов, изменение значений слотов и т.д. Таким образом, происходит постепенное продвижение к получению целевого значения.

Выделим плюсы и минусы фреймовых моделей при представлении знаний:

(+) Фреймы как модели представления знаний отражают концептуальную основу организации памяти человека, обладают гибкостью и наглядностью.

(+) Фреймовые модели особенно эффективны для структурного описания сложных понятий и решения задач, в которых в соответствии с ситуацией желательно применять различные способы вывода.

(-) Здесь затруднено управление завершенностью и постоянством целостного образа. В частности, поэтому существует большая опасность нарушения присоединенной процедуры.

Фреймовые системы без механизма присоединенных процедур (и, следовательно, и механизма пересылки сообщений) часто используют как базу данных системы продукций. Специальные языки представления знаний в сетях фреймов FRL (Frame Representation Language) и другие позволяют эффективно строить промышленные ЭС. Широко известны такие фреймо-ориентированные экспертные системы, как ANALYST, МОДИС.

4.3.4. Продукционные модели

Определения. В общем виде под продукцией понимается выражение вида (*i*); *Q*; *P*; *A*⇒*B*; *N*.

Рассмотрим компоненты продукционной модели более подробно.

Имя продукции *i* может задаваться через лексему (покупка книги) или с помощью номера. Элемент *Q* характеризует сферу применения продукции и близок по значению к ключевому элементу фрейма.

Основной элемент продукции – ее ядро *A*⇒*B*. Стандартная интерпретация ядра: ЕСЛИ А, ТО В. Возможна более сложная конструкция: ЕСЛИ А, ТО В1, ИНАЧЕ В2.

Элемент *P* – условие применимости ядра продукции. Обычно *P* представляет собой логическое выражение (как правило, предикат). Когда P=ИСТИНА, ядро продукции активизируется, например:

```
НАЛИЧИЕ ДЕНЕГ: ЕСЛИ ХОЧЕШЬ КУПИТЬ ВЕЩЬ X, ТО ЗАПЛАТИ В
КАССУ ЕЕ СТОИМОСТЬ И ОТДАЙ ЧЕК ПРОДАВЦУ.
```

Если в этой продукции P=ЛОЖЬ (т.е. денег нет), то применить ядро продукции невозможно.

Элемент *N* – постусловие продукции – актуализируется в случае, если ядро

реализовалось, и содержит действия, которые выполняются после реализации В (в приведенном примере – уменьшить в БД запас товара на 1 единицу).

Управление системой продукций. При использовании продукционной модели база знаний состоит из набора правил. Программа, управляющая перебором правил, называется машиной вывода. Чаще всего вывод бывает *прямой* (от данных к поиску цели) или *обратный* (от цели для ее подтверждения – к данным). Данные – это исходные факты, на основании которых запускается машина вывода. Проблема заключается в том, что в ходе работы машины вывода одновременно актуализируются несколько продукций (до нескольких тысяч), т.е. образуется "фронт готовых продукций". Нужно каким-то образом выбрать из них наиболее важные, т.е. задать стратегию управления выполнением продукций. Возможны несколько таких стратегий.

Принцип "стопки книг" основан на идее, что наиболее часто используемая продукция является наиболее важной. При актуализации некоторого фронта готовых продукций для исполнения выбирается та, у которой частота использования максимальна. Если при этом каждое использование имеет положительную оценку, то таким образом организуется обучающая процедура адаптации информационной системы. Принцип целесообразно применять, когда продукции независимы друг от друга (например, в планировщиках для роботов).

Принцип наиболее длинного условия основан на соображении "здравого смысла": частные правила (относящиеся к узкому классу ситуаций) важнее общих правил, так как несут больше информации. Из фронта готовых продукций выбирается та, у которой стало истинным наиболее длинное условие выполнимости ядра. Целесообразно применять, если знания и сами продукции хорошо структурированы, т.е. привязаны к типовым ситуация, на которых задано отношение "частное – общее".

Принцип метапродукций основан на вводе в систему специальных управляющих метапродукций, например:

ЕСЛИ имеется фронт продукций типа $A \Rightarrow B$ и в A входит выражение X, ТО выполнять продукцию $A(X) \Rightarrow B$.

Принцип "классной доски" основан на идее спусковых функций. В ИС выделяется специальное рабочее поле памяти, в котором параллельно выполняющиеся процессы находят информацию, инициализирующую их запуск, а также выносят информацию о своей работе, полезную для других процессов. Если система продукций работает над некоторой сетевой моделью в базе знаний, то требуется защита от "порчи знаний". Для этого на "классной доске" вводится специальное рабочее поле памяти, куда временно переносится фрагмент знаний, которым оперируют продукции.

Принцип приоритетного выбора – введение приоритетов на продукции:

- статические приоритеты задаются экспертом;
- динамические приоритеты вырабатываются в процессе функционирования системы (например, время нахождения продукции на фронте готовых продукций).

Принцип управления по именам. Для имен продукций задается формальная

грамматика (например, приоритеты или другая алгоритмическая процедура по именам).

Как и при логическом выводе, существуют два типа выполнения систем продукций – прямой (восходящий) и обратный (нисходящий). При поямом выводе проверяются условия А и актуализируются те продукции, для которых А имеет место; при обратном выводе задаются В, ищутся необходимые для них А, и т.д.

Выделим плюсы и минусы продукционных моделей.

(+) Продукции близки к логическим моделям, что позволяет организовывать на них эффективные процедуры вывода, но, как считается, более наглядно отражают структуру организаций знаний человека. Поэтому продукции (наряду с фреймами) – наиболее популярные средства представления знаний в ИС.

(–) Продукционные модели не имеют строгой теории. При задании модели предметной области в виде системы продукций, в отличие от системы предикатов, нельзя быть уверенным в ее полноте и непротиворечивости.

(–) Имеются проблемы при построении систем продукций реального времени (адаптивных), которые бы меняли состав продукций или перестраивали алгоритм управления выбором в зависимости от текущей ситуации.

Продукционная модель чаще всего применяется в промышленных экспертных системах. Она привлекает разработчиков своей наглядностью, высокой модульностью, легкостью внесения дополнений и изменений и простотой механизма логического вывода.

Имеется большое число программных средств, реализующих продукционный подход – языки, "оболочки" или "пустые" экспертные системы, а также промышленные экспертные системы на его основе.

4.3.5. Байесовские сети

Байесовская сеть (или байесовская сеть доверия) – это графическая вероятностная модель, представляющая собой множество переменных и их вероятностных зависимостей.

Содержательно байесовская сеть – это направленный ациклический граф, каждой вершине которого соответствует случайная переменная, а дуги графа кодируют отношения условной независимости между этими переменными. Вершины могут представлять переменные любых типов, быть взвешенными параметрами, скрытыми переменными или гипотезами.

Формально граф является байесовской сетью, если:

- каждой вершине графа поставлена в соответствие случайная переменная из заданного множества случайных переменных **V**;
- дуги в графе удовлетворяют марковскому условию: любая переменная V_i из **V** должна быть условно независима от всех вершин, не являющихся ее потомками;
- заданы все ее прямые родители *parents*(V_i)=$\mathbf{A}_i$ в графе G, то есть для $V_i \in \mathbf{V}$

$$P(V_i \mid \mathbf{pa}_i, \mathbf{s}) = P(V_i \mid \mathbf{pa}_i),$$

где v_i – значение V_i; **S** – множество всех вершин, не являющихся потомками V_i; **s** – конфигурация **S**; $\mathbf{pa}_i$ – конфигурация $\mathbf{PA}_i$.

Тогда полное совместное распределение значений в вершинах можно записать в виде декомпозиции (произведения) локальных распределений:

$$P(V_1, ..., V_n) = \prod_i^n P\left(V_i \middle| parents(V_i)\right).$$

Вероятности в вершинах задаются следующим образом. Если у вершины V_i нет предков, то её локальное распределение вероятностей называют безусловным, иначе условным. Если вершина получила означивание (например, в результате наблюдения), то такое означивание называют свидетельством (evidence). Если значение переменной в вершине было установлено извне (а не наблюдалось), то такое означивание называется вмешательством (action) или интервенцией (intervention).

Чтобы произвести расчет на сетях, каждому компоненту сети соплоставляются меры доверия – к признаку P_X, к правилу P_F, к заключению P_Y. Меры доверия можно измерять средствами теории вероятностей, а также средствами нечеткой логики. Однако при использовании теории вероятности в чистом виде могут возникнуть непреодолимые математические трудности. Поэтому вводятся модифицирующие коэффициенты – коэффициент достаточности LS и коэффициент необходимости LN: LS – коэффициент достаточности посылки E для того, чтобы была верна гипотеза H; LN – коэффициент необходимости посылки E для того, чтобы была верна гипотеза H. Эти коэффициенты позволяют уйти от проблем 0 и ∞ при расчетах с длинными цепочками посылок. Коэффициенты LS, LN и априорные шансы ($O_H^{(0)}$) назначаются экспертным путем. При расчете шансов необходимо добиться их согласованного назначения.

Такие цепочки используются в экспертных системах.

4.4. Средства обработки неопределенности

Противоречие между полнотой и непротиворечивостью модельных представлений о реальном мире, выразившееся в фундаментальных теоремах метаматематики (см. раздел 1), можно интерпретировать и по-другому: в мире есть то, что может быть описано строго, и то, что может быть описано нестрого, т.е. с неопределенностью. В настоящее время является общепризнанным, что все глобальные задачи обработки неопределенной информации могут быть разделены на три больших класса:

- обращение с неполными знаниями – теория нечетких множеств Заде, обработка экспертной информации, интервальные вычисления и пр.;
- комбинирование противоречивых частей информации – теория Демпстера–Шефера и различные еѐ модификации;
- использование грубых (необработанных) массивов данных – теория грубых множеств Павляка.

4.4.1. Нечеткие модели

Теория нечетких (fuzzy) множеств была разработана математиком Лотфи Заде в 1965 г. [Zadeh, 1965].

Нечетким множеством (fuzzy set) F на универсальном множестве U называется совокупность пар ($\mu_F(u)$, u), где $\mu_F(u)$ – степень принадлежности элемента $u \in U$ к нечеткому множеству F. Степень принадлежности – это число из диапазона [0, 1]. Чем выше степень принадлежности, тем в большей мере элемент универсального множества соответствует свойствам нечеткого множества.

Если $\mu_F(u)$ представима в виде функции, то она называется функцией принадлежности (membership function). Функция принадлежности задает как бы "распределение возможностей" того, что данный элемент принадлежит данному множеству (рис. 4.11).

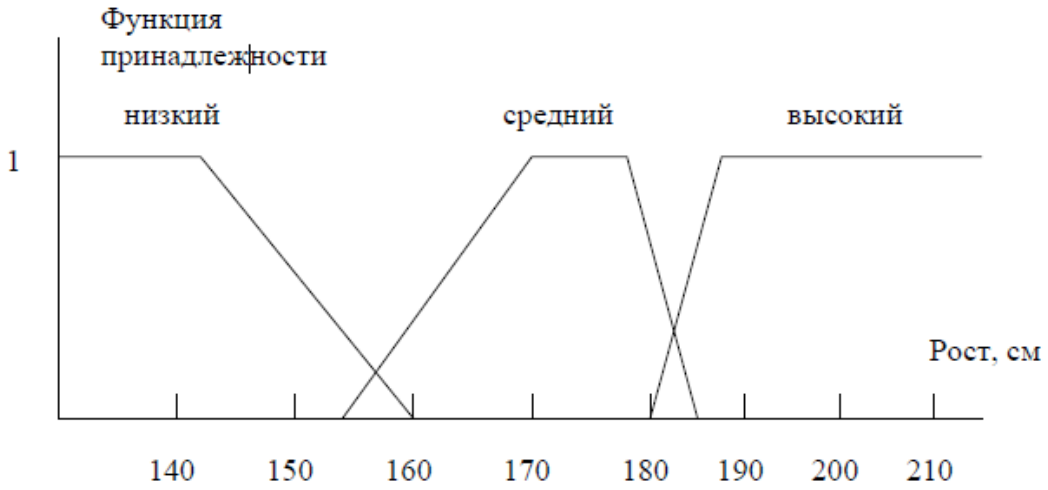


Рис. 4.11. Функции принадлежности для нечетких множеств "мужчины низкого роста", "мужчины среднего роста", "мужчины высокого роста"

Лингвистической переменной (linguistic variable) называется переменная, значениями которой могут быть слова или словосочетания некоторого естественного или искусственного языка.

Терм-множеством (term set) называется множество всех возможных значений лингвистической переменной.

Термом (term) называется любой элемент терм-множества. В теории нечетких множеств терм формализуется нечетким множеством с помощью функции принадлежности.

Носителем нечеткого множества F называется четкое подмножество универсального множества U, элементы которого имеют ненулевые степени принадлежности:.$supp(F)=\{u: \mu_F(u)>0\}$. Например, на рис. 4.11 представлена переменная “рост человека”, которая оценивается по шкале “низкий", "средний", "высокий”. В этом примере лингвистической переменной является “рост человека”, термами – лингвистические оценки “низкая", "средняя", "высокая”, которые и составляют терм-множество.

Нечеткой базой знаний (fuzzy knowledge base) о влиянии факторов $x_2, \ldots, x_n\}$ на значение параметра y называется совокупность логических высказываний следующего типа:

ЕСЛИ $\left(x_1 = a_1^{j1}\right)$ И $\left(x_2 = a_2^{j1}\right)$ И ... И $\left(x_n = a_n^{j1}\right)$

ИЛИ $\left(x_1 = a_1^{j2}\right)$ И $\left(x_2 = a_2^{j2}\right)$ И ... И $\left(x_n = a_n^{j2}\right)$

...

ИЛИ $\left(x_1 = a_1^{jk_j}\right)$ И $\left(x_2 = a_2^{jk_j}\right)$ И ... И $\left(x_n = a_n^{jk_j}\right)$

ТО $y = d_j$, для всех $j = 1, \ldots, m$.

Здесь a_i^{jp} – нечеткий терм, которым оценивается переменная x_i в строчке с номером jp ($p = 1,...,k_j$); k_j – количество строчек-конъюнкций, в которых выход y оценивается нечетким термом d_j, $j = 1,...,m$; m – количество термов, используемых для лингвистической оценки выходного параметра y.

С помощью операций $\bigcup$ (ИЛИ) и $\bigcap$ (И) нечеткую базу знаний можно переписать в более компактном виде:

$$\bigcup_{p=1}^{k_j}\left[\bigcap_{i=1}^{n}\left(x_i = a_i^{jp}\right)\right] \to y = d_j,\ j = 1,...,m.$$

Тогда *нечетким логическим выводом (fuzzy logic inference)* называется аппроксимация зависимости $y = f(x_1, x_2, \ldots, x_n)$ с помощью нечеткой базы знаний и операций над нечеткими множествами.

Используя нечеткие переменные x_i на входе, мы получаем нечеткую оценку результата y на выходе.

В реальной практике вычислений форма функций принадлежности обычно упрощается. Например, они могут быть представлены треугольными нечеткими числами. Для преобразования оценок в виде треугольных нечетких чисел в четкие числа (дефаззификации) разработано несколько методов, в том числе методы средних максимумов, центра суммы, центра тяжести, α-среза. Например, метод α-среза определяет нечеткое число $x=(a_1, a_2, a_3)$ через его α-срезы $\left[a_1^{(\alpha)}, a_3^{(\alpha)}\right]$ следующим образом:

$$x_\alpha = \left[a_1^{(\alpha)}, a_3^{(\alpha)}\right] = \left[\left(a_2 - a_1\right)\alpha + a_1, -\left(a_3 - a_2\right)\alpha + a_3\right],$$

т.е. α-срезы формируют своего рода доверительный интервал, в котором располагается дефаззифицированное нечеткое число. Можно также работать непосредственно с парой чисел, определяющих функцию принадлежности на каком-то ее уровне (срезе) – это интервальный подход.

Нечеткие системы были очень популярны в 80-х гг., особенно при построении систем управления. Сегодня подход нечетких чисел используют для теоретического обоснования результатов работы системы, если входные данные для нее являются экспертными. В целом теория нечетких множеств Л. Заде считается наиболее общей из существующих сегодня моделей описания неопределенности.

4.4.2. Модели на основе логики Демпстера-Шафера

Теория была предложена Артуром П. Демпстером [Dempster, 1968] и развита Гленном Шафером [Shafer, 1976].

Пусть X – универсальное множество, которое представлядет собой набор всех рассматриваемых утверждений. Вводится показательное множество $P(X)$ – совокупность всех подмножеств множества X, включая пустое множество $\varnothing$. Например, если

$$X = \{a, b\},$$

то

$$P(X) = \{\varnothing, \{a\}, \{b\}, X\}.$$

Вводится масса $m(A)$ элемента показательного множества A, которая выражает соотношение всех доступных свидетельств, которые поддерживают утверждение, что определённый элемент X принадлежит A, но не принадлежит ни одному подмножеству A. При этом величина $m(A)$ относится только к множеству A и не создаёт никаких дополнительных утверждений о других подмножествах A, каждое из которых, по определению, имеет свою собственную массу.

Используя значения $m(A)$, можно определить верхнюю и нижнюю границы интервала возможностей. Этот интервал содержит точную величину вероятности рассматриваемого подмножества (в классическом смысле); он ограничен двумя неаддитивными непрерывными мерами, называемыми доверием (belief) (или поддержка (support)) и правдоподобием (plausibility):

$$bel(A) \le P(A) \le pl(A),$$

причем эти две меры соотносятся между собой следующим образом:

$$pl(A) = 1 - bel(\bar{A}).$$

Содержательно введенные величины имеют следующую интерпретацию: доверие к гипотезе представляет собою сумму масс свидетельств, поддерживающих гипотезу, а правдоподобие гипотезы отличается от абсолютной достоверности на сумму масс всех свидетельств, противоречащих гипотезе. Такие оценки вполне соответствуют практике «здравого смысла» и позволяют интерпретировать доверие и правдоподобие как границы интервала возможного значения истинности гипотезы:

доверие ≤ какая-то мера истинности ≤ правдоподобие.

Подход, принятый в теории Демпстера-Шефера, отличается от байесовского подхода и метода коэффициентов уверенности тем, что, во-первых, здесь используется не точечная оценка уверенности (коэффициент уверенности), а интервальная оценка. Такая оценка характеризуется нижней и верхней границей, что более надежно. Во-вторых, теория позволяет исключить взаимосвязь между неопределенностью (неполнотой знаний) и недоверием, которая свойственна байесовскому подходу.

В целом теория Демпстера-Шафера основана на строгом научном фундаменте и широко используется в экспертных системах.

4.4.3. Модели на основе грубых множеств

Теория приближенных (неточных, грубых) множеств (rough sets) была разработана [Pawlak, 1982] как новый математический подход для описания неопределенности, неточности и неуверенности. В отличие от подхода нечетких множеств, теория грубых множеств применяется не к одному атрибуту, представленному носителем нечеткого множества, а к набору атрибутов, представленному таблицей. рассматриваемой как информационная система.

Главная концепция теории грубых множеств заключается в том, что знания находят отражение в разделении (классификации) элементов универсума. Это разделение может пониматься как семантика представления знаний. Однако, чтобы успешно выявить семантику знаний, эти знания должны быть представлены в подходящей синтаксической форме. Такой формой является таблица данных, строки которой соответствуют элементам (объектам), а столбцы – признакам (атрибутам) этих элементов. В ячейке на пересечении *i*-й строки и *j*-го столбца отображается значение *j*-го атрибута для *i*-го элемента. В практических приложениях теории грубых множеств такие таблицы принято называть информационными системами. Если в ней явно выделен один атрибут, то она называется таблицей принятия решений.

Таблица принятия решений (decision table) – это тройка T = (U, C, D), где U – это множество объектов, C – это множество атрибутов условий (condition attributes), D – это множество атрибутов решений (decision attributes).

По существу, каждая строка таблицы принятия решений – это продукция (или логическое следование – в зависимости от строгости задания), а вся таблица – это база знаний.

Таблица. 4.1. Пример таблицы принятия решений

	C		D
X	**возраст**	**Двигательная активность нижних конечностей**	**Способность ходить**
x_1	16–30	50	Yes
x_2	16–30	0	No
x_3	31–45	1–25	No
x_4	31–45	1–25	Yes
x_5	46–60	26–49	No
x_6	16–30	26–49	Yes
x_7	46–60	26–49	No

Рассмотрим пример таблицы принятия решений (табл. 4.1).

Легко видеть, что представленные в этой таблице данные, например x3, x4 – противоречивые, а x5, x7 повторяются. Но можно, не вводя никаких экспертных мнений (это – главная особенность подхода), сделать различные разбиения множества X. Например, разбиение множества X в соответствии со значениями атрибута **возраст** имеет вид:

$\mathbf{X}_{age}$ = {{x1, x2, x6}, {x3, x4}, {x5, x7}}

На основании этого факта вводятся базовые концепты теории грубых множеств. Нижняя аппроксимация множества **X** включает в себя элементы, которые действительно (по всем атрибутам) принадлежат множеству X. Верхняя аппроксимация множества **X** + включает в себя элементы, которые, возможно (по каким-то атрибутам), принадлежат множеству **X**, т.е. сюда входят и противоречивые строки таблицы. Граница (разница между верхней и нижней аппроксимацией) представляет собой область неразличимости.

Введенные концепты графически проиллюстрированы на рис. 4.12.

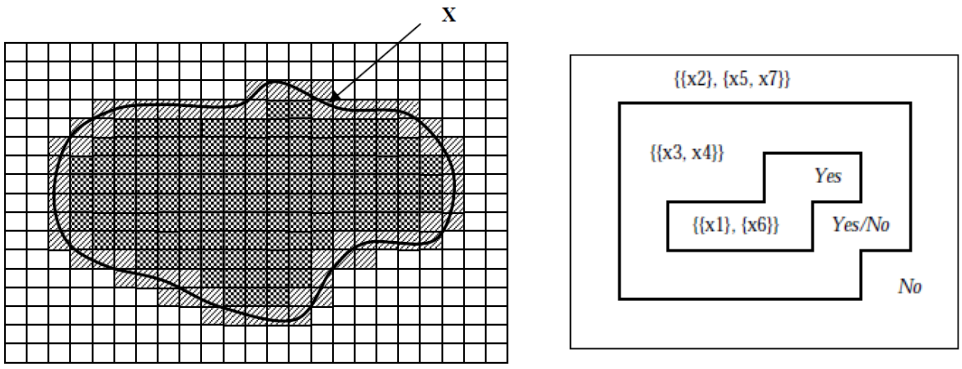


Рис. 4.12. Базовые концепты теории грубых множеств

Используя эти концепты, можно ввести количественную (т.е. объективную, а не экспертную) оценку параметров приближенных множеств. Например, для оценки качества аппроксимации множества *X* признаком B можно использовать соотношение верхней и нижней аппроксимации:

$$\alpha_B(\mathbf{X}) = \frac{\left|B_{\text{нижн}}(\mathbf{X})\right|}{\left|B_{\text{верх}}(\mathbf{X})\right|},$$

где знак модуля означает кардинальность множества.

Аналогично можно ввести обоснование для выделения значимых атрибутов. Самый простой вариант – считать значимым только ядро, т.е. нижнюю аппроксимацию множества **X** – слишком жесткое условие. Пусть мы имеем два подмножества атрибутов *B* и *C*, и на основе этих подмножеств произведено разделение элементов универсума. Мы можем выделить положительную область разделения C на основе разделения B: $POS_B(C)$. Содержательно эта область содержит те элементы разделения C, которые могут быть корректно классифицированы, используя разделение на основе подмножества B. Тогда степень зависимости между двумя разделениями можно оценивать по выражению

$$\gamma_B(\mathbf{C}) = \frac{\left|POS_B(\mathbf{C})\right|}{\left|\mathrm{X}\right|}.$$

Эту идею можно развивать дальше. Пусть из подмножества **B** удален один атрибут или некоторое подмножество атрибутов **B'** и произведено разделение элементов на основе подмножества атрибутов (**B** – **B'**). Тогда в качестве оценки значимости подмножества атрибутов **B** принимается разность

$$\gamma_B(\mathbf{C}) - \gamma_{B-B'}(\mathbf{C}).$$

Содержательно эта разность интерпретируется как степень изменения положительной области $POS_B(\mathbf{C})$ при переходе от подмножества атрибутов **B** к подмножеству **B** – **B'**. Аналогично вводятся сечения (cuts) и логические правила для анализа подмножеств признаков. Теория грубых множеств позволяет также определять степень взаимозависимости атрибутов более формально, чем традиционные теоретико-информационные меры, а также выводить правила из неполностью определенных данных [Grzymala-Busse, 2007].

Методы грубых множеств могут применяться как компонент гибридных решений в области машинного обучения и интеллектуального анализа данных.

4.5. Онтологические модели

Определения и классификации онтологий. Идею онтологии в информатике выдвинул Томас Грубер [Gruber, 1993]. Он предложил описывать знания двумя способами: в канонической форме, которая представляет собой описание знаний на языке логики предикатов (например, в виде фактов языка Prolog), или в форме онтологии, которая представляет собой множество классов, связанных между собой отношением обобщения (обратным для отношения наследования). Таким образом, онтология по Груберу представляет собой описание декларативных знаний, сделанное в виде классов с отношением иерархии между ними.

Существует другая трактовка, более близкая к области информационных технологий [Guarino, 1995]: онтология – выраженная формальными средствами спецификация концептуализации. В свою очередь, концептуализация – это переход от описания предметной области на естественном языке к описанию в виде кортежа [Гаврилова, 2000]:

$O = <X, R, Ф>$,

где X – конечное множество концептов (понятий, терминов) предметной области, которую представляет онтология O; R – конечное множество отношений между концептами (понятиями, терминами) заданной предметной области; F – конечное множество функций интерпретации (аксиоматизации).

Таким образом, онтология на сегодняшний день является широким понятием и содержательно покрывает различные виды формализаций предметной области. Но как минимум в онтологии должны присутствовать следующие структурные элементы: множество объектов (концептов, понятий); алфавит отношений; правила установления отношений; аксиомы, задающие правила вывода на множестве отношений.

Как видно из определений, термину «онтология» удовлетворяет широкий спектр структур, представляющих знания о той или иной предметной области. В связи с этим онтологии разделяются на фундаментальные (fundamental ontologies), которые описывают предметную область максимально полно, безотносительно к приложениям и обычно с максимальной степенью формализации, и прикладные (application ontologies), которые также называются «легкими» онтологиями (lightweight ontologies) и формализуются настолько, насколько это необходимо для конкретного приложения. В качестве в разной степени формализованных онтологий могут рассматриваться словарь с определениями, простая

таксономия, модель с произвольным набором отношений. Например, при наложении ограничений R=0 и F=0 онтология O трансформируется в простой словарь:

$O = V = <X , \{\}, \{\}>$

С точки зрения использования онтологий в задачах автоматической обработки текста выделяются онтологии предметной области и лингвистические онтологии.

При построении онтологии предметной области сначала создается система понятий, которым затем приписываются наборы языковых выражений (слов, терминов, словосочетаний). С другой стороны, существующие лингвистические ресурсы (словари, глоссарии, тезаурусы) также задают определенную концептуализацию предметной области. Такого рода онтологии называются лингвистическими онтологиями.

В качестве примера лингвистической онтологии часто приводится ресурс WordNet, который представляет в виде иерархической структуры систему значений слов общезначимого английского языка. Целью его разработки являлось не описание системы понятий, а установление системы отношений между лексическими значениями. Наиболее ярко различие между описаниями лексики и иерархии понятий в ресурсах типа WordNet проявляется в расчленении иерархической сети на подсети по частям речи, когда совпадающим по значению, но различающимся по частям речи словам (например, приватизация, приватизировать, приватизационный) соответствуют разные узлы иерархической сети. Ясно, что понятие, соответствующее этим словам, должно быть одно и то же.

Однако в конкретных предметных областях значения предметной лексики и понятия предметной области максимально сближаются.

Как правило, в системах ИИ конкретная онтология $О_{пр}$ используется не сама по себе, а в расширенном виде, т.е. в рамках сети онтологий $О_{расширен}$:

$О_{расширен} = <О_м <О_{пр}, О_з>>$.

Здесь $О_м$ – метаонтология (онтология верхнего уровня), которая описывает наиболее общие понятия, не зависящие от предметных областей («объект», «свойство», «значение» и т.д.); $О_{пр}$ – онтология предметной области, т.е. формальное описание предметной области для определения общей терминологической базы; $О_з$ – онтология конкретной задачи, где в качестве понятий выступают типы решаемых задач, а отношения специфицируют разделение задачи на подзадачи.

Сетевые онтологии часто используют для описания конечных результатов действий, выполняемых объектами предметной области или задачи.

Требования к типам связей в онтологиях. Чтобы стать онтологией, сетевая модель должна приобрести, как учит философия, онтологический статус, т.е. в максимально широком контексте применения соответствовать закономерностям и ограничениям описываемой предметной области.

Следовательно, онтологии должны строиться в расчете на неопределенный контекст, т.е. на такие условия, когда ни об одном выявленном в тексте понятии не будет точно и полно известен даже набор явно упоминаемых о нем в тексте фактов и других видов информации. Поэтому нужны такие связи, которые

максимально не зависят от контекста – не исчезают и не меняются в течение всего срока существования любого или подавляющего большинства экземпляров понятия.

Наиболее известным типом отношения, которое выполняется для всех экземпляров понятия, является таксономическое отношение. Например, любой лес всегда состоит из деревьев, т.е. связь «состоит из» можно считать онтологичной. Желательно найти и другие типы отношений, обладающие, подобно таксономическим отношениям, свойствами транзитивности и наследования. В [Guarino, 1995] они выделены как отношения онтологической зависимости, изучаемые в рамках философской дисциплины «формальная онтология». Отношения онтологической зависимости описывают, подразумевает ли существование одного понятия существования каких-либо других понятий.

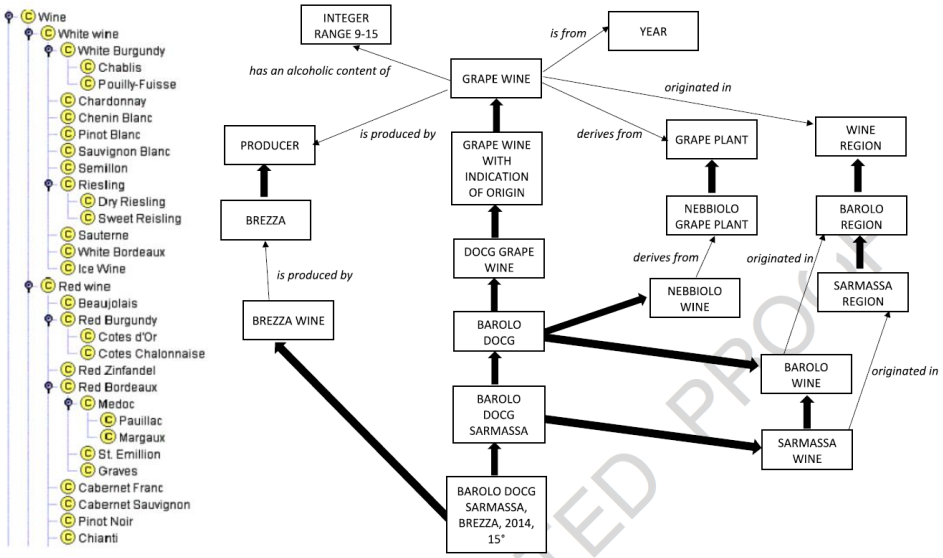
Эти отношения подразделяются на следующие виды:

- строгая зависимость (rigid dependence) имеет место тогда, когда существование сущности подразумевает существование чего-либо еще. Например, *кипение* невозможно без существования конкретного объема жидкости, которая кипит;
- родовая зависимость (generic dependence) имеет место тогда, когда предполагается существование примеров некоторого класса некоторых сущностей. Например, возникновение понятия *гараж* невозможно без существования понятия *автомобиль,* хотя конкретный гараж может возникнуть безотносительно к конкретному автомобилю;
- историческая зависимость (historical dependence) имеет место тогда, когда существование X в некоторый момент времени T предполагает существование Y в некоторый другой момент времени T1. Например, *солома* исторически зависит от *молотьбы*, поскольку *солома* не может возникнуть без предварительного процесса *молотьбы*, вместе с тем эти работы заканчиваются, а солома длительное время продолжает существовать.

Создание онтологий. Процесс создания онтологий, достаточно широко представленный в литературе [Добров, 2020; Ной, 2007], является предметом изучения в дисциплине «Инженерия знаний». Для построения систем ИИ существенными являются фундаментальные правила разработки онтологии [Ной], кратко охарактеризованные ниже.

Во-первых, понятия в онтологии должны быть близки к объектам (физическим или логическим) и отношениям в выбранной предметной области. При разработке онтологий на основе текстовых описаний понятиям (объектам) чаще всего соответствуют существительные, а отношениям – глаголы.

Во-вторых, не существует единственного правильного способа моделирования предметной области. Лучшее решение почти всегда зависит от предполагаемого приложения и ожидаемых расширений. Например, два варианта высокоуровневой онтологии вин представлены на рис. 4.13, а [Ной, 2007], и 4.13, б [Amoretti, 2020]. Хотя в обоих случаях рассматривается одна и та же предметная область (виноделие в близко расположенных регионах – Франции и Италии), сравнение рисунков показывает существенные различия в выборе базовых понятий и отношений.



а б

Рис. 4.13. Онтологии вин

Наконец, разработка онтологии – это, как правило, итеративный процесс. При этом целесообразно использовать уже готовые модели, расширяя их структуру за счет добавления новых элементов или производя их интеграцию посредством слияния (merging) или отображения (mapping). Слияние и интеграция онтологий определяется как построение новой онтологии из нескольких онтологий, состоящих из перекрывающихся подонтологий. Отображением онтологий называют связывание сходных (по некоторой метрике) понятий или отношений из разных источников друг с другом путем указания соответствия между ними.

Например, для диагностирования такого сердечно-сосудистого заболевания, как синдром ранней реполяризации желудочков (РРЖ), базовая онтология электрокардиограмм [Gonçalves, 2011] (рис. 4.14, а) была модифицирована (рис. 4.14, в) в части, регистрирующей информацию об ударе сердца (рис. 4.14, б). Для этого добавлены:

сущность, фиксирующая, можно ли выставить диагноз:

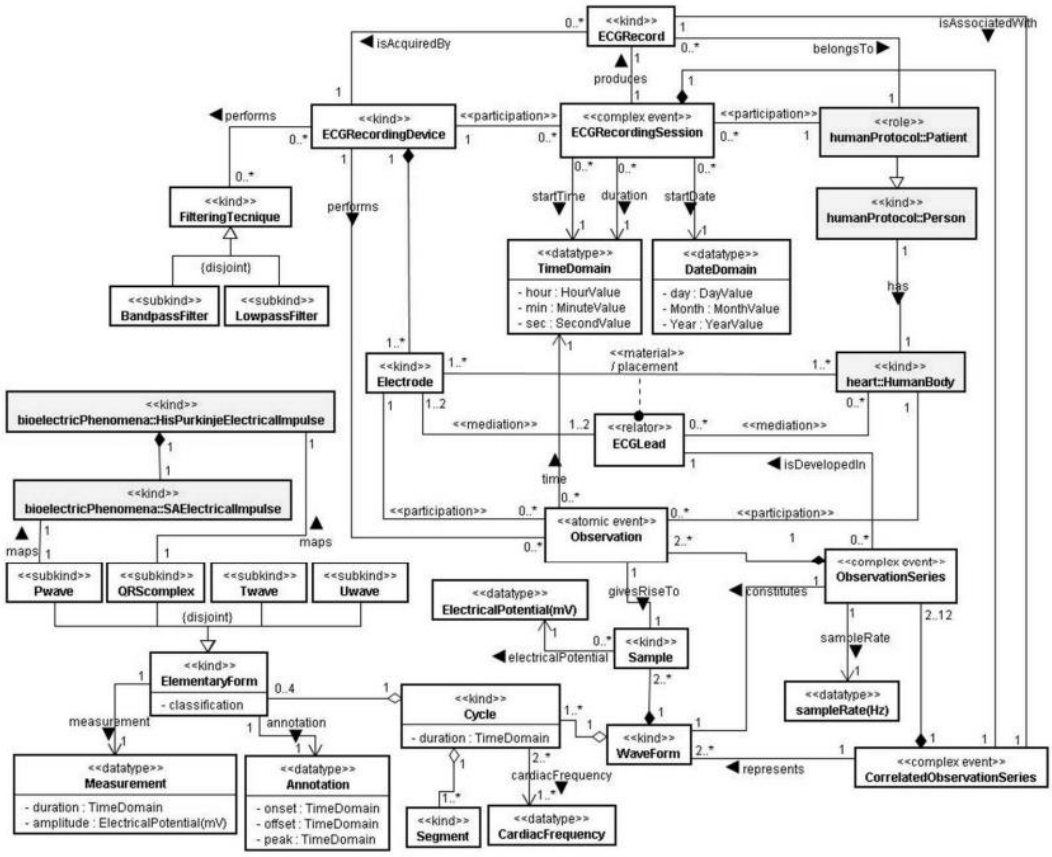
```
DiagnosisERV
isGiven: Bool
```

сущность, в которую записываются полученные отклонения:

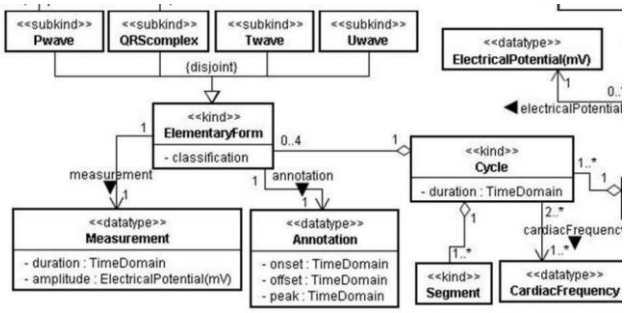
```
PathologyChecker
jElevation: JEleveationDomain
patientHasAnemy: Bool
ECGDiversion: DiversionDomain
```

Для нее использованы новые типы данных:

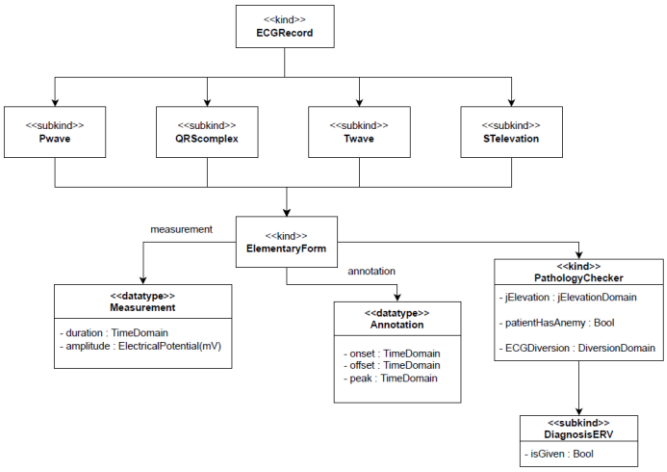
```
«datatype»
DiversionDomain
- amount: DiversionAmountValue(amount)
«datatype»
JElevationDomain
- diversion: DiversionValue
```



а



б



в

Рис. 4.14. Расширение структуры онтологии электрокардиограмм: а – исходная онтология; б – фрагмент онтологии, связанный с информацией об ударе сердца; в – модификация онтологии для описания синдрома РРЖ

4.6. Нейросетевые модели

Глубокие НС (ГНС) занимают доминирующие позиции в списке технологий ИИ. В разделе 2.4.1 представлены история появления, базовые структурные компоненты и первые архитектурные решения ГНС, такие как LeNet [Lecun, 1998], AlexNet [Krizhevsky, 2012], VGG [Simonyan, 2015] и GoogLeNet [Szegedy, 2014], ResNet [He, 2016] и UNet [Ronneberger, 2015]. Если архитектуры LeNet и AlexNet имеют, скорее, историческое значение, то архитектуры семейств VGG, ResNet и UNet сегодня находят широкое применение в качестве базы для разработки различных систем ИИ. Но развитие нейросетевых моделей в последние годы идет, скорее, по пути реализации посредством архитектуры ГНС принципиально новых концепций обработки информации. Некоторые примеры расширения «зоопарка» современных ГНС, находящие применение в первую очередь в медицинских системах ИИ, представлены ниже.

4.6.1. Рекуррентные сети

Как показывает анализ развития НС, простое увеличение объема используемых вычислительных ресурсов в архитектуре НС уже не обеспечивает существенного улучшения выразительных свойств. Теперь мысль разработчиков, скорее, идет в направлении использования контекста в ГНС. Тут есть несколько путей, первым из которых мы рассмотрим рекуррентные нейронные сети (РНС).

РНС (англ. Recurrent neural network; RNN) – вид нейронных сетей, где связи между элементами образуют направленную последовательность. Благодаря этому появляется возможность обрабатывать серии событий во времени или последовательные пространственные цепочки. В отличие от многослойных перцептронов, рекуррентные сети могут использовать свою внутреннюю память для обработки последовательностей произвольной длины. Поэтому сети RNN применимы в таких задачах, где нечто целостное разбито на части, например распознавание рукописного текста или распознавание речи. Было предложено много различных архитектурных решений для рекуррентных сетей – от простых до сложных. В последнее время наибольшее распространение получили сеть с долговременной и кратковременной памятью (LSTM) и управляемые рекуррентные сети (GRU).

Все РНС имеют форму цепочки повторяющихся модулей нейронной сети. В стандартных РНС этот повторяющийся модуль имеет очень простую структуру – один слой с функцией активации tanh (рис. 4.15). Такие конструкции были придуманы достаточно давно: в 1982 г. были предложены сети Хопфилда [Hopfield, 1982], в 1993 году – НС «очень глубокого обучения», в которой в рекуррентной сети разворачивалось более 1000 последовательных слоёв. Но у такой архитектуры есть принципиальная проблема: с ростом длины необходимого контекста ее устойчивость падает.

Примеры:

- пытаемся предсказать последнее слово в конструкции «облака в …» –очевидно, что следующим словом будет «небо», т.е. разрыв между источником информации и местом, в котором она нужна, невелик;
- пытаемся предсказать последнее слово в конструкции «Я вырос во Франции… я бегло говорю по-…»; если учесть три предыдущих слова, то понятно, что следующим словом будет название языка; если же учесть шесть предыдущих слов, то получим более точное предсказание – «французски».

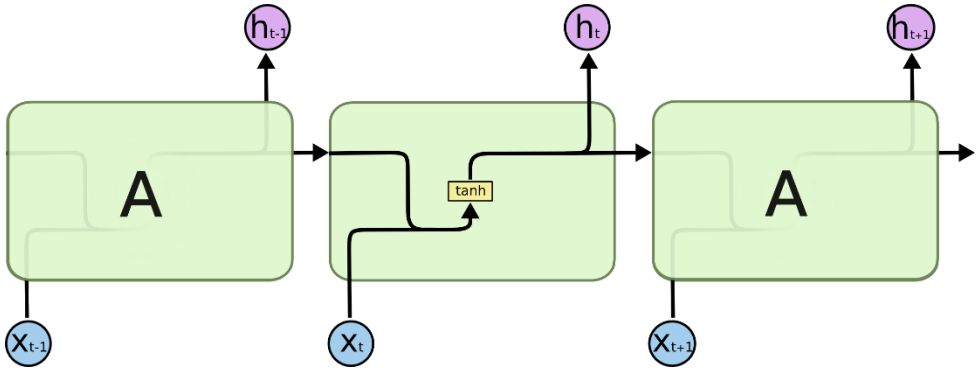


Рис. 4.15. Стандартная RNN

Принципиально лишены этого недостатка сети с долговременной и кратковременной памятью (LSTM) [Hochreiter, 1997]. LSTM (рис. 4.16) также представляют собою цепь, но в повторяющемся модуле вместо одного слоя нейронной сети существует четыре, взаимодействующих совершенно особым образом.

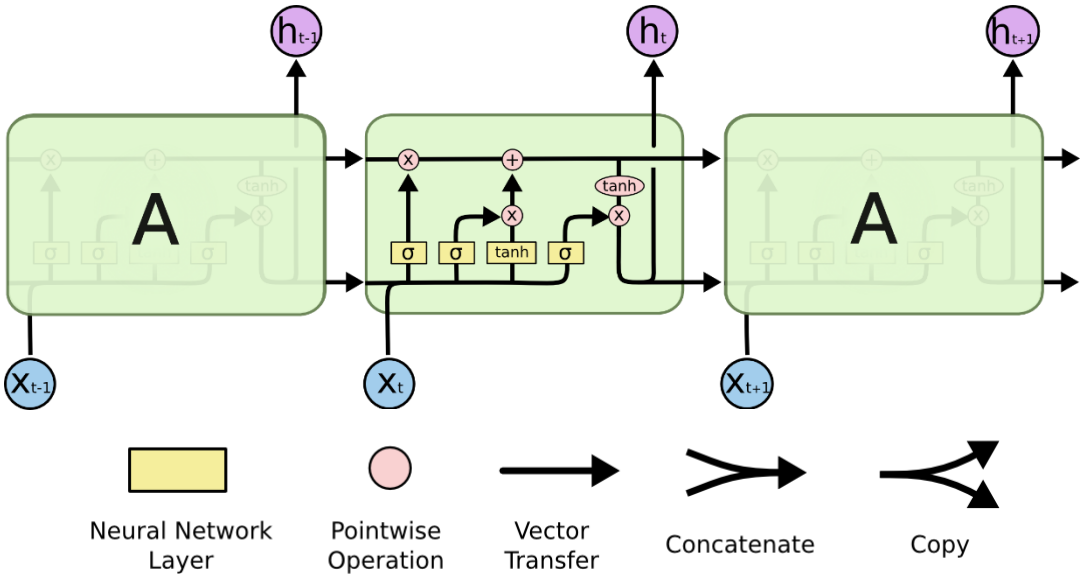


Рис. 4.16. Сеть LSTM

Ключом к LSTM является состояние ячейки – горизонтальная линия, проходящая через верх диаграммы. Кроме того, у LSTM есть возможность удалять или добавлять информацию о состоянии ячейки; для этого используются структуры под названием gates (шлюзы, вентили, ворота) (рис. 4.17). Они состоят из сигмоидного слоя нейронной сети и операции скалярного произведения, при этом

сигмоидный слой выводит числа от нуля до единицы, описывающие, какая доля каждого компонента должна быть пропущена.

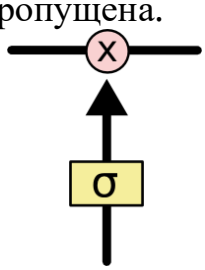
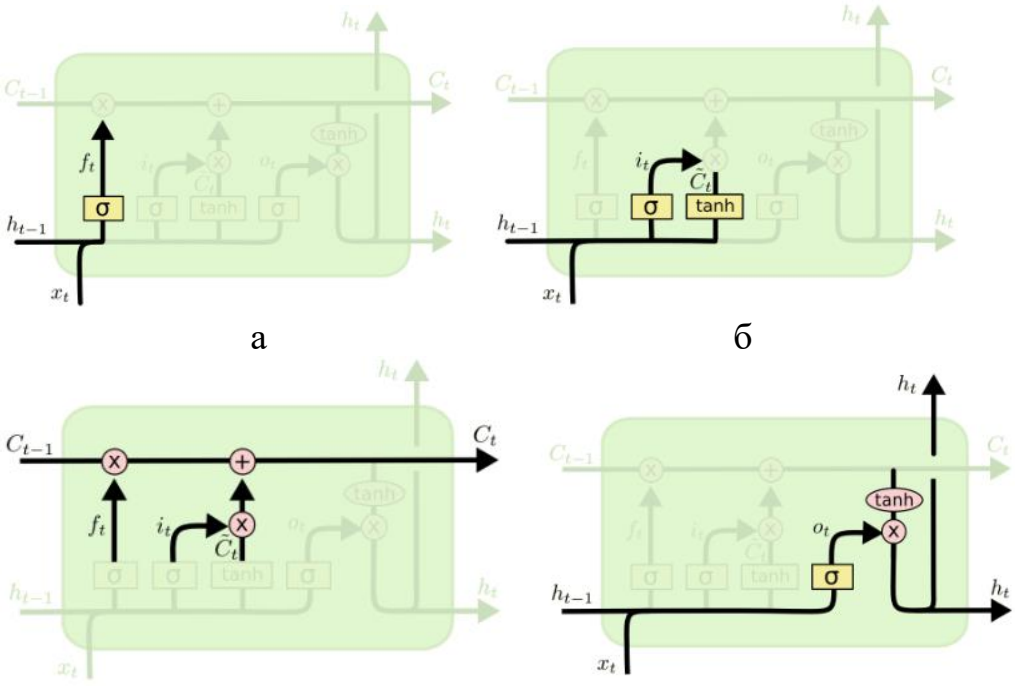


Рис. 4.17. Структура шлюза

В LSTM существует три типа шлюзов (рис. 4.18). Шлюз забывания (forget gate layer) (рис. 4.18, а) смотрит на значения h_{t-1} и x_t и выдает степень забывания от 0 до 1. Шлюз памяти (input gate layer + block input) (рис. 4.18, б, в) решает, какую новую информацию хранить в состоянии ячейки. Сигмоидный слой, называемый входным слоем затвора (input gate layer), решает, какие значения будут обновляться. Затем слой block input (в нашем случае tanh) создает вектор новых значений-кандидатов, Ct, которые могут быть добавлены к состоянию. Для реализации всех принятых решений умножаем старое состояние на fT, тем самым забывая то, что мы решили забыть ранее, и добавляем It×Ct в зависимости от того, насколько мы решили обновить каждое значение состояния. В выходном шлюзе (output gate) (рис. 4.18, г) сигмоидный слой определяет, какая часть состояния ячейки будет выводиться; параллельно слой tanh нормирует вывод между -1 и 1; затем все перемножается.



а б

в г

Рис. 4.18. Типы шлюзов: а – шлюз забывания, б, в – шлюз памяти, г – выходной шлюз

Управляемые рекуррентные сети (gated recurrent units, GRU) (рис. 4.19), по сути, представляют собой облегченную версию LSTM. Ячейки имеют такой же

объем памяти, но обработка информации идет несколько другим путем, слегка «обрезанным» по функциональности (отсутствует выходной вентиль).

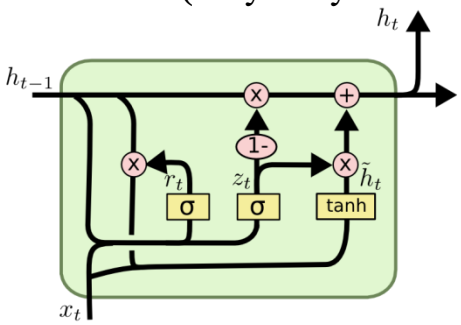


Рис. 4.19. Управляемая рекуррентная сеть (GRU)

Преимущества и недостатки RNN [Sharkawy, 2020]:

- RNN способны аппроксимировать произвольные нелинейные динамические системы (в условиях слабой регулярности) с произвольной точностью, реализуя сложные отображения из входных последовательностей в выходные последовательности;
- RNN – достаточно простая и мощная модель;
- RNN трудна в настройке (присутствует проблема исчезающего градиента) и стабилизации;
- RNN не могут обрабатывать очень длинные последовательности при использовании tanh или relu в качестве функции активации.

4.6.2. Байесовские нейронные сети

Определение, структура и реализация БНС. Байесовские нейронные сети (БНС) – это стохастические сети, обучаемые на основе байесовского подхода [Goan, 2020: Jospin, 2022].

Традиционная НС состоит из входного слоя l_0, последовательности скрытых слоев l_i, $i = 1, \ldots, n-1$, и выходного слоя l_n, причем каждый слой, кроме последнего, выполняет линейную трансформацию, а затем нелинейное преобразование посредством функции активации s; таким образом, вся сеть выполняет следующие преобразования:

$l_0 = x$,

$l_i = s_i(W_i l_{i-1} + b_i)\ \forall i \in [1, n]$,

$y = l_n$,

где W – веса связей, b – смещения. Для воспроизведения произвольной функции $y = \Phi(x)$ традиционная нейронная сеть выполняет регрессию параметров $\theta = (W, b)$ на обучающих данных $D(x, y)$, где D состоит из ряда входных данных x и соответствующих им меток y.

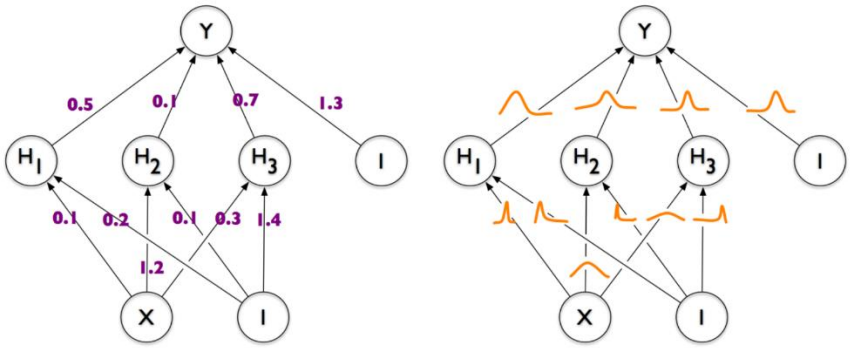
Стохастические НС строятся путем введения в обычную архитектуру стохастических компонентов – либо стохастической функции активации s, либо стохастического вектора параметров:

$\theta \sim p(\theta)$,

в результате функция Ф становится аппроксимацией значения *y* при наличии случайной шумовой компоненты ε:

$y = \Phi_\theta(x) + \varepsilon$.

Пример такого перехода показан на рис. 4.20. Слева – обычная НС, у которой каждая связь между парой нейронов задана каким-то числом (весом). Справа – байесовская нейронная сеть, веса которой представлены не числами, а облаками вероятности. Если такая сеть делает аппроксимацию набора точек (рис. 4.21), то получается набор кривых с некоторой вероятностью распределения.



а б

Рис. 4.20. Сравнение обычной (а) и байесовской (б) НС

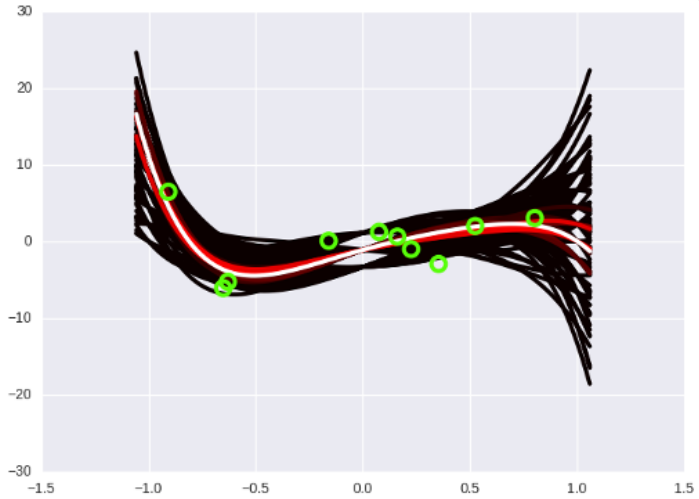


Рис. 4.21. Регрессия с помощью байесовской НС

В случае байесовской НС $p(\theta)$ в соответствии с теоремой Байеса можно записать:

$$p\left(\theta \middle| D\right) = \frac{p\left(\theta_y \middle| D_x, \theta\right) p(\theta)}{\int p\left(\theta_y \middle| D_x, \theta'\right) p(\theta') d\theta'} \sim p\left(D_y \middle| D_x, \theta\right) p(\theta),$$

однако даже в этом случае непосредственный расчет регрессии крайне затруднен. На практике при построении байесовских НС используют упрощающие подходы – метод Монте-Карло и вариационный вывод. В обоих случаях мы пытаемся восстановить неизвестное распределение $p(\theta)$, но в первом случае запускаем итеративную процедуру сэмплирования значений $p(\theta)$, а во втором случае

предполагаем, что $p(\theta)$ описывается типовым распределением (например, гауссоидой), но с неизвестными параметрами, и находим их вариационным методом.

На рис. 4.22 представлен один из вариантов метода Монте-Карло – алгоритм Метрополиса. Алгоритм стартует с некоторого случайного начального значения θ_0, а затем выбирается новая точка-кандидат θ' в окрестности предыдущей θ, используя предполагаемое распределение $Q(\theta'|\theta)$. Если θ' более вероятно, чем θ, в соответствии с целевым распределением, оно принимается. Если оно менее вероятно, оно принимается с определенной вероятностью или отвергается в противном случае.

Draw $\boldsymbol{\theta}_0 \sim Initial\ probability\ distribution$;
while $n = 0$ **to** N **do**
  Draw $\boldsymbol{\theta}' \sim Q(\boldsymbol{\theta}'|\boldsymbol{\theta}_n)$;
  $p = \min\left(1, \frac{Q(\boldsymbol{\theta}'|\boldsymbol{\theta}_n)}{Q(\boldsymbol{\theta}_n|\boldsymbol{\theta}')}\frac{f(\boldsymbol{\theta}')}{f(\boldsymbol{\theta}_n)}\right)$;
  Draw $k \sim \text{Bernoulli}(p)$;
  if k **then**
    $\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}'$;
    $n = n + 1$;
  end if
end while

Рис. 4.22. Алгоритм Метрополиса

Алгоритм достаточно плохо масштабируется. Разброс $Q(\theta'|\theta_n)$ необходимо настраивать. Если он слишком велик, процент отказов будет слишком высоким. Если он слишком мал, выборки будут более автокоррелированными. Не существует общего метода настройки этих параметров, однако разумные стратегии уже предложены (см., например, [Hoffman, 2014]).

Вариационный вывод не является точным методом. Вместо того, чтобы работать с выборкой из точного апостериорного распределения, можно использовать распределение $q_\varphi(H)$, называемое вариационным распределением, параметризованное набором параметров φ. Затем значения параметров φ обучаются так, чтобы вариационное распределение $q_\varphi(H)$ было как можно ближе к точному апостериорному $P(H|D)$. В качестве меры близости обычно используется дивергенция Кульбака-Лейблера (KL-дивергенция).

Вариационный вывод масштабируется лучше, чем метод Монте-Карло. Для его адаптации к глубоким нейронным сетям предложен алгоритм байесовского обратного распространения (Bayes-by-backprop) [Blundell, 2015].

$\boldsymbol{\phi} = \boldsymbol{\phi}_0$;
for $i = 0$ **to** N **do**
  Draw $\varepsilon \sim q(\varepsilon)$;
  $\boldsymbol{\theta} = t(\varepsilon, \boldsymbol{\phi})$;
  $f(\boldsymbol{\theta}, \boldsymbol{\phi}) = \log(q_{\boldsymbol{\phi}}(\boldsymbol{\theta})) - \log(p(D_{\boldsymbol{y}}|D_{\boldsymbol{x}}, \boldsymbol{\theta})p(\boldsymbol{\theta}))$;
  $\Delta_{\boldsymbol{\phi}} f = \text{backprop}_{\boldsymbol{\phi}}(f)$;
  $\boldsymbol{\phi} = \boldsymbol{\phi} - \alpha\Delta_{\boldsymbol{\phi}} f$;
end for

Рис. 4.23. Алгоритм Bayes-by-backprop

Идея алгоритма (рис. 4.23) состоит в том, чтобы использовать в качестве невариационного источника шума случайную величину $\varepsilon \sim q(\varepsilon)$. θ не выбирается напрямую, а получается с помощью детерминированного преобразования $t(\varepsilon, \varphi)$, так что $\theta = t(\varepsilon, \varphi)$ следует за $q_\varphi(H)$. ε выбирается и, следовательно, изменяется на каждой итерации, но все же может считаться константой относительно других переменных.

Простейшим методом формирования БНС является метод Bayes via Dropout. В этом случае выполняется обучение обычной ГНС с использованием регуляризации Dropout, но на последнем шаге к целевому слою применяется мультипликативный шум типа Бернулли. В результате мы получаем ансамбль ГНС, который работает как одна БНС.

Метод прост в реализации и не требует дополнительных знаний или усилий по моделированию, что часто обепечивает более быстрое обучение по сравнению с другими подходами. Если в обученной модели уже присутствуют слои с Dropout, то ее можно использовать в качестве BNN без необходимости повторного обучения. С другой стороны, Bayes via Dropout имеет ограниченную выразительность и может не полностью отражать неопределенность, присутствующую в обрабатываемых данных, особенно по сравнению с другими байесовскими методами онлайн или активного обучения.

Типы неопределенности и их оценка посредством БНС. Принципиальное преимущество БНС по сравнению со стандартными нейронными сетями заключается в том, что в ней предусмотрен ответ «Не знаю». Например, если вы пять раз подаете на вход БНС одни и те же данные и получаете пять очень разных результатов прогнозирования, вы можете рассматривать прогноз как результат «Я не уверен». Другими словами, БНС может дать представление о неопределенности обрабатываемого процесса. Это достигается путем сравнения предсказаний нескольких выборочных параметризаций модели θ. Если разные модели согласуются, то неопределенность низка. Если они не совпадают, то неопределенность высока. Больше того, применение БНС позволяет уточнить тип этой неопределенности – алеаторическая или эпистемическая [Kendall, 2017]:

- Алеаторическая неопределенность отражает шум, присущий наблюдениям (например, шум датчика или шум движения). Ее невозможно уменьшить, даже если собрать больше данных.
- Эпистемическая неопределенность связана с неопределенностью параметров модели, использованной при генерации обрабатываемого датасета, и ее часто называют модельной неопределенностью. Ее можно объяснить при наличии достаточного количества данных.

Разницу между ними иллюстрирует рис. 4.24 на примере семантической сегментации датасета CamVid [Brostow, 2009]. Система сегментации демонстрирует повышенную алеаторическую неопределенность в зонах с высоким шумом наблюдения – на границах объектов и для объектов, находящихся далеко от камеры (d), и повышенную эпистемическую неопределенность для семантически и визуально сложных зон изображения (e). В нижнем ряду показан случай отказа модели сегментации, когда модель не может сегментировать пешеходную дорожку из-за повышенной эпистемической, но не алеаторической неопределенности.

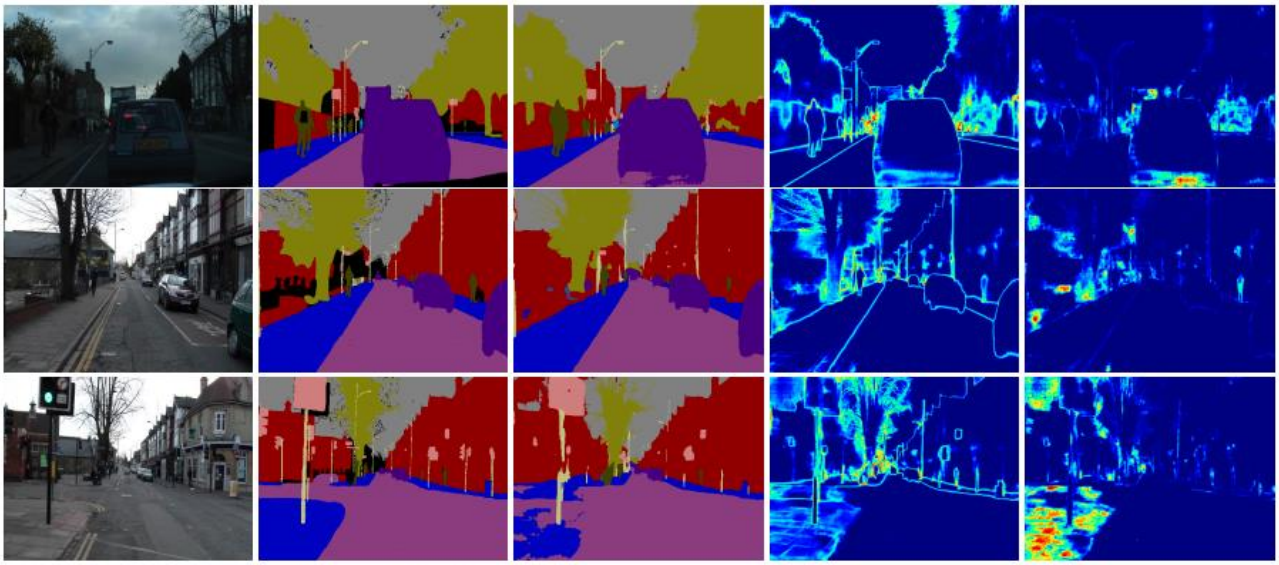


Рис. 4.24. Иллюстрация разницы между алеаторической и эпистемологической неопределенностью

С точки зрения построения систем ИИ алеаторическая неопределенность наиболее важна при обработке больших данных (здесь эпистемологическая неопределенность устраняется) и в приложениях реального времени (здесь можно формировать алеаторические модели без дорогостоящих выборок методом Монте-Карло). В то же время эпистемологическая неопределенность важна в приложениях, критических с точки зрения безопасности (здесь требуется понимание примеров, отличных от обучающих данных), а также при работе на небольших обучающих выборках.

Для оценки алеаторической неопределенности уже предложены подходы (см., например, [Kendall, 2017]), в то время как оценка эпистемической неопределенности остается зоной активного поиска.

Почему при построении систем ИИ важно иметь оценку неопределенности и ее типа? Ограничимся двумя примерами. Первый смертельный исход из-за автоматизированной системы вождения был вызван тем, что система приняла белую сторону трейлера за яркое небо. Во втором случае система классификации изображений ошибочно идентифицировала двух афроамериканцев как горилл, что вызвало скандал по поводу расовой дискриминации. Если бы алгоритмы, использованные в обоих случаях, могли присвоить своим ошибочным предсказаниям высокий уровень неопределенности, то СИИ смогли бы принимать более правильные решения и, вероятно, избежать катастрофы.

Преимущества и недостатки БНС обусловлены введением стохастичности в архитектуру НС. Начнем с преимуществ.

- Работу БНС можно рассматривать как моделирование нескольких возможных моделей θ с соответствующим распределением вероятностей $p(\theta)$, т.е. как частный случай ансамблевого обучения. Так как агрегирование прогнозов большого набора среднеэффективных, но независимых предикторов может привести к лучшим прогнозам, чем один высокоэффективный предиктор,

то БНС могут обеспечить лучшую эффективность по сравнению с обычными НС (с точечной оценкой).

- Встроенная изменчивость БНС делает их устойчивыми к переобучению.
- Модель сообщает об уровне уверенности в прогнозе, что исключительно важно в свете требований объяснимого ИИ.
- БНС позволяет различать эпистемическую неопределенность $p(\theta|D)$ и алеаторическую неопределенность $p(y|x, \theta)$. Это делает BNN очень эффективными с точки зрения данных, поскольку они могут учиться на небольшом наборе данных без переобучения [Depeweg, 2018]. Во время прогнозирования точки распределения, не покрытые в процессе обучения, будут иметь высокую эпистемическую неопределенность, и сеть сообщит об этом в форме «Я не знаю» (вместо того, чтобы слепо давать неверный прогноз).

Однако БНС значительно сложнее стандартных нейронных сетей, что затрудняет их реализацию и процесс обучения. Большая часть текущих исследований, связанных с БНС, направлена на поиск методов, облегчающих их обучение.

4.6.3. Графовые НС

Определение и структура ГНС. Графовые нейронные сети [Scarselli, 2008] предоставляют расширение существующих нейронные сети с помощью теории графов, рассматривая распространение информации на соседние узлы. В 2014 г. была предложена реализация свертки на графах, т.е. появились полноценные графовые сверточные сети (ГСС, GCN).

Граф может быть представлен как G = (V, E, W), где V – набор из N узлов, |V| = H; E – набор ребер, соединяющих эти узлы, а W – матрица смежности, задающая веса связи между узлами w_{ij} (рис. 4.25).

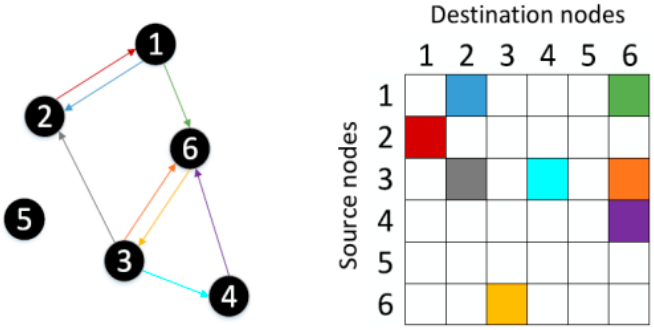


Рис. 4.25. Пример ориентированного графа (слева) и соответствующей матрицы смежности (справа)

Упрощенно процесс построения графовой НС можно представить следующим образом (рис. 4.26). Исходные данные преобразуются в графовую структуру, которая описывает узлы и связи между ними. Слой объединения графов в ГНС (pooling layer) комбинирует информацию о нескольких вершинах в одну вершину, чтобы уменьшить размер графа и расширить рецептивное поле фильтров сигнала графа. В результате граф преобразуется в сравнительно небольшую структуру, в которой каждая вершина несет в себе информацию обо всем дереве в исходном графе, корнем которого она является. Векторы признаков из

последнего сверточного слоя графа объединяются в один вектор признаков, который подается в полносвязный слой для получения результатов классификации.

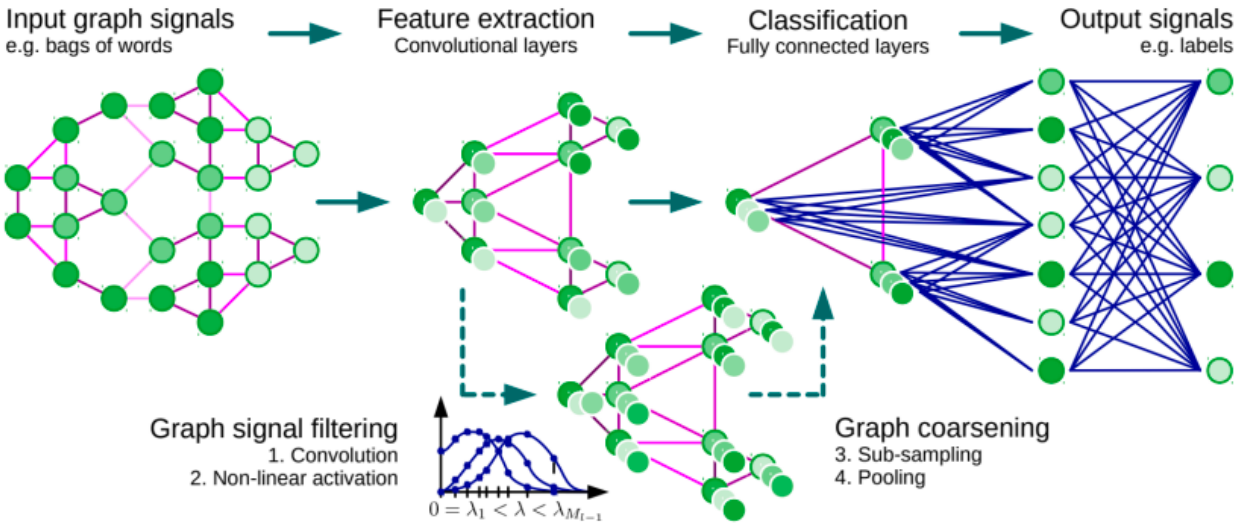


Рис. 4.26. Упрощенная схема формирования ГНС

Таким образом, исходный граф обрабатывается набором модулей, которые связаны между собой в соответствии со связями графа. В процессе обучения сети модули обновляют свои состояния и обмениваются информацией. Это продолжается до тех пор, пока модули не достигнут устойчивого равновесия (для того, чтобы была гарантия того, что такое устойчивое состояние существует, этот механизм распространения ограничен). Выходные данные ГНС вычисляются на основе состояния модуля на каждом узле.

Построение ГНС существенно отличается от построения обычных НС.

Во-первых, необходимо преобразовать необработанные данные в графовое представление (рис. 4.27) и определить тип получающегося графа (направленный/ненаправленный, гомогенный/гетерогенный, статический/динамический).

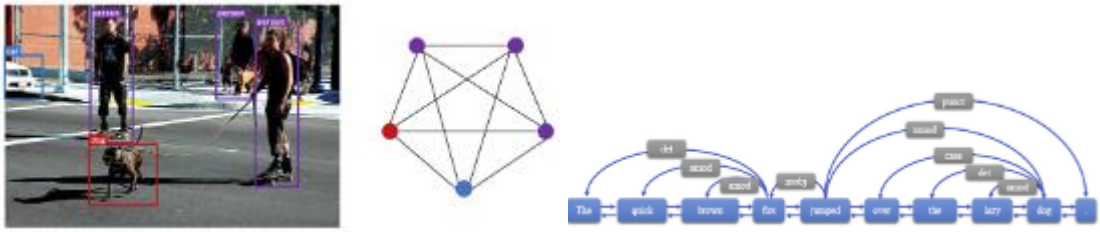


Рис. 4.27. Формирование графа из различных модальностей

Во-вторых, нужно определить не только тип задачи (классификация/регрессия/кластеризация), но и уровень обрабатываемых единиц (вершины/ребра/граф целиком). Задачи на вершинах включают классификацию, регрессию, кластеризацию и т.д.: классификация вершин пытается разделить их на несколько классов, регрессия предсказывает непрерывное значение для каждой вершины, кластеризация вершин направлена на разделение их на несколько непересекающихся групп похожих в каком-то смысле вершин. К задачам на ребрах относится классификация типов ребер и прогнозирование существования ребра между двумя заданными

узлами. Задачи на уровне графа (подграфа) включают классификацию графов, регрессию графов и сопоставление (matching) графов.

В-третьих, для построения ГНС используются специальные вычислительные модули. Модуль распространения (propagation module) отвечает за то, чтобы между вершинами распространялась информация как о фичах, так и о топологии сети. Для агрегирования информации от соседних узлов могут использоваться оператор свертки или рекуррентный оператор. В случае больших графов дополнительно может использоваться сэмплирование. Для повышения глубины ГНС без излишнего сглаживания информации о промежуточных узлах применяется соединение с пропуском (skip connection). Когда требуется переходить к представлениям подграфов более высокого уровня, используется пулинг.

Базовый пайплайн для построения ГНС [Zhou, 2020] изображен на рис. 4.28.

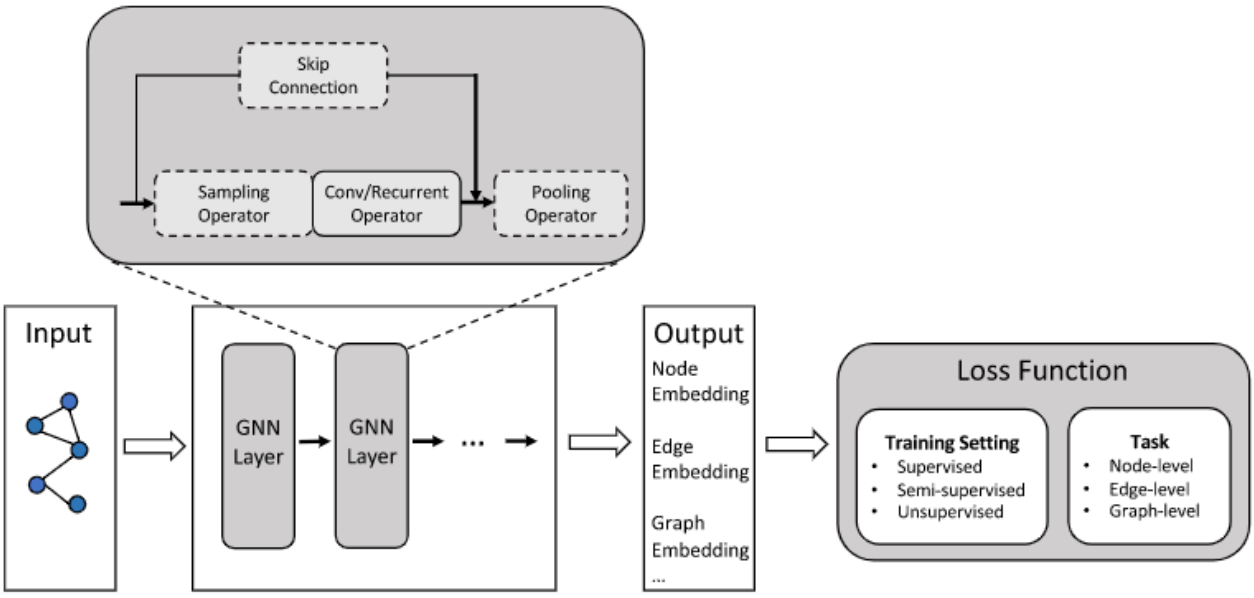


Рис. 4.28. Общий пайплайн проектирования для модели ГНС

Операция свертки в ГНС. Операция свертки является наиболее специфичной в архитектуре ГНС. Она может выполняться в спектральной или в пространственной области, а также на основе механизма внимания.

В *спектральных методах* входной вектор графа **x** сначала преобразуется в спектральную область с помощью графового преобразования Фурье $\mathscr{F}$, затем проводится операция свертки, после которой полученный сигнал преобразуется обратно с помощью обратного графового преобразования Фурье $\mathscr{F}^{1}$:

$\mathscr{F}(\mathbf{x})=\mathbf{U}^{T}(\mathbf{x})$,

$\mathscr{F}^{1}(\mathbf{x})=\mathbf{U}(\mathbf{x})$,

где **U** – матрица собственных векторов нормализованного лапласиана **L** исходного графа,

$\mathbf{L} = \mathbf{I}_N + \mathbf{D}^{1/2}\mathbf{A}\mathbf{D}^{1/2}$,

(**D** – матрица степеней графа, **A** – матрица смежности графа). Операция свертки в исходном виде вычислительно сложна, поэтому предложены различные упрощения, каждое из которых приводит к появлению нового типа ГНС.

Например, в архитектуре ChebNet [Defferrard, 2016] спектральное преобразование аппроксимируется полиномами Чебышева младших порядков.

Популярная архитектура GCN [Kipf, 2017] вводит еще более сильное упрощение операции свертки, резко уменьшая число свободных параметров и тем самым устраняя проблему переобучения сети (рис. 4.29). Кроме того, для борьбы с исчезающими градиентами здесь использована ренормализация (renormalization trick) лапласиана L. Работа сети GCN описывается компактным выражением

$$\mathbf{H} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}\mathbf{X}\mathbf{W},$$

где $\mathbf{X} \in \mathbb{R}^{N\times F}$ – входная матрица, $\mathbf{W} \in \mathbb{R}^{F\times F'}$ – матрица параметров, $\mathbf{H} \in \mathbb{R}^{N\times F'}$ – свернутая матрица, FF' – входная и выходная размерность соответственно, волной обозначены ренормализованные матрицы. Содержательно сеть GCN представляет собой стек нескольких сверточных слоев, за каждым из которых следует поточечная нелинейность.

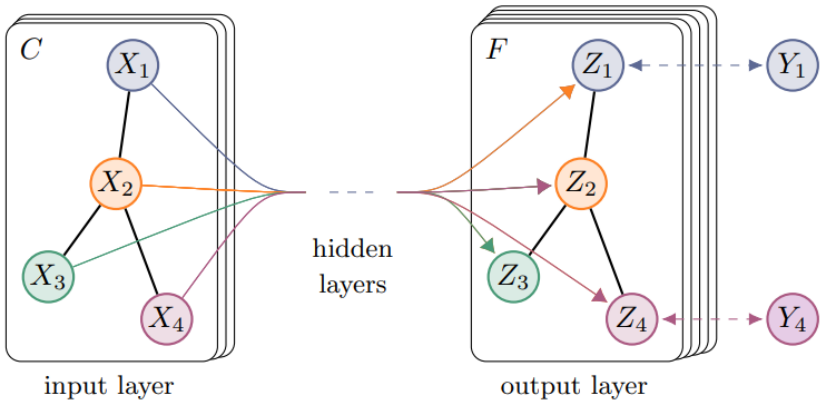


Рис. 4.29. (а) Схема многослойной сверточной ГНС. Структура графа (ребра показаны черными линиями) является общей для слоев, метки обозначаются Y_i

[Xu, 2019] вместо Фурье-преобразования графа используется его вейвлет-разложение, что приносит существенные преимущества: вейвлеты рассчитываются без вычислительно затратной декомпозиции матрицы, кроме того, они локализуются вблизи зон интереса, что важно с позиции объяснимого ИИ.

Пространственные методы определяют свертку непосредственно на графе на основе его топологии. Основной проблемой здесь является определение операции свертки с окрестностями разного размера и поддержание локальной инвариантности CNN.

В архитектуре LGCN [Gao, 2018] выполняется max-pooling матриц смежности каждой вершины, чтобы получить набор верхнеуровенвых признаков, а затем используется одномерная CNN для вычисления скрытых представлений.

Архитектура GraphSAGE [Hamilton, 2017] вместо обработки полного набора матриц смежности каждой вершины генерирует их эмбеддинги путем выборки и агрегирования признаков из локальной окрестности вершины (рис. 4.30).

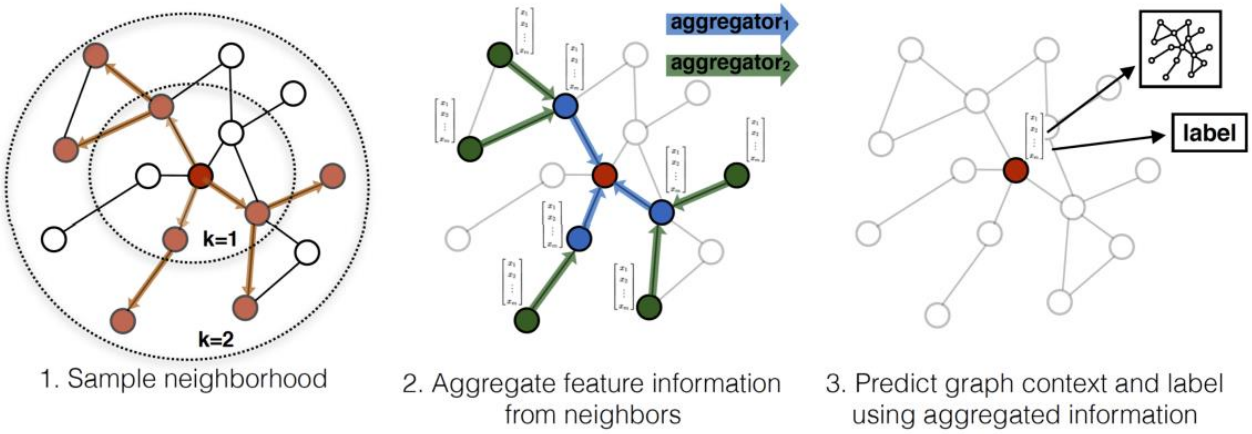


Рис. 4.30. Выборка и агрегирование признаков в GraphSAGE

Методы на основе внимания присваивают соседям обрабатываемой вершины разные веса. Например, архитектура графовой сети внимания (GAT) [Velickovic, 2018] вычисляет скрытое состояние каждой вершины, применяя механизм многоголовочного внимания к ее соседям. Для этого используется слой внимания, который вычисляет коэффициенты внимания

$$e_{ij} = a\left(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j\right),$$

отражающие важность вектора фич $\vec{h}_i$ узла *i* относительно вектора фич $\vec{h}_j$ узла *j* (рис. 4.31, слева). Затем агрегированные фичи, полученные каждой головкой, конкатенируются и усредняются (рис. 4.31, справа).

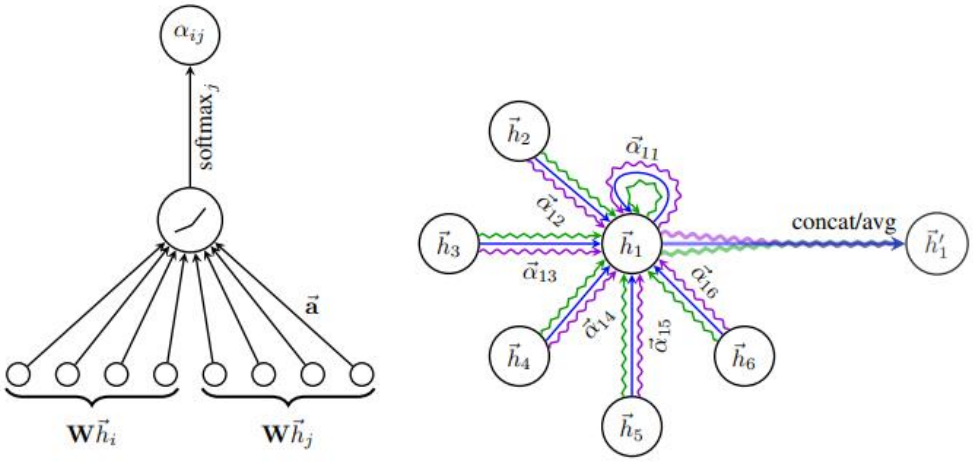


Рис. 4.31. Формирование выходных признаков с помощью трех векторов внимания (обозначены разными цветами)

К настоящему времени предложено очень большое число программных и архитектурных решений для ГНС, с которыми можно познакомиться в литературе. Подробный список решений с открытым исходном кодом приводится в [Zhou, 2020].

Сферы применения и проблемы ГНС. Области применения ГНС сегодня включают обработку как структурированных, так и неструктурированных предметных областей.

К первому варианту относятся моделирование систем реального мира в физике (движение человека и роботов), химии и биологии (выделение характерных

участков молекул и прогнозирование взаимодействия белков, необходимые при поиске лекарственных средств), построение и обработка графов знаний (построение эмбеддингов с богатой семантикой, предсказание недостающих связей между сущностями, интершаговые рассуждения – multihop reasoning, рекомендательные системы), различные задачи комбинаторной оптимизации. ГНС успешно используются в традиционных задачах обработки текста, таких как классификация, извлечение именованных сущностей, связей и событий, верификация фактов, а также машинный перевод.

При обработке изображений ГНС используются в актуальной проблеме классификации по сверхмалому набору данных (zero-shot and few-shot learning), а также для совместной обработки визуальной и семантической информации (в задачах семантической сегментации, в визуальных вопросно-ответных системах). Например, авторы [Teney, 2017] параллельно строят граф сцены изображения и синтаксический граф вопроса, а затем с помощью ГНС тренируют эмбеддинги для формирования ответа на вопрос.

Тем не менее, многие проблемы в области построения и использования ГНС остаются открытыми:

- концептуальную сложность может представлять преобразование необработанных данных в графовое представление: например, как при обработке изображений выделить зоны интереса, которые правомерно считать узлами;
- трудно обрабатывать концептуально сложные графы (динамические, гетерогенные и др.);
- графовые сети в целом обучаются хуже, чем обычные; для обучения требуются очень большие наборы данных и большие вычислительные мощности;
- типовые методы построения объяснимых НС и XAI фокусируются только на интенсивности элементов обрабатываемого множества (например, на яркости пикселей), т.е. не могут объяснить, как сеть обрабатывает ребра графа. Обзор графо-ориентированных методов XAI представлен в [Ahmedt-Aristizabal, 2021];
- как и все другие НС, ГНС подвержены действию состязательных атак, причем, в отличие от изображений, атакам подвергаются не только собственно признаки, но и структура связей между ними, соответственно, требуются более изощренные методы защиты [Zhu, 2019].

4.6.4. Генеративные модели в ИИ

Дискриминативные vs генеративные модели. Большинство задач машинного обучения идеологически делятся на две большие группы:

– классификация имеющихся экземпляров (например, изображений) – задача дискриминативного моделирования;
– создание новых экземпляров, аналогичных уже имеющимся (например, создание реалистичных изображений) – задача генеративного моделирования.

Дискриминативная модель имеет на входе размеченный датасет, т.е. набор объектов X из какого-то распределения и метки их классов Y, и на выходе должна

определить вероятность принадлежности нового объекта (из того же распределения) $x \in X$ к классу $y \in Y$. Другими словами, дискриминативная модель учится моделировать границы принятия решений среди классов (например, кошек, собак и тигров). Граница решения может быть линейной или нелинейной. Точки данных, которые находятся далеко от границы принятия решения (так называемые выбросы), не очень важны – учитываются только те, кто находится ближе всего к этой границе. Принципиально важно, что дискриминативные модели классифицируют точки данных, не принимая во внимание то, как эти точки были созданы.

Примеры дискриминативных моделей – машина опорных векторов (SVM), логистическая регрессия, метод k ближайших соседей (kNN), случайный лес (Random Forest), глубокие нейронные сети.

Генеративная модель пытается воспроизвести способ создания набора данных, т.е. понять распределение точек данных, работая в терминах вероятностной модели. Цель состоит в том, чтобы сгенерировать новые образцы из того, что уже было распределено в обучающих данных.

Рис. 4.32. Примеры работы генеративных моделей

Предположим, имеется набор данных об автономном вождении с настройками городской сцены (рис. 4.32). Задача генеративного моделирования в этом случае – сгенерировать из исходного набора данных изображения, которые семантически и пространственно похожи на то, что уже есть в нем. Для этого генеративная модель должна понимать основную структуру данных и изучить обобщенное представление набора данных (через базовые фичи), например, небо голубое, здания обычно высокие, а пешеходы ходят по тротуарам. Таким образом, задача генеративного моделирования – создавать реалистичные образцы X с распределением вероятностей, аналогичным P_{data} (исходным данным из обучающей выборки). Шум добавляет модели случайности и обеспечивает разнообразие сгенерированных изображений. Генеративное моделирование учится приближать вероятность $p(x)$, т.е. вероятность наблюдения x в обучающем наборе данных.

Некоторые примеры генеративных моделей – наивные байесовские модели, скрытые марковские модели, автоэнкодеры, машины Больцмана, вариационные автоэнкодереы, генеративные состязательные сети, трансформеры.

Сопоставим оба типа моделей:

- Для классификации можно использовать оба типа моделей.
- Генеративные модели могут обучаться как с учителем, так и без учителя; дискриминативные модели используются только для задач обучения с учителем.
- Цель дискриминативной модели – оценить условную вероятность $P(Y|X)$. Генеративная модель при обучении с учителем учится приближать $P(X)$ и $P(X|Y)$, а при обучении без учителя самостоятельно определяет $P(Y|X)$.
- Дискриминантные модели изучают линейную либо нелинейную границу решения, используя точки данных и их метки, без знания того, как были созданы данные. Генеративные модели, обучаясь моделировать распределение вероятностей данных, в определенном смысле «понимают» основные характеристики данных.
- В условиях обучения с учителем дискриминативные модели работают лучше генеративных, особенно когда выбранная генеративная модель плохо соответствует имеющимся данным. В то же время генеративные модели применимы при отсутствии разметки данных, где дискриминативные модели не работают.

Автоэнкодеры. Автокодировщик (автоэнкодер) – нейронная сеть, которая используется для неконтролируемого обучения сжатию данных. Цель автоэнкодера – изучить обобщенное скрытое представление (кодировку) набора данных.

Автоэнкодер [Hinton, 2006] (рис. 4.33) состоит из двух блоков. Кодер сжимает входные данные большой размерности, переводя их в низкоразмерное представление скрытого пространства. Декодер. наоборот, распаковывает низкоразмерное представление в исходный многомерный вход.

Рис. 4.33. Схема работы автоэнкодера

Получаемое скрытое представление очень полезно: на этом уровне можно производить различные трансформации исходного изображения, оставляя только нужную информацию; его легче передавать по сетям; здесь наиболее эффективно выполнять шумоподавление (рис. 4.34); и т.д.

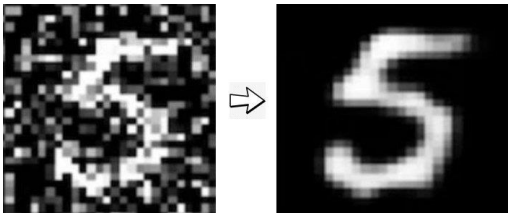


Рис. 4.34. Шумоподавление в автоэнкодере

По существу, автоэнкодер решает ту же задачу, что и другие методы снижения размерности признакового пространства, например анализ главных компонент (PCA). Чем же отличаются автоэнкодеры от PCA (рис. 4.34)?

PCA представляет собой линейное преобразование. Компоненты вектора признаков, получаемого в PCA, являются проекциями на ортогональный базис и линейно не коррелированы друг с другом. В простейших случаях автоэнкодер и PCA эквивалентны: автоэнкодер только с одной функцией активации ведет себя как PCA, и для линейного распределения оба работают одинаково. PCA быстрее и дешевле в вычислении, чем автоэнкодеры. Из-за большого количества параметров автоэнкодер склонен к переобучению. Однако главное преимущество автокодировщиков состоит в том, что они способны моделировать сложные нелинейные функции.

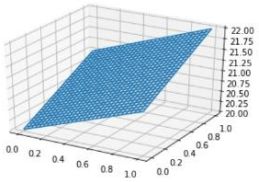
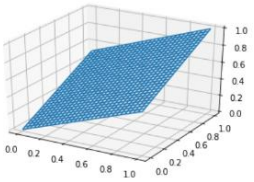
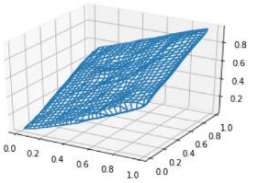
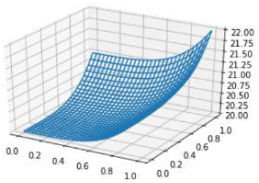
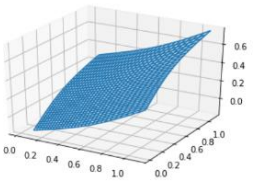
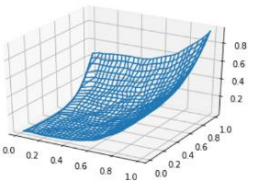
Function	Feature Space	PCA Reconstruction	Auto Encoder Reconstruction
Plane			
Curved Surface			

Рис. 4.35. Сравнение PCA и автоэнкодера

Важно отметить, что автоэнкодеры не являются чисто генеративной моделью. Дело в том, что скрытое пространство в автоэнкодерах (рис. 4.35) не является плотным, т.е. для каждого изображения имеется вектор в этом пространстве, но не каждый вектор представляет разумное изображение. Разумеется, декодер автокодировщика создаст изображение из любого вектора, но большинство таких изображений не будет представлять никаких узнаваемых образов.

Чтобы сделать автоэнкодеры (рис. 4.36, а) генеративными, нужно перейти к вариационным автоэнкодерам (VAE) (рис. 4.36, б) [Kingma, 2014]. Они имеют непрерывное скрытое пространство, т.е. кодируют входное изображение в два отдельных вектора, среднее значение μ и логарифм дисперсии σ, посредством задания функции потерь в виде суммы:

$$L(\phi,\theta,x)=\frac{1}{N}\sum_{i=1}^{N}\left(X_i-\hat{X}_i\right)^2+KL\left[\left(Z_\mu,Z_\sigma\right),\mathcal{N}(0,1)\right].$$

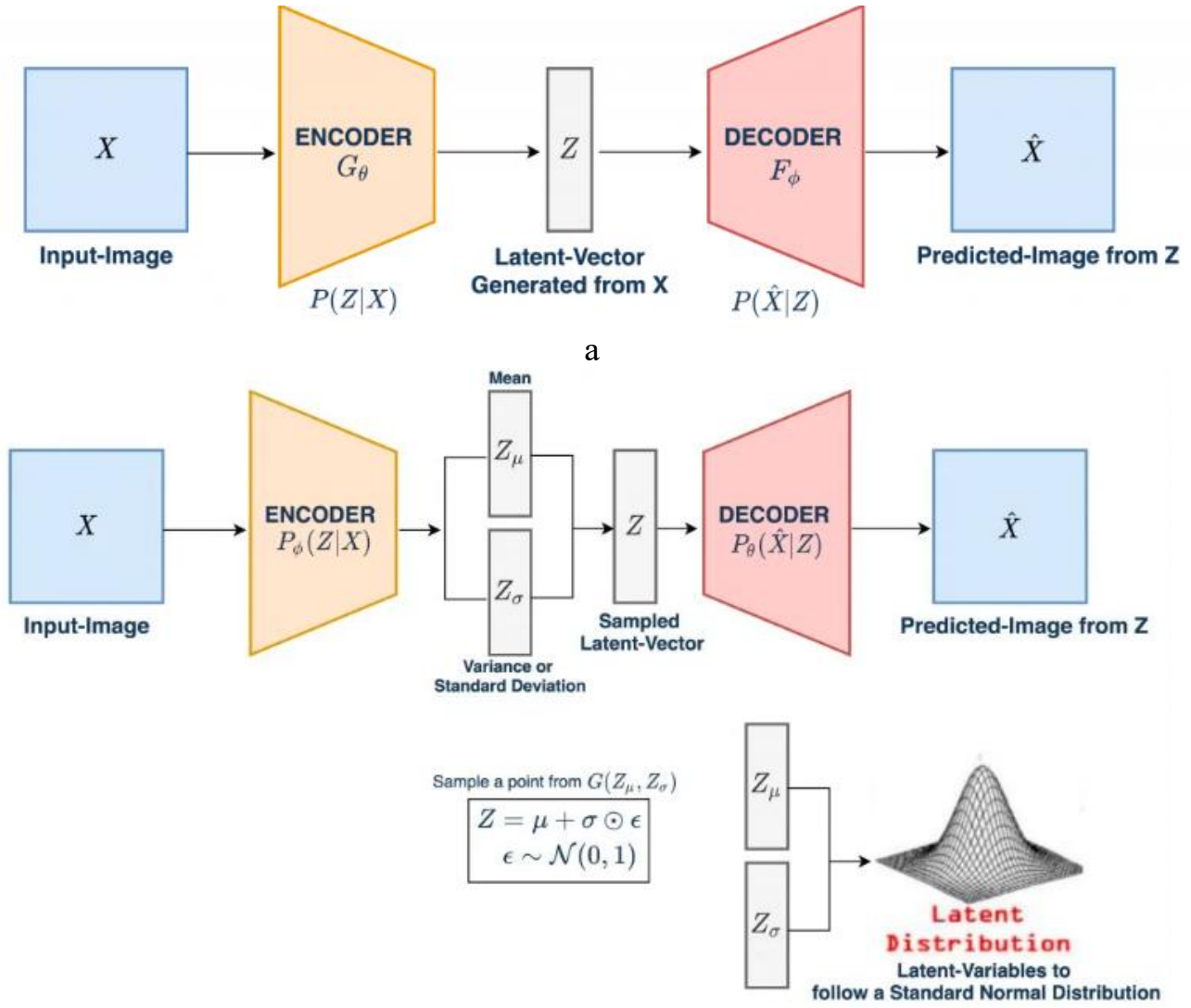


Рис. 4.36. Сравнение автоэнкодера (а) и вариационного автоэнкодера (б)

Первое слагаемое – потери реконструкции – это функция потерь обычного автокодировщика. Второе слагаемое – дивергенция Кульбака-Лейблера (KL) – действует как регуляризатор, сохраняющий достаточное разнообразие кодировок Z. Этот член описывает разницу в статистическом распределении изображений и абстрактных представлений. Другими словами, KL-дивергенция оптимизирует параметры распределения вероятностей μ и σ, чтобы они очень напоминали единичное гауссово распределение $\mathcal{N}(0,1)$.

Другими словами, VAE обучается так, чтобы вектор скрытых переменных соответствовал многомерному нормальному распределению. Если теперь взять случайную выборку из распределения и затем передать ее декодеру, он восстановит изображение (рис. 4.37).

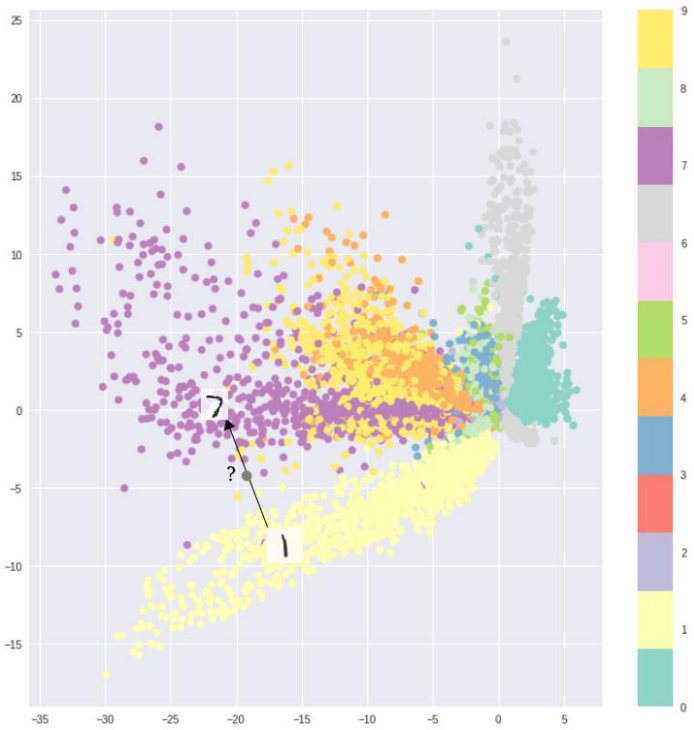


Рис. 4.37. Латентное пространство автоэнкодера для датасета цифр MNIST

Рис. 4.38. Примеры работы автокодировщика: верхний ряд – исходные изображения кошек; нижний ряд – изображения, восстановленные автокодировщиком [Осинга, 2019]

Принципиальное ограничение VAE состоит в том, что его скрытое пространственное представление должно быть близко к нормальному распределению, т.е. он обучается с целью уменьшения среднеквадратичной ошибки, которая принципиально обусловливает размытые (blurred) границы. В результате VAE не способен генерировать четкие изображения. Этот недостаток преодолевается в GAN.

Генеративные состязательные сети (generative afversarial networks, GAN**).** ГАНы занимают центральное положение среди генеративных моделей: они характеризуются умеренной сложностью архитектуры и настройки и в то же время обладают достаточно большой выразительной силой. Впервые идея GAN была опубликована в [Goodfellow, 2014], и сейчас GAN'ы являются одними из лучших генеративных моделей. Как и у любой другой генеративной модели, задача GAN – научиться генерировать сэмплы из распределения, максимально близкого к распределению данных (обычно имеется датасет ограниченного размера, распределение данных в котором мы хотим промоделировать).

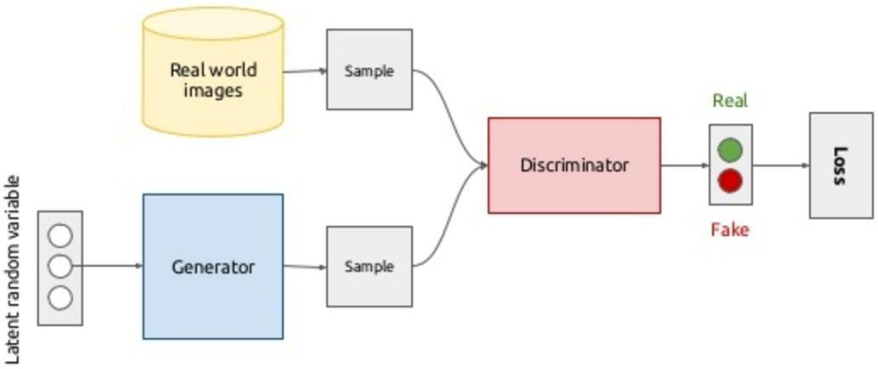


Рис. 4.39. Схема работы ГАН: генератор – нейронная сеть, которая получает на вход так называемые скрытые переменные (latent variable) в виде случайного шума, а выдает свой вариант сгенерированных данных (sample); дискриминатор –бинарный классификатор, который выдаёт 1 для реальных (real) и 0 – для поддельных (fake) данных

Процедура обучения GAN (на примере генерации изображений) содержательно заключается в следующем:

- получаем порцию реальных картинок;
- генерируем шум, на базе которого генератор генерирует картинки;
- формируем батч для обучения дискриминатора, который состоит из реальных картинок (метка 1) и «подделок от генератора» (метка 0);
- обучаем дискриминатор;
- обучаем GAN (в нём обучается генератор, так как обучение дискриминатора отключено), подавая на вход шум и ожидая на выходе метку 1.

Обратите внимание, что для обучения генератора не используются реальные изображения, а только метка дискриминатора, т.е. генератор обучается на градиентах ошибки от дискриминатора.

Главное отличие GAN от вариативного автоэнкодера состоит в функции потерь – в GAN используется бинарная кросс-энтропийная функция потерь:

$$L(\hat{y},y)=-\frac{1}{N}\left[y_i\log\hat{y}_i+(1-y_i)\log(1-\hat{y}_i)\right],$$

где $\hat{y}_i$ – предсказание дискриминатора, а y_i – истинная метка, независимо от того, является ли образец реальным (1) или поддельным (0). Вот как эта функция потерь реализуется в ходе обучения GAN.

Цель генератора – выучить распределение p_g по исходным данным x.

Начальная позиция определяется на входных шумовых переменных $p_z(z)$, которые выбираются из нормального распределения. Затем вектор входного шума отображается в пространство данных как $G(z;\theta_g)$, где G – дифференцируемая функция, представленная набором полносвязных сетей с обучаемыми параметрами θ_g. Вторая полносвязная сеть $D(x;\theta_d)$ выводит единственное скалярное значение [0, 1]. $D(x)$ представляет собой вероятность того, что x получен из истинного распределения данных, а не из p_g или генератора G.

Таким образом, мы обучаем сеть максимизировать вероятность того, что D присвоит правильную метку как обучающим примерам, так и образцам, полученным из G. В то же время мы обучаем G минимизировать $-\log(1-D(G(z)))$.

Другими словами, D и G играют в минимаксную игру для двух игроков с функцией стоимости $V(G, D)$:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}\left[\log D(x)\right] + E_{z\sim p_z(z)}\left[\log\left(1 - D(G(z))\right)\right].$$

Если посмотреть на кривые функции потерь (рис. 4.40), то видно, что дискриминатор быстро обучается отличать реальную картинку от первоначального «мусора», выдаваемого генератором, но потом кривые начинают колебаться — генератор учится генерировать всё более подходящее изображение.

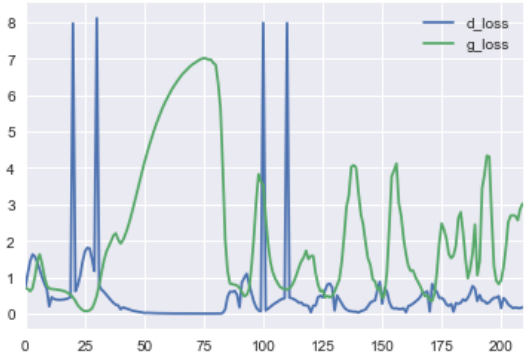


Рис. 4.40. Динамика функции потерь при обучении ГАН

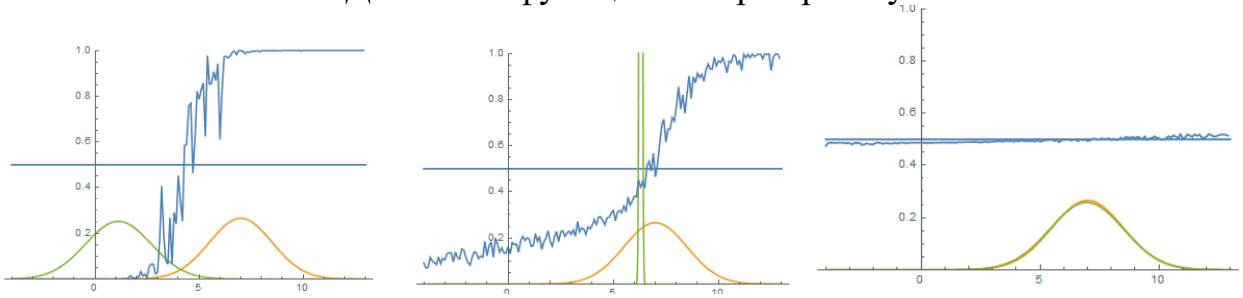


Рис. 4.41. Визуализация процесса обучения модели [Strakhov, 2018]. Неподвижная (желтая) гауссиана – плотность распределения реальных данных, движущаяся (зеленая) гауссиана – плотность распределения генерируемых примеров, синяя кривая – результат работы дискриминатора, т.е. вероятность примера быть настоящим

Игровой характер работы GAN предопределяет сложность его настройки. В случае обычной задачи классификации для обновления параметров нейронных сетей используется только вектор градиентов, т.е поле является потенциальным. А потенциальные векторные поля обладают некоторыми замечательными свойствами, одним из которых является отсутствие замкнутых кривых, т.е. в этом поле невозможно «ходить кругами». Напротив, при обучении GAN, несмотря на то что векторные поля для генератора и дискриминатора по отдельности являются потенциальными, суммарное векторное поле не будет потенциальным. А это означает, что в этом поле могут быть замкнутые кривые, а это плохо.

Тем не менее, показано, что все классические GAN (за исключением Wassertein GAN, которая имеет свои способы улучшения стабильности) обладают «хорошими» полями, т.е. следование этим полям имеет единственную точку покоя, в которой распределение генератора равно распределению данных. Задача – как выйти на эту точку.

Если переписать функционал обучения GAN в общем виде:

$$J = \int p_d(x) f_1\big(D(x)\big)dx + \int p_g(x) f_2\big(D(x)\big)dx;$$

$$I = \int p_g(x) f_3\big(D(x)\big)dx,$$

причем необходимо оптимизировать оба функционала одновременно, то ход обучения определяется эволюцией двух параметров – p_g и D (рис. 4.42).

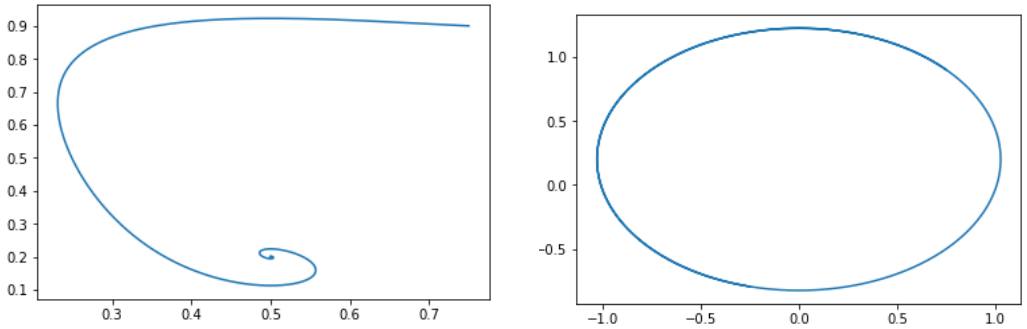


Рис. 4.42. Траектории обучения для ГАН из классической работы Goodfellow (а) и для Wasserstein GAN (б)

К сожалению, на данный момент нет ни единого способа теоретически проверить, как отдельные элементы нейросети меняют поле. Поле также очень чувствительно к параметризации нейронных сетей (выбору функций активации, использованию DropOut, BatchNormalization и т.д.), поэтому настройка GAN во многом напоминает «танцы с бубнами».

Тем не менее, результаты работы GAN впечатляют: они позволяют создать изображения, не отличимые от натуральных, в самых разных предметных областях, и производить с ними разнообразные трансформации (рис. 4.43).

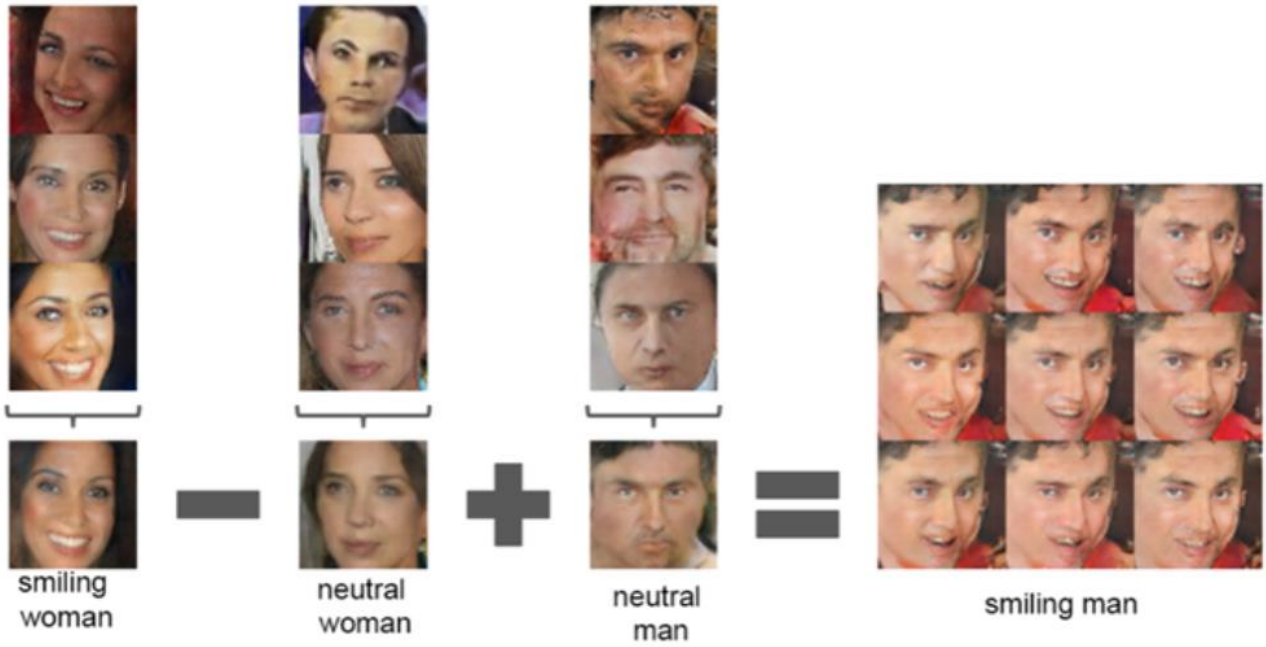


Рис. 4.43. Пример работы ГАН

Трансформеры были предложены в работе [Vaswani, 2017]. Архитектура трансформера (рис. 4.44) сохраняет структуру энкодер-декодер, но для генерации выходных данных в ней не используются ни рекуррентный, ни сверточный подход. Задача энкодера (левая половина) состоит в том, чтобы отобразить

входную последовательность в последовательность непрерывных представлений, которые затем подаются в декодер. Декодер (правая половина) получает выходные данные кодера вместе с выходными данными декодера на предыдущем временном шаге для создания выходной последовательности. На каждом шаге модель является авторегрессионной, используя ранее сгенерированные символы в качестве дополнительных входных данных при генерации следующего символа.

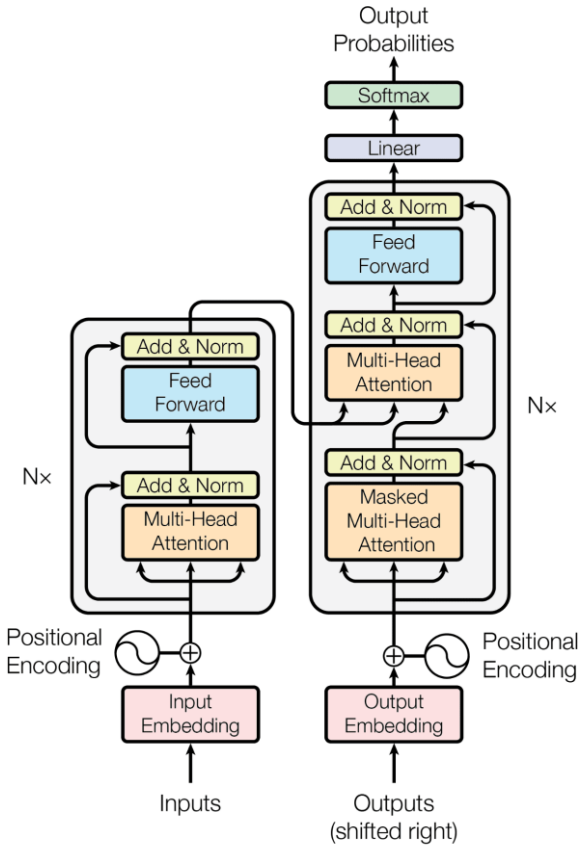


Рис. 4.44. Энкодер-декодерная архитектура трансформера

Энкодер состоит из стека из $N = 6$ одинаковых слоев. Каждый слой имеет два подслоя. Первый представляет собой механизм самовнимания (self-attention) с несколькими головками, а второй – позиционную полносвязную сеть прямого распространения.

Декодер также состоит из стека из $N = 6$ одинаковых слоев. В дополнение к двум подуровням в каждом уровне энкодера декодер вставляет третий подуровень, который обеспечивает многоголовочное внимание (multi-head attention) к выходным данным стека энкодера. Подуровень самовнимания в стеке декодера настроен таким образом, чтобы текущие позиции не обращались к последующим позициям; кроме того, выходные эмбеддинги смещены на одну позицию. В результате прогнозы для позиции i могут зависеть только от известных выходных данных в позициях меньше i.

В исходной статье [Vaswani, 2017] модель трансформера была ориентирована на машинный перевод. В этом случае она работает следующим образом:

1. Каждое слово, образующее входную последовательность, преобразуется в эмбеддинг размерности d_{model}.

2. Эмбеддинг каждого входного слова дополняется путем его суммирования (поэлементно) с вектором позиционного кодирования той же длины d_{model}, тем самым вводя позиционную информацию во входные данные.

3. Увеличенные эмбеддинги подаются в блок кодера, состоящий из двух подуровней, описанных выше. Поскольку кодировщик обрабатывает все слова во входной последовательности, независимо от того, предшествуют они или следуют за рассматриваемым словом, кодировщик трансформера является двунаправленным.

4. Декодер получает на вход свое предсказанное выходное слово на временном шаге t–1.

5. Ввод в декодер также дополняется позиционным кодированием таким же образом, как это делается на стороне кодера.

6. Входной сигнал расширенного декодера подается на три подуровня, составляющих описанный выше блок декодера. Маскирование применяется на первом подуровне, чтобы декодер не обращал внимание на последующие слова. На втором подуровне декодер также получает выходные данные кодера, что позволяет декодеру обрабатывать все слова во входной последовательности.

7. Наконец, выходные данные декодера проходят через полносвязный слой, за которым следует слой softmax, чтобы сгенерировать предсказание для следующего слова выходной последовательности.

На основе архитектуры трансформера была создана модель BERT (Bidirectional Encoder Representations from Transformers) [Devlin, 2018], которая продемонстрировала уровень перевода текстов, сопоставимый с людьми-носителями языка. BERT состоит из набора блоков трансформера типа [Vaswani, 2017], который был предварительно обучен на большом корпусе текстов общего характера, состоящем из 800 миллионов слов из англоязычных книг (данные взяты из BooksCorpus) и из 2,5 миллиардов слов из текстов статей английской Википедии (без разметки).

Модель BERT использует систему токенизации WordPiece (рис. 4.45), которая формирует токены разной длины – от целых слов до отдельных символов. Например, она разбивает слова вроде walking на токены walk и ##ing. Это позволяет модели делать некоторые заключения, основанные на структуре слов: два глагола, в конце которых имеется -ing, имеют схожие грамматические функции, а два глагола, начинающиеся с walk-, имеют схожие семантические функции.

К настоящему времени в открытом доступе имеются предобученные системы BERT для огромного количества языков, в том числе для русского. Самая крупная BERT-система использует 24 блока трансформера, 1024 измерений эмбеддинга и 16 блоков механизма внутреннего внимания. В результате эта модель имеет 340 миллионов параметров. Однако постоянно предлагаются и более компактные модели, например [Дале, 2021].

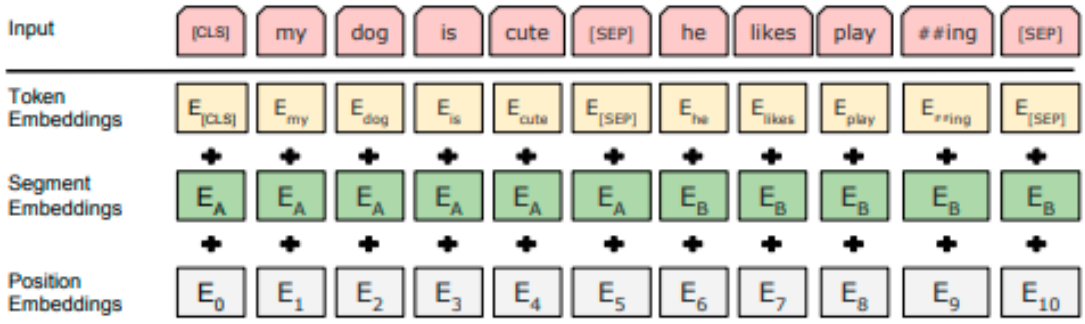


Рис. 4.45. Репрезентация входной последовательности в BERT. Входные эмбеддинги представляют собой сумму эмбеддингов токенов, эмбеддингов сегментации и эмбеддингов позиций

Архитектура трансформера оказалась очень успешной и получила распространение для обработки других модальностей, в том числе визуальной. В качестве примера ниже рассматривается применение трансформеров для обработки медицинских изображений [Shamshad, 2022].

В работе [Dosovitskiy, 2020] предложен визуальныый трансформер (Visual Transformer, ViT), в котором изображение интерпретируется как последовательность патчей и обрабатывается стандартным энкодером трансформера, как при обработке языковых последовательностей (рис. 4.46). Алгоритм работы ViT поясняется следующим псевдокодом:

```
1: Split a medical image into patches of fixed sizes
2: Vectorize image patches via flattening operation
3: Create lower-dimensional linear embedding from vectorized patches
via trainable linear layer
4: Add positional encoding to lower dimensional linear embeddings
5: Feed the sequence to ViT encoder as shown in Figure 4.43
6: Pre-train the ViT model on a large-scale image dataset
7: Fine-tune on the down stream medical image classification task
```

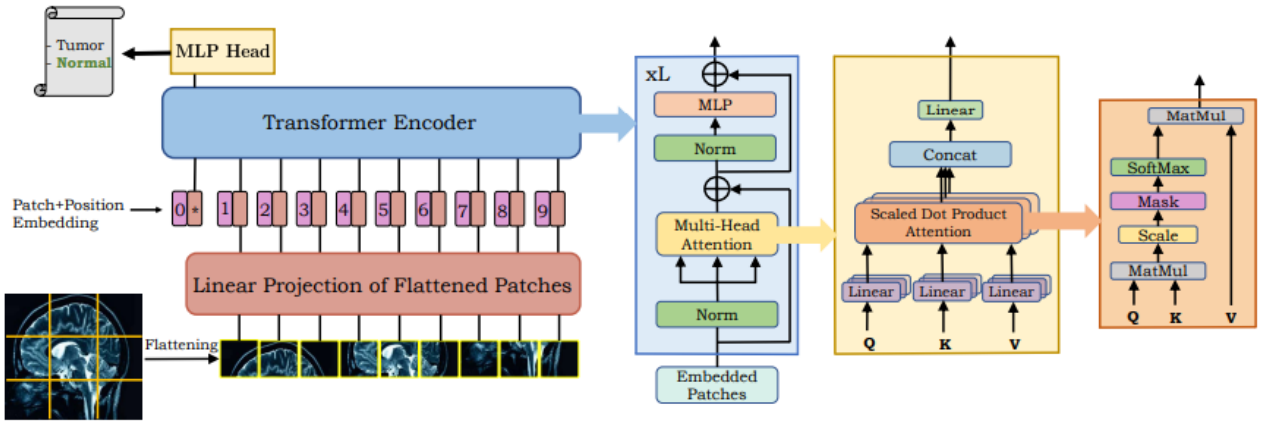


Рис. 4.46. Архитектура Vision Transformer (слева) и детали блока энкодера Vision Transformer (справа)

Эффективность трансформеров по сравнению с методами, основанными на глубоких нейронных сетях, связана в первую очередь с их способностью моделировать глобальный контекст. Например, при сегментации поражений кожи

(рис. 4.47) подходы на основе трансформеров лучше воспроизводят мелкие детали благодаря использованию априорных знаний о границах.

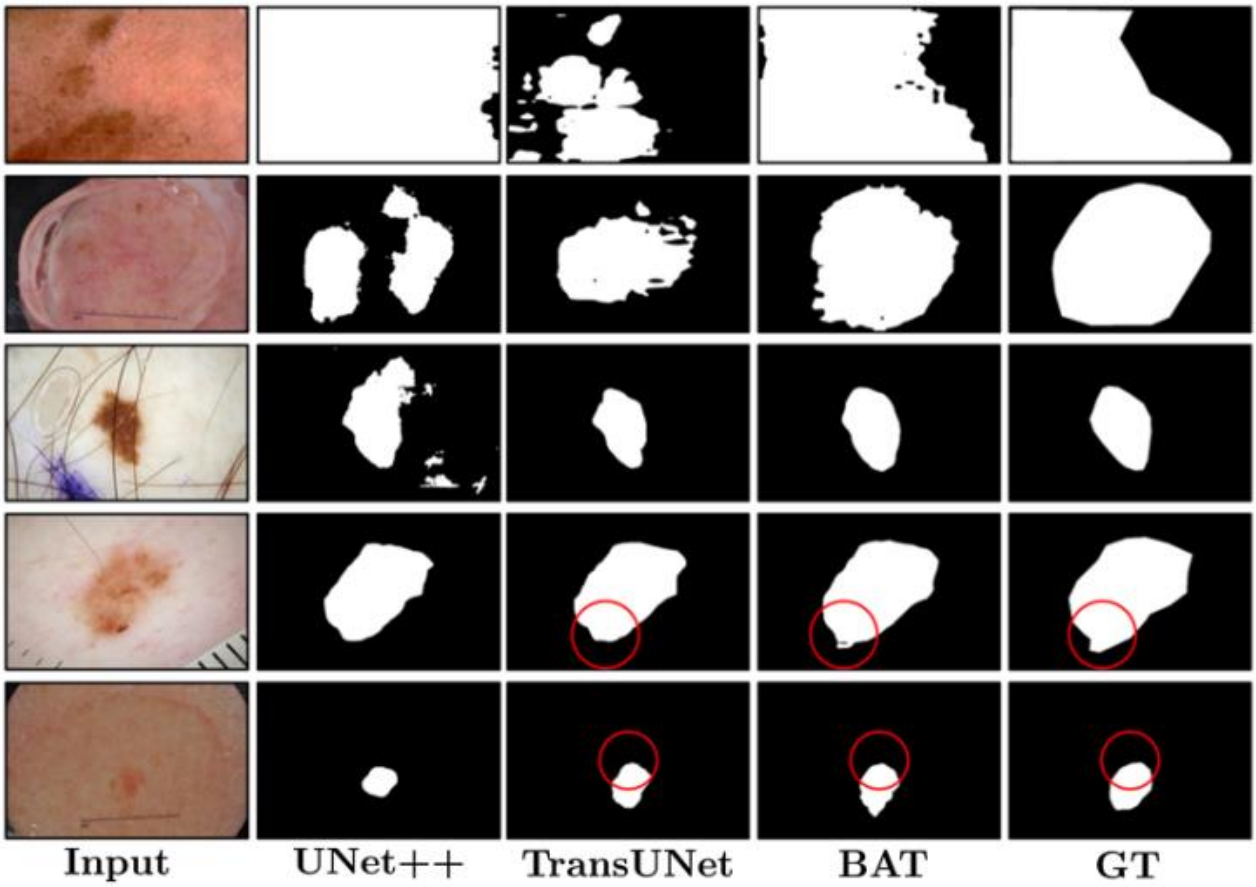


Рис. 4.47. Сравнение различных подходов к сегментации поражений кожи. Слева направо: входное изображение, UNet++ на основе CNN, TransUNet на основе ViT, трансформер с учетом границ (Boundary aware transformer, BAT) и изображение ground trooth (GT). Красными кругами выделены небольшие области с неоднозначной границей, где BAT может работать хорошо благодаря использованию априорных знаний о границах

Способность трансформеров учитывать контекст изображения делает их перспективными при решении наиболее сложных задач обработки медицинских изображений, в том числе так называемых задач регистрации медицинских изображений, где требуется оценить смещение конкретных вокселей и установить соответствия между парой фиксированных и движущихся изображений. Такие задачи жизненно важны при анализе пар изображений, полученных в разное время (например, изображения опухоли до и после химиотерапии), с разных точек зрения или с использованием разных модальностей (например, МРТ и КТ).

В целом на сегодняшний день трансформеры считаются одной из самых перспективных архитектур машинного обучения, в первую очередь для обработки последовательностей со сложной структурой взаимосвязей между компонентами. Однако они имеют и существенные недостатки, такие как сравнительно низкая скорость работы, большие ресурсные затраты на обучение и необходимость очень больших обучающих датасетов.

Вопросы для самопроверки

1. Назовите основные универсальные модели представления и обработки концептов (знаний), используемые при построении систем ИИ.
2. Дайте определение высказывания в исчислении высказываний.
3. Совпадают ли определения предиката в логике Аристотеля и в исчислении предикатов?
4. Опишите основную канву дедуктивного вывода на основе исчисления предикатов.
5. Перечислите ограничения проблемно-ориентированного языка на основе исчисления предикатов по сравнению с естественным языком.
6. Какие основные подходы к построению логических систем с изменяющимися отношениями вы знаете?
7. Опишите основную канву дедуктивного вывода на семантических сетях.
8. Являются ли сценарии сетевыми моделями?
9. Являются ли фреймы сетевыми моделями?
10. В чем отличие продукционных и логических моделей?
11. Дайте содержательное определение байесовской сети.
12. Какие основные подходы к обработке неопределенной информации в ИТ вы знаете?
13. Каковы минимально необходимые типы объектов в онтологии?
14. Какие типы связей могут использоваться в онтологиях?
15. В чем специфика рекуррентных НС по сравнению с другими видами НС?
16. В чем специфика байесовских НС по сравнению с другими видами НС?
17. Какие типы неопределенности можно оценивать посредством байесовских НС?
18. В чем специфика графовых НС по сравнению с другими видами НС?
19. Как можно выполнить операцию свертки в графовых НС?
20. Приведите примеры генеративных моделей в ИИ.
21. Как можно использовать скрытое представление, формируемое автоэнкодером?
22. Опишите принцип работы генеративных состязательных сетей (GAN).
23. Поясните пример использования трансформера в машинном переводе.
24. Возможно ли использование трансформера для обработки нетекстовых модальностей?

5. СИСТЕМЫ ИИ

«Зоопарк» систем ИИ уже достиг небывалых размеров и не перестает развиваться. Даже классификация систем ИИ, представленная в ГОСТ Р 59277–2020 (см. раздел 3.1), достаточно трудна для обозрения. В связи с этим в настоящей главе приводятся лишь примеры построения систем ИИ. Авторы ограничиваются предметной областью медицины, в которой имеют опыт собственных разработок, что позволяет выделить и продемонстрировать специфику конкретных методологических и технологических решений.

5.1. Мультимодальная система ИИ для диагностики рассеянного склероза

Постановка задачи. Диагностика заболевания – важнейший этап для успешного лечения, где требуются согласованные усилия различных специалистов, поддержанные средствами ИИ. В работе [Vatian, 2019] эта проблема рассматривается на примере диагностики рассеянного склероза, базой для которой служат МРТ-изображения головного мозга пациентов. Анализ изображений выполняет врач-рентгенолог, причем в отчете рентгенолога содержится общее описание состояния мозга пациента. Такой врач не принимает решения о наличии болезни пациента, а только описывает состояние каждой из областей мозга в текстовой форме, опираясь на свои профессиональные знания и опыт, т.е. на собственный контекст. Для выделения аномальных зон на изображении, характерных для рассенного склероза, также эффективно применение средств машинного обучения.

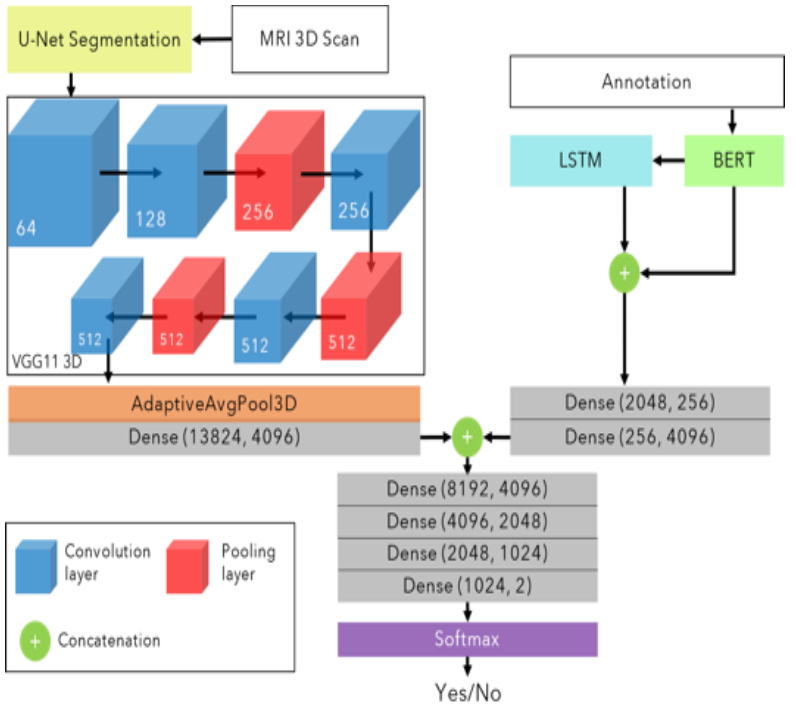
Таким образом, объединение информации об одном и том же объекте (головной мозг пациента) из разных модальностей (МРТ-изображения и их описание в виде заключений радиологов), которые частично дублируют друг друга, но содержат разный контекст, может повысить вероятность верной классификации аномальных зон средствами ИИ и тем самым увеличить эффективность диагностики, выполняемой лечащим врачом.

Архитектура. Предлагаемое решение (рис. 5.1) представляет собой комбинацию из трех модулей, которые включают в себя два типа сетевых архитектур: сверточную (CNN) и рекуррентную (RCNN). Пайплайны обработки визуальной и текстовой информации показаны на рис. 5.2.

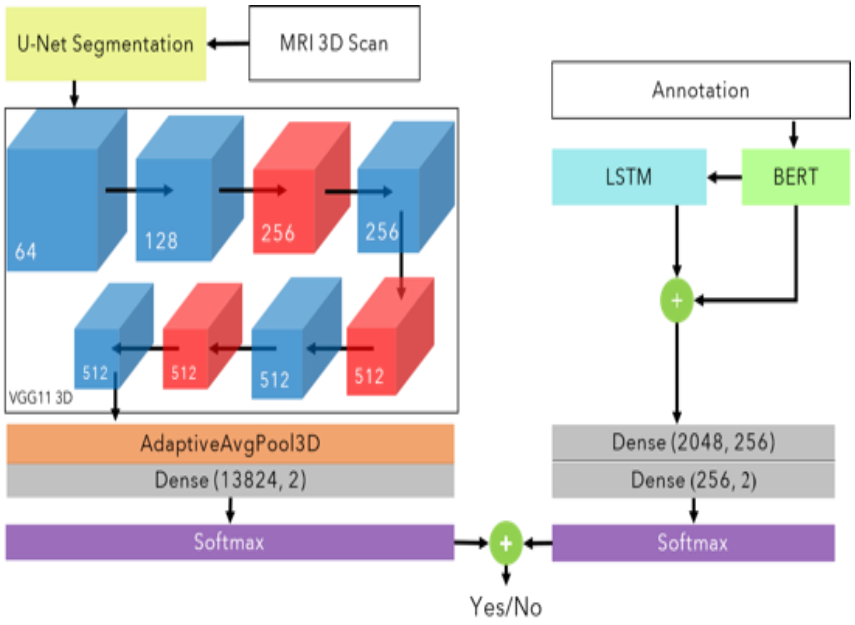
Модуль обработки изображений состоит из двух последовательно соединенных CNN типов UNet и VGG11. Серия снимков мозга пациента подается на вход модуля. UNet отмечает подозрительные места в мозге, затем VGG11 выполняет свертку, и после прохождения последнего полносвязного слоя модуль возвращает вектор признаков размерностью 4096.

Модуль обработки медицинских отчетов состоит из двух последовательно соединенных сетей – BERT и LSTM. Так как отчеты могут быть написаны на родном языке врача (в данном случае на русском), то конкретные языковые решения имеют ограниченную применимость. Чтобы сделать решение

независимым от языка, перед отправкой выводов в модуль обработки текста используется модель BERT.



а

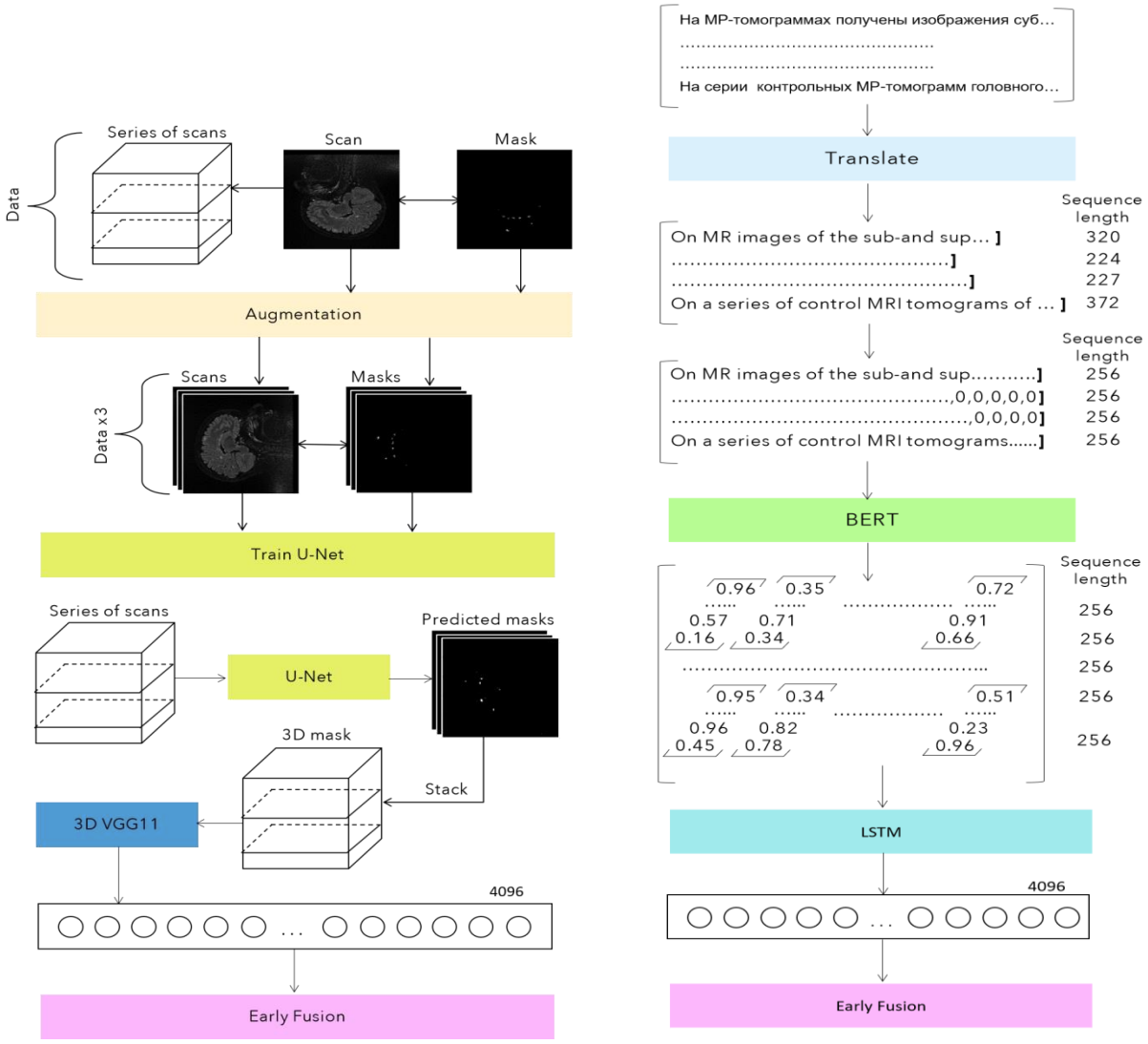


б

Рис. 5.1. Архитектура системы: а – раннее слияние, б – позднее слияние

Текст медицинских отчетов передается через предварительно обученную сеть BERT, тем самым формируя эмбеддинги слов. Эмбеддинги смежных слов объединяются и подаются на вход LSTM. Подчеркнем, что LSTM обрабатывает эмбеддинг не каждого слова как такового, а слова и его контекста, то есть, по существу, реализуется архитектура RCNN. Результат уровня LSTM для каждого элемента последовательности объединяется с эмбеддингами и, таким образом,

выравнивается и проходит через один полносвязный слой. На выходе модуль обработки текста возвращает вектор размерностью 4096.



а б

Рис. 5.2. Пайплайн обработки изображения (а) и текста (б)

Таким образом, и модуль обработки текста, и модуль обработки изображений генерируют векторы признаков одинакового размера (4096). Размер векторов, подлежащих объединению, был выбран экспериментально. Он оказался весьма выразительным для оптимизации процесса обучения сети (при меньшей размерности векторов сеть не обучалась оптимально).

В методе раннего слияния (рис. 5.1, а) конкатенированные векторы проходят через модуль слияния (fusion), состоящий из нескольких последовательных полносвязных слоев. После каждого из полносвязных слоев, кроме последнего, применялась активация Dropout и LeakyReLU, на последнем слое – активация Softmax. В качестве функции ошибки использована кросс-энтропийная функция потерь, являющаяся стандартом для задач классификации.

В методе позднего слияния (рис. 5.1, б) модули изображений и текста обучаются отдельно. Последние полносвязные слои обоих модулей заменяются полносвязными слоями с числом выходных элементов, равным 2. Затем для каждого

выхода оценивается функция потерь и определяется среднее значение для полученных функций потерь.

Таким образом, предлагаемый подход представляет собой комплексное решение, на вход которого подается изображение и текст, а на выходе получается вероятность наличия рассеянного склероза для данного пациента.

Датасеты и предварительная обработка данных. В качестве наборов данных были использованы серии МРТ-изображений 19 пациентов, у которых подозревался рассеянный склероз. Каждая серия включала от 170 до 270 снимков. Все изображения были размечены и описаны квалифицированными рентгенологами. Используемая выборка была достаточно сбалансированной, а именно, в подгруппе обучения соотношение здоровых и больных пациентов составляло 6:5, в подгруппе валидации ровно половина пациентов была здорова, а половина болела.

Для обучения сети U-Net были использованы следующие типы аугментации: HorizontalFlip, VerticalFlip, ElasticTransform, GridDistortion. После аугментации общее количество изображений для обучения составило ≈ 19 000. Для обучения и валидации U-Net наборы данных были разделены в соотношении 80:20%.

Сеть VGG11 была обучена на трехмерных массивах в следующей пропорции: обучающая выборка составляла 60% набора данных (11 пациентов), а тестовая выборка – 40% (8 пациентов).

Предварительная обработка текстовых отчетов проводилась в два этапа. Первоначально был осуществлен автоматический перевод текстов на английский язык для дальнейшего использования сетей, предварительно обученных по английскому контенту. Затем, используя предварительно обученную сеть BERT, для каждого текста была получена последовательность эмбеддингов длиной 768 токенов. Подчеркнем, что в реализации BERT используются последовательности токенов только идентичной длины (в нашем случае средняя длина отчета составляла 256 токенов). Чтобы выровнять длину входных текстов, применялось заполнение нулями или, соответственно, обрезка каждого текста длиной до 256.

Метрики и экспериментальная оценка. В качестве метрики для общей оценки предложенного подхода мы использовали точность классификации. В нашем случае эта метрика вполне допустима из-за вышеупомянутого баланса используемых наборов данных. С другой стороны, из-за небольшого объема наборов данных график точности получается негладким и, следовательно, нерепрезентативным. Поэтому в качестве характеристического значения точности рассматривалось значение, соответствующее эпохе с наименьшим значением функции потерь. По сути, в этом случае использована ранняя остановка, когда для проверки выбирается лучшая эпоха, не дожидаясь выхода сети из первого найденного локального минимума.

Чтобы сравнить эффективность подходов с объединением информации и без нее, в качестве бейзлайна использован результат классификации с единственным модулем обработки изображений (см. рис. 5.1, а, левая часть, и рис. 5.2, а). Использование других датасетов для формирования бейзлайна приведет к неверным результатам сравнения. Кроме того, насколько известно

авторам, доступные датасеты по рассеянному склерозу не имеют аннотаций. На рис. 5.3, а, показана функция потерь, полученная при обучении только на изображениях. Можно видеть, что в этом случае сеть сильно переобучена и требует больше данных для достижения качественных результатов.

Чтобы сравнить подходы раннего и позднего слияния, была проанализирована кросс-энтропийная функция потерь при валидации с каждым типом слияния соответственно (рис. 5.3, а, б). Видно, что в случае позднего слияния сеть переобучена и требует больше данных.

а б в

Рис. 5.3. Кросс-энтропийная функция потерь: а – использование только модуля обработки изображений; б – раннее слияние; в – позднее слияние

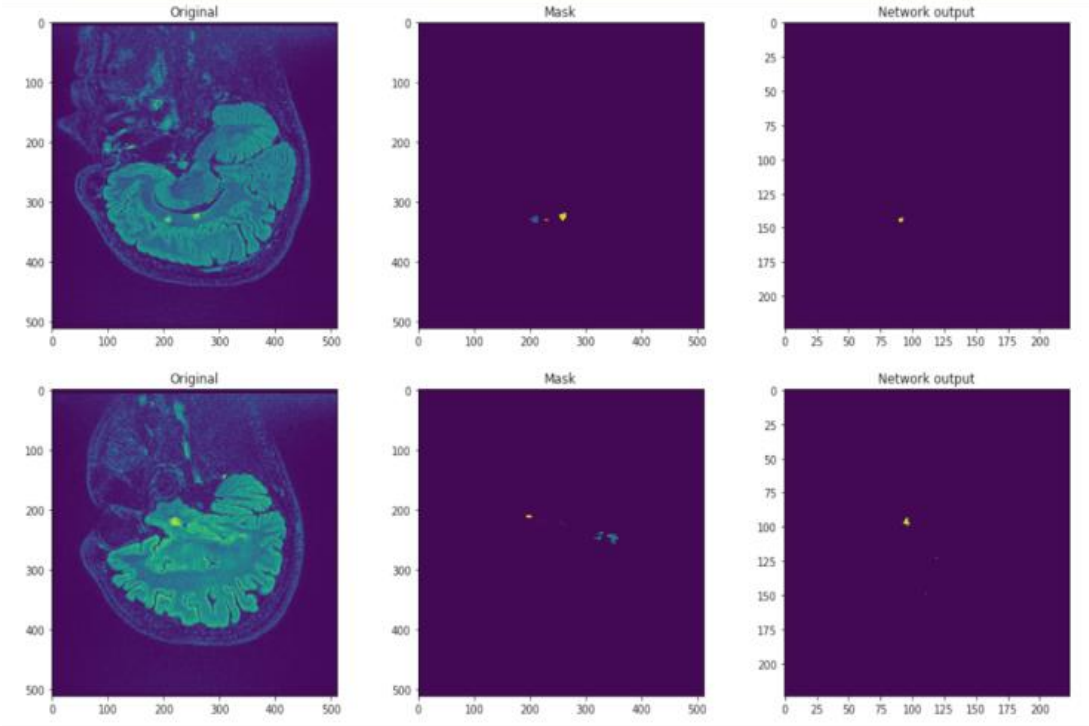


Рис. 5.4. Исходное изображение (слева), маска очагов MS, нарисованная рентгенологом (в центре), предсказанная системой маска (справа)

Таблица 5.1. Оценки эффективности разработанной системы

Метод	Точность
Без слияния (только модуль обработки изображения)	62,5%
Позднее слияние	75%
Раннее слияние	87,5%

Сравнительные оценки разработанной системы проиллюстрированы на рис. 5.4 и представлены в табл. 5.1. Видно, что с помощью метода раннего слияния получена точность 87,5% на подмножестве валидации, что является весьма высоким результатом; кроме того, рис. 5.3, б, свидетельствует, что обучение при раннем слиянии обеспечивает значительно меньшую нагрузку на сеть и, следовательно, более высокое конечное качество обучения. Однако обучение при позднем слиянии более стабильно (рис. 5.3, в): такую сеть легче обучить, поскольку процесс выбора оптимальных гиперпараметров происходит быстрее и проще. Таким образом, обучение при раннем слиянии является более трудоемким, но оно позволяет получить лучшие результаты при меньшем количестве данных, а обучение при позднем слиянии проще, но требует больше данных.

5.2. Сегментация тканей мозга на основе графовых нейронных сетей

Постановка задачи. Сегментация мозговой ткани на основе снимков магнитно-резонансной томографии (МРТ) имеет большое значение – например, для выделения границ опухоли мозга. МРТ-изображение – это псевдо-3D изображение, которое реконструируется в томографе из параллельно расположенных сканов мозговой ткани. При этом возникают эффекты смещения, которые провоцируют ошибки сегментации. Работа [Yan, 2019] направлена на преодоление этих проблем.

Метод. В работе предлагается вместо попиксельной обработки каждого скана с их последующим соединением в псевдо-3D структуру перейти к обработке супервокселей, которые генерируются из МРТ-изображения с помощью улучшенного простого линейного итеративного алгоритма кластеризации. Затем из этих супервокселей с помощью алгоритма K-ближайших соседей строится граф, после чего он передается в графовую сверточную сеть (ГСН) для классификации тканей (рис. 5.5).

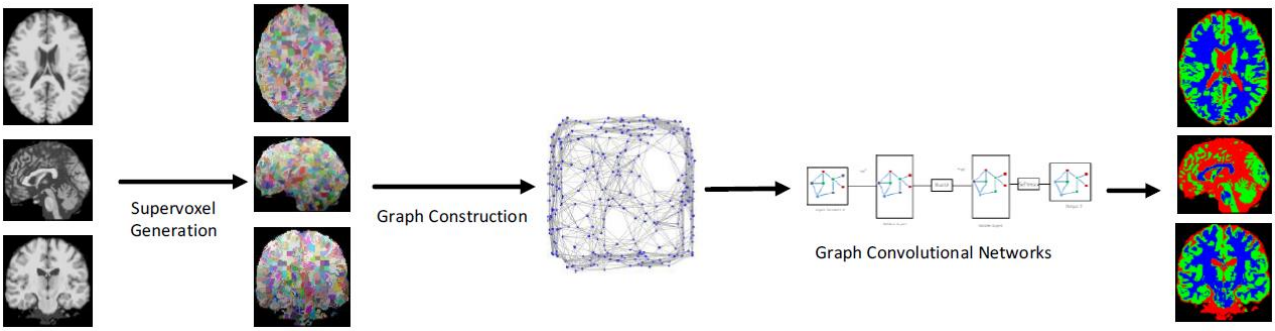


Рис. 5.5. Схема работы системы

Супервоксели генерируются путем агрегирования отдельных вокселей с идентичными характеристиками посредством алгоритма K- ближайших соседей в соответствии со следующей итеративной процедурой:

$$d_{\text{int}} = \left\| I_{N_i} - I_C \right\|_2,$$

$$D = d_{\text{int}} + \gamma d_{spa},$$

где I_{N_i} и I_C – интенсивности текущего и центрального вокселей, $\|\cdot\|_2$ – l_2-норма, d_{int} и d_{spa} – разность интенсивностей и эвклидово расстояние между текущим и центральным вокселями соответственно, γ – весовой коэффициент для настройки степени шумоподавления. В результате псевдо-3D изображение мозга преобразуется в граф из связанных супервокселей.

После построения графа необходимо провести классификацию вершин графа, то есть, в данном контексте, идентификацию тканей мозга. Это проблема semi-supervised обучения из-за того, что лишь несколько тканей мозга (спинномозговая жидкость, серое вещество и белое вещество) заранее известны. Для ее решения используется двухслойная сеть ChebNet (рис. 5.6).

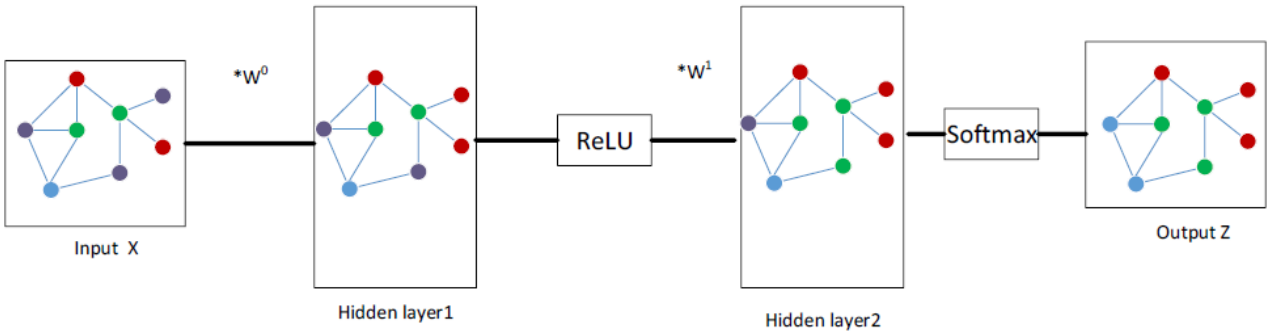


Рис. 5.6. Иллюстрация процедуры классификации супервокселей с помощью ChebNet. Цвета (красный, зеленый, синий и фиолетовый) представляют спинномозговую жидкость, серое вещество, белое вещество и неразмеченную ткань. W^0 и W^1 – обучаемые матрицы весов, * – оператор свертки

Оценка результатов. Для количественной оценки использовались два показателя – коэффициент сходства по Дайсу (DSC) и коэффициент объемной разницы (VDR):

$$DSC = \frac{2 \times TP}{2 \times TP + FP + FN},$$

$$VDR = \frac{|FP - FN|}{TP + FN},$$

где TP, FP и FN – истинно положительные, ложноположительные и ложноотрицательные воксели. Значение DSC отражает сходство между результатом сегментации и истинной информацией, в то время как VDR измеряет разницу между ними. Более высокий DSC и более низкий VDR указывают на лучший результат сегментации.

Эксперименты проводились на двух широко используемых наборах данных: датасете BrainWeb18 и наборе данных интернет-репозитория сегментации мозга

18 (IBSR18). Изображения МРТ моделировались с уровнем неравномерности интенсивности (INU) 40% и с уровнем шума до 9%.

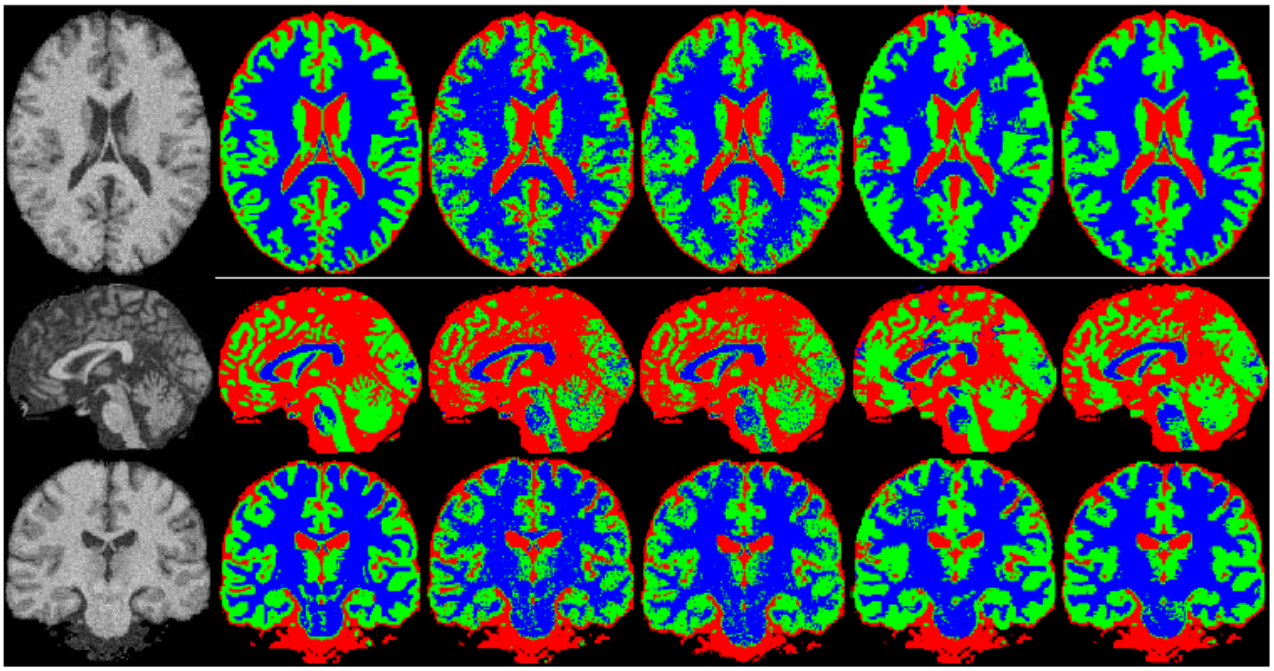


Рис. 5.7. Сравнение эффективности сегментации тканей мозга различными методами: ground-truth – ручная сегментация; FCL – набор библиотечных функций, выполняет выделение трех типов мозговых тканей (спинномозговую жидкость, серое вещество, белое вещество); SPM8 – модель описывает распределение яркостей вокселей в разных тканях мозга как смесь гауссоид с настраиваемыми параметрами; AGF – модель задает каждый супервоксель вектором признаков, описывающих гистограмму интенсивности, текстуру и форму; GCN – предложенный метод

Эксперименты показали, что предложенная система ИИ для сегментации мозговых тканей практически во всех случаях превосходит аналоги (рис. 5.7) на 1–2%, что является очень хорошим результатом для обработки медицинских изображений. В то же время система отличается простотой в настройке и программной реализации, что делает ее перспективной для дальнейшего развития.

5.3. Бенчмаркинг систем синтеза медицинских изображений на основе GAN на пилотной стадии проектирования

Постановка задачи. Генеративно-состязательные сети (GAN) нашли широкое применение для формирования искусственных, но реалистичных изображений самого различного содержания, в том числе в медицине. За последние два-три года спектр сценариев использования GAN в медицине резко расширился. GAN в области медицины используются как вспомогательное средство для аугментации датасетов в машинном обучении. С помощью GAN формируют медицинские изображения различной природы, которые можно использовать в

качестве эталонных изображений при настройке автоматизированных классификаторов соответствующих заболеваний, а также для обучения менее опытных патологоанатомов и рентгенологов.

Особенно важно моделирование тканей, пораженных заболеванием, так как они встречаются редко и в то же время отличаются большим разнообразием, что затрудняет построение сбалансированных датасетов для обучения медицинских диагностических систем ИИ.

Один из наиболее актуальных примеров – диагностика рака легких с применением систем ИИ. Основную диагностическую ценность в изображении легочной ткани представляют легочные узлы – периферические образования в лёгочной ткани размером менее 3 см. Чаще лёгочный узел представляет собой доброкачественное новообразование, однако в 20% случаев обусловлен злокачественной опухолью, особенно у пожилых пациентов и курильщиков. Для повышения точности диагностики нужно уметь различать зло- и доброкачественные легочные узлы. Для этого требуются обширные датасеты, создание которых вручную – крайне сложная и ресурсозатратная проблема для радиологов. В связи с этим возникает задача генерации искусственных изображений легочных узлов на реальных компьютерных томограммах (рис. 5.8).

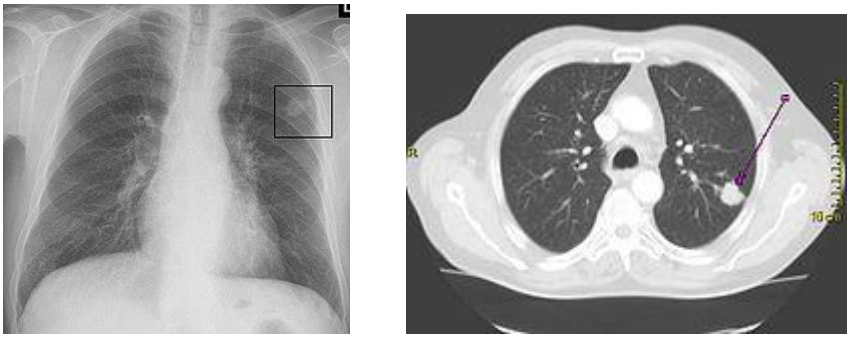


Рис. 5.8. Изображение легочного узла: а – рентгенограмма,
б – компьютерная томограмма

Рис. 5.9. Пример сгенерированного изображения легочного узла: а – вид узла на отдельных слайсах КТ-изображения; б – гистограмма яркости узла по слайсам

Анализируя использование ГАН для генерации изображений легочной ткани, можно видеть пеструю картину различных архитектур и подходов

реализации GAN, а также оценки качества получаемых изображений. В работе [Semiletov, 2021] рассматривается задача бенчмаркинга таких систем еще на пилотной стадии их разработки, т.е. в условиях ограниченных временных и вычислительных ресурсов. Для этого необходимо:

- провести пилотную разработку нескольких GAN с различной архитектурой и размерностью формируемого изображения, предназначенных для генерации изображений легочной ткани с раковыми узлами, в условиях ограниченных ресурсов;
- выбрать набор метрик для оценки качества формируемых изображений и параметров, необходимых для их адаптации к легочной ткани;
- сравнить построенные GAN по выбранным метрикам и выявить ключевые параметры влияния, которые необходимо учитывать при дальнейшей разработке.

Архитектурные решения и обучение GAN. Для сравнения были разработаны два варианта GAN, отличающихся как архитектурой, так и подходом к формированию изображения (в 3D или 2D проекции).

Для 3D-GAN в качестве исходной архитектуры выбран CT-GAN [Mirsky, 2019] с открытым кодом (рис. 5.10), но в него были внесены некоторые изменения, которые включали обновление веса генератора и дискриминатора, соотношение частот и объединение потерь Вассерштейна (WL) с базовой среднеквадратической ошибкой (MSE). Кроме того, мы использовали AdaIN вместо пакетной нормализации после каждого блока свертки с параметром нормализации:

$$AdaIN\left(x,y\right)=\sigma\left(y\right)\frac{x-\mu(x)}{\sigma\left(y\right)},$$

где x – выход предыдущего слоя, y – аффинное преобразование. Модифицированные блоки 3D-GAN изображены на рис. 5.11–5.12.

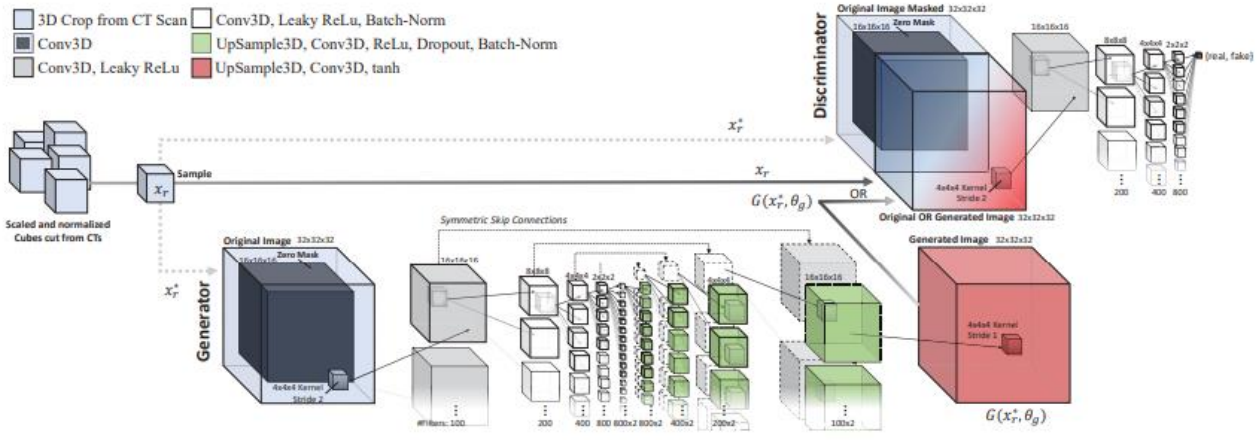


Рис. 5.10. Архитектура CT-GAN

Для 2D-GAN в качестве исходной архитектуры выбран Style-GAN [Karras, 2019] с открытым кодом. Поскольку исходная архитектура предназначена для работы с трехканальными изображениями, количество каналов во входном и выходном слоях было преобразовано для работы с одноканальными изображениями. Был также изменен тип функции Loss на BinaryCrossEntropy. В качестве

классификатора использована нетребовательная к вычислительным ресурсам модель VGG11 [Simonyan, 2015].

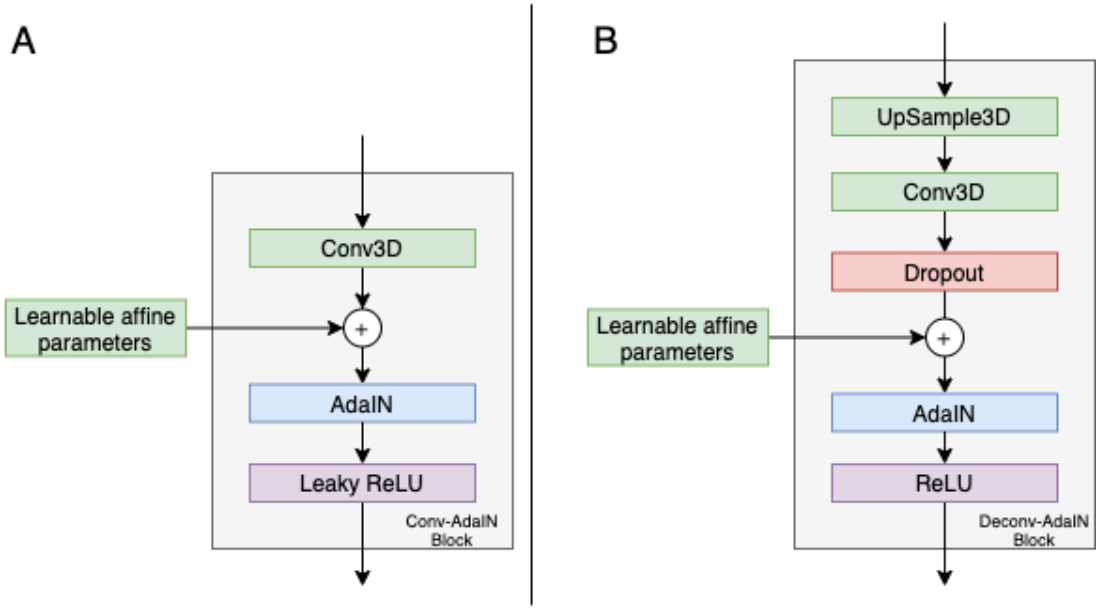


Рис. 5.11. Модифицированные блоки 3D-GAN: (A) Сверточный блок, (B) Деконволюционный блок

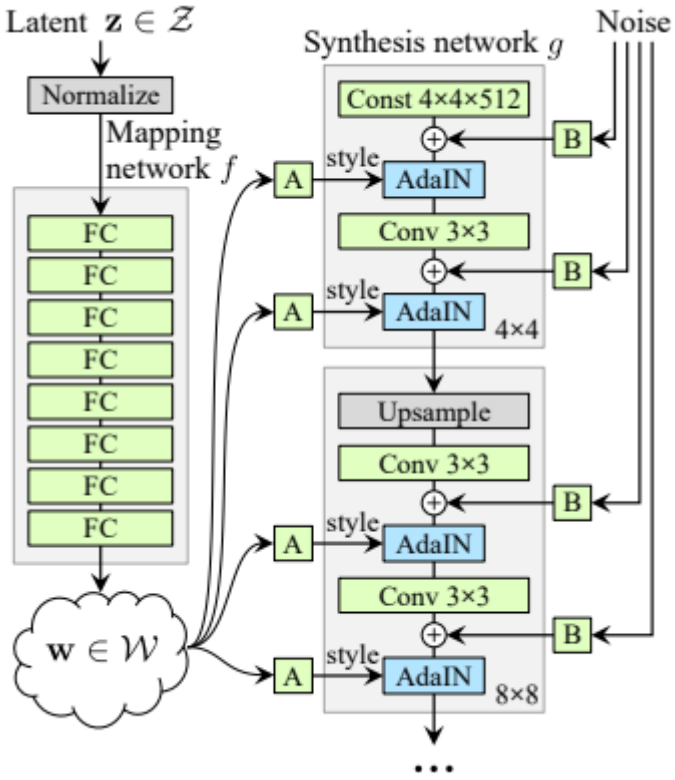


Рис. 5.12. Архитектура Style-GAN

Для обучения обеих моделей GAN использован открытый набор данных Консорциума базы данных изображений легких и Инициативы по ресурсам базы данных изображений (LIDC-IDRI). Для 3D-GAN использовано его подмножество LUNA-16, имеющее следующие свойства по сравнению с базовым набором данных LIDC: все конкреции откалиброваны более точно (средний размер

конкреций в LUNA16 составляет 8,3 мм при стандартном отклонении 4,8 мм) по сравнению с LIDC-IDRI (12,8 мм и 10,6 мм соответственно); каждый узелок уже помечен ограничивающей рамкой.

Для обучения 2D-GAN был взят LIDC-IDRI в целом, из которого был отобрана серия DICOM-снимков только с опухолевыми узлами. Для извлечения узла формировался описывающий куб, содержащий сам узелок и окружающие (контекстные) ткани. Абсолютные размеры куба выбирались в соответствии с размером извлеченного узелка (от 1 мм до 40 мм), но передискретизировались до единичного размера 1283 пикселя и до значений пикселей по шкале Хаунсфилда:

$$x = \frac{2 \cdot \left(x_{in} - in_{\min} \right)}{\left(\left(in_{\max} - in_{\min} \right) - 1 \right)},$$

с граничными значениями $in_{\max}$=800, $in_{\min}$ = –1000. Чтобы перейти от 3D-изображений к построению 2D-узлов, использовалось MIP-текстурирование (MIP-mapping, лат. multum in parvo – «много в малом») – метод текстурирования, использующий несколько копий одной текстуры с разной детализацией. Операция MIP выполнялась в пакете *numpy*.

Методы оценки качества синтезированных изображений. Методы оценки, используемые в этом случае, можно разделить на две группы – меры интегральной эффективности и модельные методы.

Интегральная эффективность 3D- и 2D-GAN измерялась как эффективность классификации легочных узлов, выполняемой внешней глубокой нейронной сетью (использована ГНС [Li, 2020]) на наборах данных, дополненных с помощью построенных GAN. Основной метрикой в этом случае является ROC AUC (Area Under ROC Curve). Для оценки классификатора на основе 2D-GAN дополнительно использована метрика PR AUC (Precison-Recall Area Under Curve), а для классификатора на основе 3D-GAN – метрика FROC (Free Response Receiver Operating Characteristic).

Для визуальной оценки качества классификации использована метрика t-SNE [van der Maaten, 2008], которая позволяет сравнивать распределения реальных и сгенерированных изображений путем перевода данных высокой размерности в пространство меньшей размерности.

Для оценки качества генерированных изображений узлов предложены специализированные модельно-независимые метрики.

- Начальное расстояние Фреше (Frechet Inception Distance, FID) [Brownlee, 2019] вычисляет расстояние между векторами признаков, рассчитанными для реальных и сгенерированных изображений. FID основан на предположении о многомерном распределении Гаусса реальных и сгенерированных изображений, т.е. сравниваются среднее значение и стандартное отклонение, поэтому формула выглядит следующим образом:

$$d^2((m,C),(m_w,C_w)) = \|m - m_w\|_2^2 + Tr\left(C + C_w - 2\left(CC_w\right)^{1/2}\right),$$

где Tr(.) – след ковариационных матриц векторов признаков *C* и *Cw*. Более низкие значения FID хорошо коррелируют с изображениями более высокого качества.

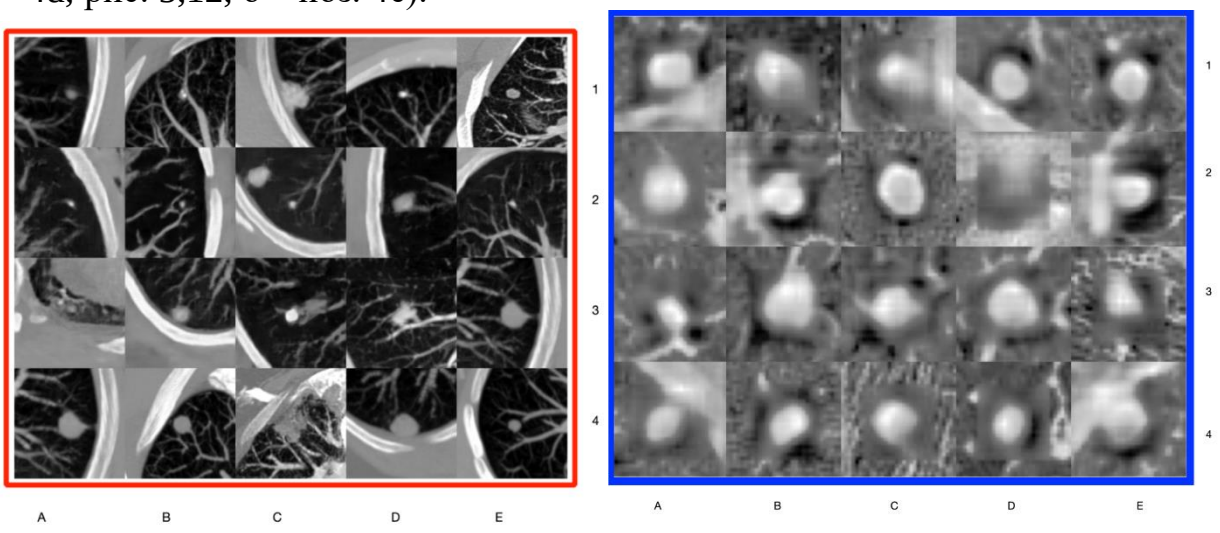
- Максимальное среднее расхождение (Maximum mean discrepancy, MMD) [Tunali, 2020] – мера расстояния между двумя распределениями, которая определяется как квадрат расстояния между их эмбеддингами в гильбертово пространство воспроизводящего ядра F:

$$MMD^2(P,Q) = \left\| \mu_P - \mu_Q \right\|_F^2.$$

Чем ниже значение MMD, тем более похожи распределения.

Для модельной оценки применялся визуальный тест Тьюринга [Salimans, 2016] в проблемно-ориентированной модификации. Как и в традиционном визуальном тесте Тьюринга, N опытным радиологам демонстрируются реальные и виртуальные (сгенерированные GAN) изображения. Каждому радиологу предоставляется S наборов, содержащих 20 случайно выбранных изображений, сгенерированных определенной GAN, и сообщается, что экспонированный набор может содержать любую смесь реальных и сгенерированных изображений. Примеры представленных комплектов показаны на рис. 5.13. Врачу-рентгенологу предлагается ответить на вопросы:

- Если в представленном наборе есть узелки, то какие из них настоящие?
- Если в представленном наборе есть узелки, то какие из них одиночные, а какие пристеночные? (Примеры пристеночных узлов – рис. 5.12, а – поз. 1c и 4d; рис. 5,12, б – поз. 4e).



а б

Рис. 5.13. Примеры презентаций: а – для оценки 2D-GAN; изображения 1–10 содержат узлы, созданные 2D-GAN, изображения 11-20 полностью реальны; б – для оценки 3D-GAN; все изображения содержат узлы, созданные 3D-GAN

По результатам теста рассчитывается коэффициент ложного распознавания (FRR) как доля узлов, правильно идентифицированных радиологами, среди всех сгенерированных узлов (рис. 5.14).

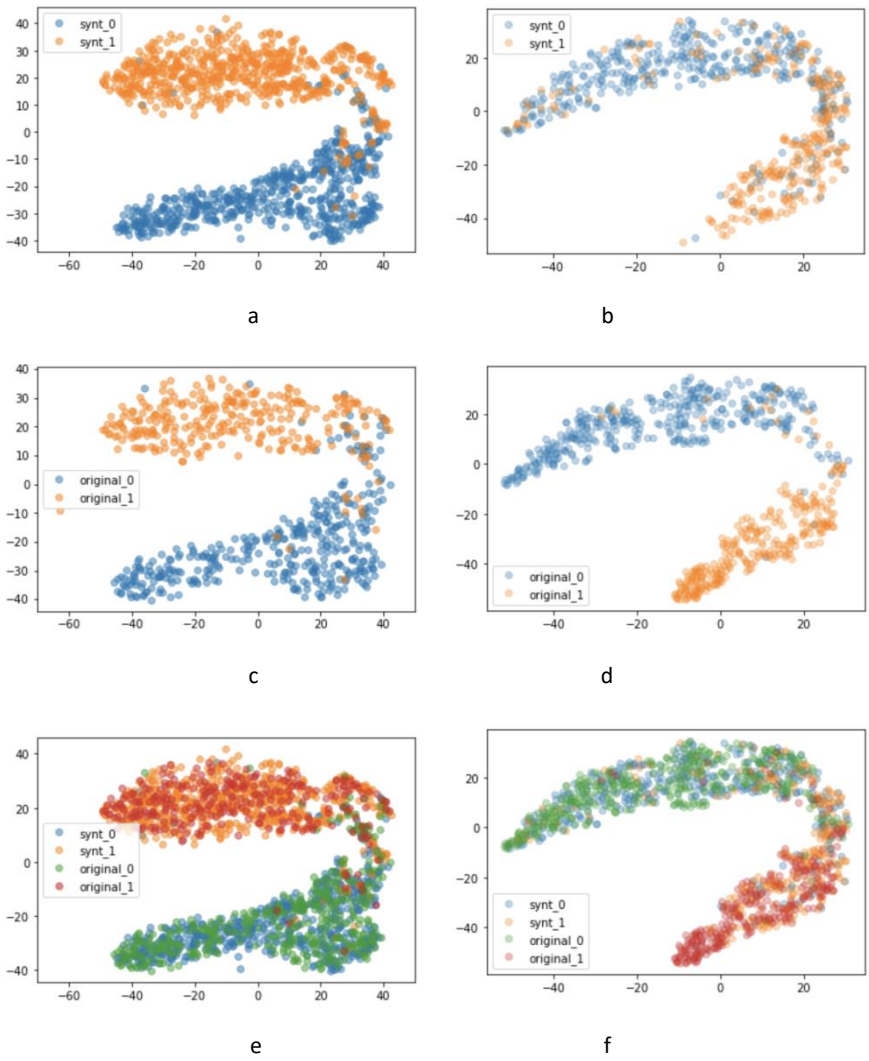


Рис. 5.14. Визуализированные метрики t-SNE для различных комбинаций эмбеддингов изображений с легочными узлами: а – доброкачественные и злокачественные узлы, сгенерированные 2D-GAN; б – доброкачественные и злокачественные узлы, созданные с помощью 3D-GAN; с – доброкачественные и злокачественные узлы, реальные изображения; г – доброкачественные и злокачественные узлы, реальные изображения; е – все узлы набора данных, дополненного 2D-GAN; f – все узлы набора данных, дополненного 3D-GAN

Экспериментальные результаты. Экспериментальные оценки интегральной эффективности ГНС с датасетами, аугментированными GAN, составили: для 2D-GAN – ROCAUC=0.9604, PRAUC=0.9625; для 3D-GAN – ROCAUC=0.95. Это лучше, чем результат [Onishi, 2019] на сопоставимом наборе данных, и лишь немного уступает результату тех же авторов, полученному на гораздо более мощных вычислительных ресурсах. Визуальное качество классификации, оцениваемое по t-SNE (рис. 5.14), также вполне удовлетворительно. Таким образом, с точки зрения аугментации наборов данных, используемых для обучения ГНС, разработанные GAN находятся на уровне существующих решений, и их правомерно использовать как базу для отбора метрик для бенчмаркинга.

При анализе визуального качества сгенерированных изображений важно количество рентгенологов, выполняющих визуальный тест Тьюринга, и их

профессиональный опыт; привлечение каждого дополнительного эксперта влечет увеличение затрат. Результаты наших экспериментов показывают, что при переходе от усреднения значений FRR по 6 экспертам (19,4±5,4% и 41 ±11% для 2D-GAN и 3D-GAN соответственно) к усреднению по 2 экспертам (20,6±8% и 35,8±14% для 2D-GAN и 3D-GAN соответственно) математическое ожидание, хотя и изменялось, но оставалось в пределах стандартного отклонения. Это позволяет говорить о том, что при пилотных исследованиях архитектуры GAN можно ограничиться двумя экспертами-радиологами, но использование только одного рентгенолога и тем более неспециалиста не гарантирует от наличия смещений в оценке и не может быть оправдано. В то же время для масштабных исследований необходимо использовать более сильные статистические меры подобия, например каппу Флейсса [Gwet, 2008].

Анализируя модельно-независимые метрики качества изображения, можно констатировать, что метрики MMD и PID демонстрируют разное ранжирование качества получаемых изображений. При этом изменение функции ядра, хотя и влияет на абсолютные значения метрики ММД, но не меняет ее соотношения для разных групп изображений. При этом, по результатам экспериментов, градация метрик по FID и MMD в 2D не совпадает. Это связано с тем, что метрика FID, в отличие от MMD, учитывает не только средние значения, но и дисперсию данных. Например, на визуализации t-SNE (рис. 5.14, в) видно, что реальные данные имеют несколько выбросов. Из-за них среднее реальных и сгенерированных данных отличается, но на дисперсию это практически не влияет. Поэтому для бенчмаркинга метрику FID можно считать более справедливой, чем метрику MMD. Кроме того, необходимо подчеркнуть самостоятельную роль метрики t-SNE в оценке качества формируемых изображений легочных узелков, поскольку она позволяет визуально оценить степень взаимопроникновения формируемых объектов разных классов (ср. рис. 5.14, а и б).

Таким образом, набор метрик и их параметров для оценки качества формируемых изображений тканей легких в составе:

визуальный тест Тьюринга, выполненный не менее чем двумя рентгенологами
+ метрика t-SNE + метрика FID,

представляется весьма показательным для принятия решения о перспективах той или иной архитектуры GAN.

Сравнивая разработанные GAN, можно сделать следующие выводы. Хотя, как отмечалось выше, оба GAN обладают достаточно высокой эффективностью в плане аугментации наборов данных для обучения глубоких нейронных сетей, тем не менее визуальное качество изображений, созданных 3D-GAN, значительно различается. Это можно связать с тем, что большинство существующих GAN ориентировано на генерацию относительно крупных новообразований, а задача их вставки в существующие изображения сводится к устранению дефектов на границах между формируемым изображением и подложкой (нативной тканью).

В случае небольших узелков этот эффект не наблюдается, т.е. не удается четко выделить наиболее критичные для визуальной оценки зоны формируемого изображения. Таким образом, прямое масштабирование решений для крупных

новообразований до малых размеров вряд ли эффективно, а имитацию мелких новообразований следует рассматривать как отдельную задачу при построении. Помимо использованного в работе MIP-текстурирования, здесь возможны и другие варианты. Поэтому относительный размер моделируемого медицинского новообразования является одним из важных параметров влияния, который необходимо выделить и оценить при разработке архитектуры GAN для медицинских целей.

5.4. Подход «human-in-the-loop» в онлайн системе обработки и оценки медицинских изображений

Постановка задачи. Профилактика и лечение заболеваний, вызывающих наибольшее количество смертей в мире, таких как ишемическая болезнь сердца, инсульт, хроническая обструктивная болезнь легких, рак легких и т.д., все больше опирается на высокие медицинские технологии, где получение информации связано с наличием соответствующего оборудования, а его использование для диагностики и лечения – с возможностью его адекватной интерпретации.

В развитых в медицинском отношении странах пациенты не только имеют доступ к высокотехнологичным медицинским обследованиям, но и могут практически сразу получить их результаты, например, через портал для пациентов, без личной встречи с врачом. Неверная или неадекватная (без учета состояния здоровья в целом) интерпретация такой информации самим больным может вызвать у него психологические и социальные проблемы, не говоря уже о лечении самого заболевания. Однако спрос на такую информацию со стороны пациентов во всем мире только возрастает.

Причин, провоцирующих такую ситуацию, несколько. Первая – объективно существующая нехватка специалистов, например, квалифицированных рентгенологов, что является серьезной мировой проблемой [Allyn, 2020], особенно в контексте пандемии COVID-19.

Проблема усугубляется тем, что даже квалифицированные специалисты не свободны от ошибок. Например, о расхождениях между двумя экспертами-интерпретаторами сообщалось в 22–57% случаев исследования изображений [Sawan, 2017].

Еще одна причина, особенно характерная для развивающихся стран, –это недоверие к врачам, вызванное неудовлетворенностью пациентов существующим медицинским обслуживанием. В большинстве промышленно развитых стран уровень удовлетворенности системой здравоохранения достаточно высок – около 90% респондентов в Швеции и Швейцарии, около 70% в Великобритании, Германии, Испании и Словении. В США только 43% респондентов удовлетворены качеством медицинских услуг, а 28% недовольны. Много недовольных состоянием системы здравоохранения отмечается в России и Бразилии – 50 и 67% соответственно [Romir, 2020].

В этой ситуации резко возрастает значимость для пациентов так называемого второго мнения. Второе мнение – это практика получения дополнительной консультации врача-специалиста по результатам медицинского обследования для уточнения диагноза и плана лечения. Наиболее востребованным среди

пациентов является получение второго мнения по оценке высокотехнологичной медицинской информации, например, второго мнения врача-рентгенолога (повторный анализ диагностических изображений – КТ, МРТ, рентгенографии, маммографии, ПЭТ-КТ и др.) или кардиолога (ЭКГ, МРТ сердца и др.).

В принципе, такой анализ врач-эксперт может провести дистанционно, используя современные телекоммуникации. В этом случае пациент получает полное и официально подтвержденное экспертное мнение, которое должен соответственно оплатить. Такие сервисы широко представлены в Интернете. Однако переход к телемедицине не может полностью решить ни проблему отсутствия специалистов высокого класса, ни проблему финансовой недоступности таких услуг в развивающихся странах. Именно поэтому для интерпретации своей высокотехнологичной медицинской информации люди предпринимают попытки использовать доступные интернет-источники и коллективный сетевой интеллект, т.е. апеллировать к коллективному мнению. Действительно, большинство современных платформ социальных сетей имеют сообщества, в которых пользователи обсуждают свои медицинские проблемы (подробный обзор также представлен ниже) и посредством «спонтанного краудсорсинга» получить какое-то коллективное мнение. Однако вопросы правомерности и авторитетности такого мнения остаются открытыми.

Перспективным способом решения проблемы является автоматизированная интерпретация высокотехнологичных медицинских информации средствами искусственного интеллекта. Однако эффективность систем, основанных на принципах машинного обучения, критически зависит от достаточного количества и качества данных, собранных в реальных клинических условиях и размеченных профессиональными экспертами. Такие датасеты являются большой интеллектуальной ценностью и приносят существенные социальные выгоды их владельцам.

Высокотехнологичная медицинская информация, содержащаяся в запросах пациентов, может служить хорошей основой для формирования таких наборов данных, но для этого необходимы соответствующие организационно-мотивационные процедуры и технические решения. Таким образом, возникает проблема интеграции социальных медиа-платформ и специализированных веб-ресурсов для эффективного использования высокотехнологичной медицинской информации, что особенно актуально для медицины развивающихся стран.

Структура веб-ресурса. Для решения этой проблемы в результате комплексной работы авторов [Boitsov, 2019; Lobantsev, 2020; Pchelkin, 2021; Kozyreva, 2021] создан специализированный веб-ресурс (рис. 5.15). Ресурс работает по принципу телерадиологии. Основными акторами этого ресурса являются пациент, врач-эксперт, врач-стажер и владелец ресурса.

Основной сценарий использования ресурса выглядит следующим образом. Пациент, желающий получить «второе мнение» относительно своего заболевания, загружает свои медицинские изображения и другие диагностические материалы (при наличии) на ресурс. При этом он соглашается на использование своих диагностических материалов (в анонимизированном виде) для формирования соответствующих датасетов. Врач-эксперт, с другой стороны, имеет под своим руководством команду врачей-стажеров, которые желают получить ценные навыки

в области радиологии посредством работы на реальных, а не учебных примерах. Он передает полученный комплект материалов стажеру, который проводит разметку снимка и формулирует предварительное мнение по диагностике заболевания. Это мнение обсуждается в группе стажеров под руководством врача-эксперта и в окончательном виде передается пациенту в качестве «второго мнения». Окончательные результаты разметки снимков с необходимой сопроводительной информацией передаются владельцу ресурса для внесения в постоянно пополняемый датасет.

Рис. 5.15. Диаграмма вариантов использования ресурса

Выделим преимущества предлагаемого решения.

- Существенной особенностью ресурса является работа в полностью дистанционной форме: обсуждения, консультации и, главное, разметка и анализ изображения производятся в вебе, без необходимости личного присутствия и стационарного рабочего места для разметки, что очень привлекательно для врачей на фоне их высокой ежедневной рабочей нагрузки.
- Ресурс может использоваться как средство поддержки учебного процесса в учреждениях медицинского образования, что облегчает работу преподавателей, которые в данном случае выступают в роли врачей-экспертов.
- Ресурс имеет высокий уровень конфиденциальности и безопасности, что вызывает доверие у пользователей.
- В качестве инструмента интеграции служит чат-бот, реализованный в социальных сетях и обеспечивающий постоянный приток пациентов.

- Каждый из акторов процесса получает определенную выгоду от использования ресурса, что обеспечивает самоподдержание его функционирования без существенных материальных вложений.

Интерфейс ресурса, доступный на стороне врача, представлен на рис. 5.16.

Рис. 5.16. Интерфейс ресурса на стороне врача

В настоящее время на платформе Университета ИТМО (Санкт-Петербург, Россия) запущена пилотная версия специализированного веб-ресурса для эффективного использования высокотехнологичной медицинской информации «Медицинская краудсорсинговая платформа Университета ИТМО» (ITMO_MCP). Ресурс функционирует на двух языках – русском и арабском.

Подход «human-in-the-loop» в автоматизированной разметке медицинских изображений. «Сердцем» предлагаемого решения является автоматизированная онлайн разметка медицинских изображений. На ресурсе реализованы три

инструмента разметки: пороговая сегментация, интерактивная ML-сегментация и неинтерактивная ML-сегментация.

Пороговая сегментация реализована средствами библиотеки OpenCV на базе алгоритма Canny, состоящего из следующих шагов:

- Исходное изображение сглаживается фильтром Гаусса 5×5.
- Градиенты ищутся путем фильтрации ядром Собеля в горизонтальном и вертикальном направлениях.
- В качестве границ отмечаются только локальные максимумы. Для этого каждый пиксель тестируется на наличие локального максимума в его окрестности в направлении градиента.
- Границы потенциального контура определяются пороговыми значениями, которые задает пользователь.

Затем выполняется морфологическое преобразование, позволяющее замкнуть контуры. Заключительный вариант контура ищется с помощью метода findContours () из OpenCV.

Последовательность преобразований исходного МРТ-изображения мозговой ткани методом пороговой сегментации представлена на рис. 5.17.

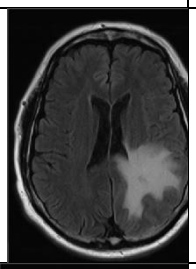
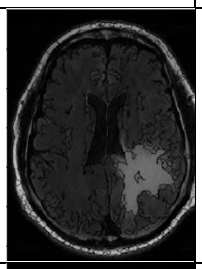
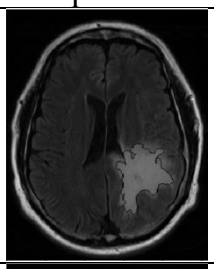
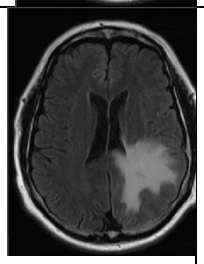
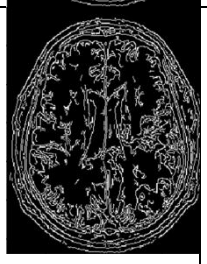
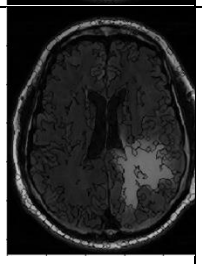
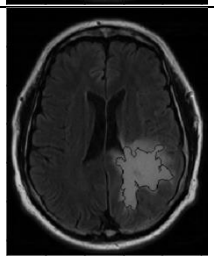
Пороговый интервал	Исходное изображение	Применение алгоритма Canny	Морфологическое преобразование	Поиск контуров	Результат селекции новообразования
t1= 27 t2= 55					
t1= 23 t2= 45					

Рис. 5.17. Трансформация исходного изображения при пороговой сегментации

Интерактивная ML-сегментация предполагает выполнение пользователем некоторых предварительных действий (предоставление сети информации о каждом сегментируемом сегменте), что сокращает время на уточнение результатов (за счет более высокого качества такой сегментации). В этом случае пользователь ставит на изображении региона интереса четыре крайние точки (верхняя, нижняя, крайняя левая, крайняя правая). Нейронная сеть выполняет семантическую сегментацию предложенной области и возвращает маску области. Процесс продолжается итеративно до тех пор, пока все нужные регионы не будут

сегментированы сетью, и на каждой итерации пользователь ставит четыре крайние точки на нужный регион. На заключительном этапе рентгенологу предлагается скорректировать результаты сегментации, выполненной нейронной сетью, с помощью стандартных инструментов рисования, предусмотренных на ресурсе.

Неинтерактивная ML-сегментация не требует предварительных дополнительных действий со стороны пользователя. Это реализовано следующим образом: пользователь выбирает необработанное изображение, затем сеть сегментирует все области интереса (поражения, бляшки), после чего пользователь при необходимости обновляет результаты сегментации стандартными инструментами рисования. В качестве архитектуры сети было выбрано достаточно легкое решение на основе FPN с магистралью RESNET34 [Lin, 2017].

Сравнение результатов разных методов показано на рис. 5.18.

а б в

Рис. 5.18. Примеры масок для выделения одной и той же зоны интереса, полученных разными методами: а – ручная разметка экспертом; б – интерактивная ML-сегментация; в – неинтерактивная ML-сегментация

Экспериментальные оценки показали, что интерактивный метод обеспечивает весьма высокое качество сегментации (средний коэффициент Дайса на валидации 0,8622), но является сравнительно медленным (среднее время обработки одного изображения 28 сек). В то же время неинтерактивный метод обеспечивает высокую скорость сегментации (среднее время инференса 5,8 мс), но более низкое качество сегментации (средний коэффициент Дайса на валидации 0,7830).

Все предложенные методы имеют свои сценарии применения. Например, в случае большого количества объектов интереса радиологу целесообразно использовать неинтерактивную сегментацию, а когда объекты интереса более очевидны и их мало, точнее и быстрее будет сегментировать их с помощью интерактивного подхода, который позволяет пользователю получить более точные результаты на этапе предварительной сегментации и тратить меньше времени на постобработку изображений.

Заметим, что ручная обработка одного изображения занимает 20–30 мин и сопровождается прогрессивным падением качества разметки из-за усталости эксперта.

Таким образом, разработанный сервис успешно реализует подход «human-in-the-loop»: ИИ формирует предварительные результаты (предлагает контуры разметки изображений), в которые пользователь итеративно вносит уточнения,

достигая тем самым быстрого и эффективного результата обработки и оценки медицинских изображений.

Вопросы для самопроверки

1. Почему объединение информации об одном и том же объекте из разных модальностей позволяет увеличить эффективность диагностики?
2. Как обрабатывается текст медицинских отчетовперед слиянием информации из разных модальностей?
3. Что показывает кросс-энтропийная функция потерь при валидации?
4. В чем преимущество перехода от попиксельной обработки МРТ-изображения к обработке супервокселей? Как генерируются супервоксели?
5. Зачем необходима генерация искусственных изображений легочных узлов на реальных компьютерных томограммах?
6. Почему для обработки МРТ-изображений легких можно использовать и 2D-GAN, и 3D-GAN?
7. В чем состоит бенчмаркинг систем синтеза медицинских изображений на основе GAN? Какие метрики можно использовать для этого?
8. В чем состоит преимущество подхода «human-in-the-loop» при обработке и оценке медицинских изображений?
9. Каков основной сценарий использования онлайн системы обработки и оценки медицинских изображений?
10. Какие инструменты разметки релизованы на ресурсе?
11. Чем отличаются интерактивная ML-сегментация и неинтерактивная ML-сегментация? В каких сценариях они применяются?

ЗАКЛЮЧЕНИЕ

Предметная область «Искусственный интеллект», с одной стороны, развивается весьма быстро, а с другой стороны, уже имеет набор апробированных решений, выступающих в качестве «строительных модулей» для создания новых систем ИИ. В настоящем пособии сделана попытка отразить обе эти тенденции. Насколько известно авторам, материал такого охвата предлагается русскоязычным читателям впервые.

Безусловно, прогресс в области ИИ во многом связан с доступностью вычислительных ресурсов. Для многих эффектных решений, таких как программа AlphaGo, победившая лучшего в мире игрока в го, требуются гигантские вычислительные мощности, и даже использование облачных технологий не всегда доступно начинающему исследователю. Учитывая это, из огромного разнообразия технологических решений в области ИИ авторы пособия старались отобрать те, которые имеют открытый код или могут быть достаточно легко реализованы «домашними средствами».

Если посмотреть на список использованных источников, то можно заметить, что большая часть технологических решений последних лет в области ИИ предложена или реализована при участии очень молодых людей, в том числе студентов и аспирантов специализированных вузов. Большинство систем ИИ, представленных в последней, прикладной части пособия, разработаны при непосредственном участии студентов и аспирантов Университета ИТМО. Учитывая репутацию, которую имеет Университет ИТМО в области компьютерных и информационных технологий, можно надеяться, что нынешние студенты – адресаты настоящего пособия – будут успешно пополнять этот список.

ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

[Agarwal, 2021] Agarwal R. et al. Neural Additive Models: Interpretable Machine Learning with Neural Nets. 35th Conference on Neural Information Processing Systems. NeurIPS. 2021.

[Ahmedt-Aristizabal, 2021] Ahmedt-Aristizabal D. et al. Graph-Based Deep Learning for Medical Diagnosis and Analysis: Past, Present and Future. arXiv:2105.13137v1 [cs.LG] 27 May 2021.

[Alammar, 2018] Alammar J. The Illustrated Transformer. June 27, 2018 https://jalammar.github.io/illustrated-transformer/ (дата обращения 26.06.2021).

[Allyn, 2020] Allyn J. International Radiology Societies Tackle Radiologist Shortage. Feb. 20, 2020. https://www.rsna.org/en/news/2020/February/International-Radiology-Societies-And-Shortage.

[Sawan, 2017] Sawan P. et al. Specialized second-opinion radiology review of PET/CT examinations for patients with diffuse large B-cell lymphoma impacts patient care and management. Medicine (Baltimore). 2017; 96:e9411.

[Amoretti, 2020] Amoretti M. C., Frixione M. Representing wine concepts: A hybrid approach. Applied Ontology -1 (2020), 1–17, 1. DOI 10.3233/AO-200239. IOS Press.

[Ancona, 2018] Ancona M. et al. Towards better understanding of gradient-based attribution methods for deep neural networks. ICLR 2018. https://arxiv.org/pdf/1711.06104.pdf (дата обращения 26.06.2021).

[Apache, 2012] Apache Software Foundation, 2012. https://www.apache.org/ (дата обращения 26.06.2021).

[Bach, 2015] Bach S. et al. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. PLOS ONE, vol. 10, no. 7, p. e0130140, Jul 2015.

[Bahdanau, 2015] Bahdanau D., Cho k., Bengio Y. Neural Machine Translation by jointly learning to align and translate. ICLR 2015. arXiv:1409.0473 [cs.CL].

[Blundell, 2015] Blundell C. et al. Weight uncertainty in neural network. Proceedings of the 32nd International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, vol. 37, 2015, pp.1613–1622.

[Bobrow, 1964] Bobrow D. G. Natural language input for a computer problem solving system (PhD). Massachusetts Institute of Technology. 1964.

[Boitsov, 2019] Boitsov V. et al. Software Tools for Manual Segmentation of Tomography Images Supporting Radiologist's Personal Context. 2019. 25th Conference of Open Innovations Association (FRUCT), 2019, pp. 64-76, doi: 10.23919/FRUCT48121.2019.8981541.

[Bramhall, 2020] Bramhall S. et al. Qlime-a quadratic local interpretable model-agnostic explanation approach. SMU Data Science Review, vol. 3, no. 1, p. 4, 2020.

[Brostow, 2009] Brostow G. J, Fauqueur J., Cipolla R. Semantic object classes in video: A highdefinition ground truth database. Pattern Recognition Letters, 30(2), 2009, pp.88–97

[Brownlee, 2019] Brownlee J. How to Implement the Frechet Inception Distance (FID) for Evaluating GANs. August 30, 2019. https://machinelearningmastery.com/how-to-

implement-the-frechet-inception-distance-fid-from-scratch (дата обращения 26.06.2021).
[Carroll, 1993] Carroll, J. B. Human Cognitive Abilities: A survey of factor-analytic studies. New York: Cambridge University Press. 1993.
[Caruana, 2015] Caruana R., et al. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 1721–1730.
[Cattell, 1963] Cattell R. Theory of fluid and crystallized intelligence: A critical experiment. Journal of Educational Psychology, 54, 1963, pp. 1-22.
[Chan, 2022] Chan et al. Explainable machine learning to predict long-term mortality in critically ill ventilated patients: a retrospective study in central Taiwan. BMC Medical Informatics and Decision Making. 2022, 22:75. https://doi.org/10.1186/s12911-022-01817-6 (дата обращения 26.06.2021).
[Chattopadhay, 2018] Chattopadhay A., et al. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2018, pp. 839–847.
[Corbetta, 2002] Corbetta, M., Shulman, G.L.: Control of goal-directed and stimulus-driven attention in the brain. Nature reviews neuroscience 3.3. 2002.
[Das, 2020] Das A., Rad P. Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey. arXiv:2006.11371v2 [cs.CV] 23 Jun 2020.
[Dean, 2004] Dean J., Ghemawat S. MapReduce: Simplified Data Processing on Large Clusters. OSDI'04: Proceedings of the 6th Conference on Symposium on Operating Systems Design & Implementation – V.6, Dec. 2004, Pages 10.
[Defferrard, 2016] Defferrard M., Bresson X., Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. Proceedings of NIPS 2016, pp. 3844-3852.
[Dempster, 1968] Dempster A.P. A generalization of Bayesian inference, Journal of the Royal Statistical Society, Series B, Vol. 30, 1968, pp. 205–247.
[Depeweg, 2018] Depeweg S., et al. Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning. Proceedings of the 35th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, vol. 80, 2018, pp. 1184–1193.
[Desai, 2020] Desai S., Ramaswamy H.G. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2020, pp. 972–980.
[Devlin, 2018] Devlin J. et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL]. 11 Oct 2018.
[Dhurandhar, 2018] Dhurandhar A. et al. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In Advances in Neural Information Processing Systems, 2018.
[Doshi-Velez, 2017] Doshi-Velez F., Kim B. Towards a rigorous science of interpretable machine learning," arXiv preprint arXiv:1702.08608, 2017.

[Dosovitskiy, 2020] Dosovitskiy A. et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.
[Elshawi, 2019] Elshawi R., et al. Interpretability in healthcare a comparative study of local machine learning interpretability techniques. Proc. IEEE Symp. Comput. Med. Syst., 2019.
[Erhan, 2010] Erhan D., Courville A., Bengio Y. Understanding representations learned in deep architectures. Department dInformatique et Recherche Operationnelle, University of Montreal, QC, Canada, Tech. Rep, vol. 1355, 2010, p. 1.
[Fikes, Nilsson, 1971] Fikes R.E., Nilsson N.J. STRIPS: A new approach to the application of theorem proving to problem solving. Artificial Intelligence. V. 2, Is. 3–4, 1971. pp. 189–208.
[Fu, 2020] Fu R., et al. Axiom-based grad-cam: Towards accurate visualization and explanation of cnns. In 31st British Machine Vision Conference, BMVC 2020.
[Gao, 2018] Gao H., Wang Z., Ji S. Large-scale learnable graph convolutional networks. Proceedings of KDD, 2018, pp. 1416-1424.
[Goan, 2020] Goan E., Fookes C. Bayesian Neural Networks: An Introduction and Survey. Cham: Springer International Publishing, 2020, pp. 45–87.
[Gonçalves, 2011] Gonçalves B., Guizzardi G., Pereira Filho J.G. Using an ECG reference ontology for semantic interoperability of ECG data. J Biomed Inform. 2011, pp. 126-136. doi: 10.1016/j.jbi.2010.08.007.
[Goodfellow, 2014] Goodfellow, I. et al. Generative adversarial nets. In Advances in neural information processing systems, 2014, pp. 2672–2680.
[Gruber, 1993] Gruber T. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal Human-Computer Studies. Vol 43, 1993, pp. 907-928.
[Grzymala-Busse, 2007] Grzymala-Busse J., Grzymala-Busse W. An experimental comparison of three rough set approaches to missing attribute values. Transactions on Rough Sets. Lecture Notes in Computer Science. Vol. 6., 2007, pp. 31–50. doi:10.1007/978-3-540-71200-8.
[Guarino, 1995] Guarino N., Giaretta P. Ontologies and Knowledge Bases. Towards Very Large Knowledge Bases, IOS Press, Amsterdam, 1995. pp. 1-2.
[Gwet, 2008] Gwet K. L. Computing inter-rater reliability and its variance in the presence of high agreement. British Journal of Mathematical and Statistical Psychology, Vol. 61, 2008, pp. 29–48.
[Hamilton, 2017] Hamilton W.L., Ying Z., Leskovec J. Inductive representation learning on large graphs. Proceedings of NIPS, 2017, pp. 1024-1034.
[He, 2016] He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE. 2016, pp. 770–778. arXiv:1512.03385. doi:10.1109/CVPR.2016.90.
[Hinton, 2006] Hinton G.E., Salakhutdinov R.R. Reducing the Dimensionality of Data with Neural Networkshttps. Science. Vol 313, Issue 5786, 2006, pp. 504–507. DOI: 10.1126/science.1127647.
[Hochreiter, 1997] Hochreiter S., Schmidhuber J. Long Short-Term Memory networks. Neural Computation 9(8), 1997, pp. 1735-80. DOI: 10.1162/neco.1997.9.8.1735.

[Hoffman, 2014] Hoffman M.D., Gelman A. The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. Journal of Machine Learning Research, vol. 15, no. 1, 2014, pp. 1593–1623.
[Holland, 2020] Holland M.A. The Black Box, Unlocked: Predictability and Understand-Ability in Military AI; United Nations Institute for Disarmament Research: Geneva, Switzerland, 2020.
[Holzinger, 2020] Holzinger A., Carrington A., Muller H. Measuring the Quality of Explanations: The System Causability Scale (SCS). KI - Kunstliche Intelligenz, 2020, pp. 193-198.
[Hopfield, 1982] Hopfield J. J. Neural networks and physical systems with emergent collective computational abilities. Proceedings of National Academy of Sciences, vol. 79 no. 8, 1982, pp. 2554–2558.
[Jessell, 2000] Jessell T.M., Kandel E.R., Schwartz J.H. Central visual pathways. Principles of neural science. New York: McGraw-Hill, 2000, pp. 533–540. ISBN 978-0-8385-7701-1.
[Jospin, 2022] Jospin L.V. et al. Hands-on Bayesian Neural Networks – A Tutorial for Deep Learning Users. arXiv:2007.06823v3 [cs.LG] 3 Jan 2022.
[Jung, 2021] Jung H., Oh Y. Towards Better Explanations of Class Activation Mapping. ICCV, 2021. https://arxiv.org/abs/2102.05228 (дата обращения 26.06.2021).
[Karras, 2019] Karras T., Laine S., Aila T. A Style-Based Generator Architecture for Generative Adver-sarial Networks. arXiv:1812.04948v3 [cs.NE] 2019. https://github.com/NVlabs/stylegan (дата обращения 26.06.2021).
[Kendall, 2017] Kendall A., Gal Y. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA. arXiv:1703.04977v2 [cs.CV], 2017.
[Kim, 2018] Kim B., et al. Interpretability beyond feature attribution: Quantita-tive Testing with Concept Activation Vectors (TCAV). 35th International Con-ference on Machine Learning, ICML 2018, 2018.
[Kiely, 2014] Kiely K.M. Cognitive Function. In: Michalos A.C. (eds) Encyclopedia of Quality of Life and Well-Being Research. Springer, Dordrecht, 2014, P. 426. https://doi.org/10.1007/978-94-007-0753-5.
[Kingma, 2014] Kingma D., Welling M. Auto-Encoding Variational Bayes. 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, 2014.
[Kipf, 2017] Kipf T.N., Welling M. Semi-supervised classification with graph convolutional networks. Proceedings of ICLR, 2017.
[Kozyreva, 2021] Kozyreva A. et al. Integration of Social Media Platforms and Specialized Web Resources for the Effective Use of High-tech Medical Information // Proceedings of the 7th International Conference on Information and Communication Technologies for Ageing Well and e-Health, 2021, pp. 154-162.
[Krizhevsky, 2012] Krizhevsky A., Sutskever I., Hinton G.E. ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems 25, 2012.

[Lecun, 1998] Lecun Y., Bottou L., Bengio Y. Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE, vol. 86, no. 11, 1998, pp. 2278–2324. doi: 10.1109/5.726791.

[Letham, 2015] Letham B., et al. Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. The Annals of Applied Statistics, vol. 9, no. 3,2015, pp. 1350–1371.

[Lezak, 1983] Lezak M. D. Neuropsychological Assessment (2nd ed.). New York Oxford University Press. 1983, p.768.

[Li, 2002] Li Z. A saliency map in primary visual cortex. Trends in Cognitive Sciences. 6 (1), 2002, pp. 9–16. doi:10.1016/s1364-6613(00)01817-9.

[Li, 2020] Li Y., Fan Y., DeepSEED: 3D squeeze-and-excitation encoder-decoder convnets for pulmonary nodule detection. 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI). IEEE, 2020.

[Li, 2021] Li B. et al. Trustworthy AI: From Principles to Practices. arXiv:2110.01167v1 [cs.AI] 4 Oct 2021.

[Li, 2021] Li X. et al. Interpretable Deep Learning: Interpretation, Interpretability, Trustworthiness, and Beyond. arXiv:2103.10689 [cs.LG]. 11 May 2021.

[Lin, 2017] Lin T.-Y. et al. Feature pyramid networks for object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.

[Linardatos, 2021] Linardatos P., Papastefanopoulos V., Kotsiantis S. Explainable AI: A Review of Machine Learning Interpretability Methods. Entropy 2021, 23(1), 18; https://doi.org/10.3390/e23010018.

[Linnainmaa, 1970] Linnainmaa S. Algoritmin kumulatiivinen pyoristysvirhe yksittaisten pyoristysvirheiden taylor-kehitelmana [The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors] (PDF) (Thesis) (in Finnish), 1970, pp. 6–7.

[Lobantsev, 2020] Lobantsev A. et al. Creation of a Publicly Accessible Resource for Increasing the Volume of Freely Distributed Medical Datasets. Proceedings of the 13th IADIS International Conference ICT, Society and Human Beings 2020, ICT 2020 and Proceedings of the 6th IADIS International Conference Connected Smart Cities 2020, CSC 2020 and Proceedings of the 17th IADIS International Conference Web Based Communities and Social Media 2020, WBC 2020 - Part of the 14th Multi Conference on Computer Science and Information Systems, MCCSIS 2020, 2020, pp. 19-26.

[Lundberg, 2017] Lundberg S.M. and Lee S.I. A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 2017, pp. 4765–4774.

[Lundberg, 2017] Lundberg S.M., Su-In Lee. A unified approach to interpreting model predictions. In Advances in neural information processing systems, 2017, pp. 4765–4774.

[Lundberg, 2019] Lundberg S.M. et al. Explainable AI for Trees: From Local Explanations to Global Understanding. arXiv:1905.04610 [cs.LG]. 11 May 2019.

[Luss, 2019] Luss R et al. Generating contrastive explanations with monotonic attribute functions. arXiv preprint arXiv:1905.12698, 2019.

[Zeiler, 2014] Zeiler M.D., Fergus R. Visualizing and understanding convolutional networks. European conference on computer vision, 2014, pp. 818–833.
[Melis, 2018] Melis D.A., Jaakkola T. Towards robust interpretability with self-explaining neural networks. Advances in Neural Information Processing Systems, 2018, pp. 7775–7784.
[Miller, 2019] Miller T. Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence, V. 267, 2019, pp. 1–38.
[Minski, 1975] Minsky, M. A framework for representing knowledge. In P. H. Winston (Ed.), The psychology of computer vision. New York: McGraw-Hill Book. 1975.
[Mirsky, 2019] Mirsky Y. et al. CT-GAN: Malicious tampering of 3D medical imagery using deep learning. 28th USENIX Security Symposium (USENIX Security 19), 2019, pp. 461-478. https://github.com/ymirsky/CT-GAN (дата обращения 26.06.2021).
[Mishra, 2017] Mishra S., Sturm B.L., Dixon S. Local interpretable modelagnostic explanations for music content analysis. Proc. 18th Int. Soc. Music Inf. Retr. Conf. ISMIR 2017, 2017, pp. 537–543.
[Molnar, 2018] Molnar C., Casalicchio G., Bischl B. iml: An r package for interpretable machine learning. Journal of Open-Source Software, vol. 3, no. 26, 2018, p. 786.
[Onishi, 2019] Onishi Y. et al. Automated Pulmonary Nodule Classification in Computed Tomography Images Using a Deep Convolutional Neural Network Trained by Generative Adversarial Networks. Hindawi BioMed Research International. Vol. 2019, https://doi.org/10.1155/2019/6051939.
[Pawlak, 1982] Pawlak Z. Rough set [J], Int. J. Comput. Inf. Sci. 11 (5), 1982.
[Pchelkin, 2021] Pchelkin A., Gusarova N., Dobrenko N., Vatyan A. Mobile Healthcare Service for Self-organization in Older Populations During a Pandemic // Communications in Computer and Information Science, Vol. 1395, 2021, pp. 379-390.
[Petsiuk, 2018] Petsiuk V., Das A., Saenko K. RISE: Randomized Input Sampling for Explanation of Black-box Models. British Machine Vision Conference 2018, BMVC 2018, 2018.
[Ribeiro, 2016] Ribeiro M.T., Singh S., Guestrin C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. arXiv:1602.04938 [cs.LG]. 2016.
[Ribeiro, 2016] Ribeiro M.T., Singh S., Guestrin C. «Why Should I Trust You?». Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16. New York, New York, USA: ACM Press, 2016, pp. 1135–1144.
[Romir, 2020] Международная аналитика Romir/GlobalNR: граждане оценили общее состояние национальных систем здравоохранения. 13 May 2020. Международная аналитика Romir/GlobalNR: граждане оценили общее состояние национальных систем здравоохранения (дата обращения 26.06.2021).
[Ronneberger, 2015] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv:1505.04597. 2015.
[Rosenblatt, 1961] Rosenblatt F., Principles of Neurodynamics. Perceptrons and the Theory of Brain Mechanisms. Cornell Aeronautical LAB INC Buffalo N Y. Defense Technical Information Center, 1961.

[Rumelhart, 1986] Rumelhart, D., Hinton, G. & Williams, R. Learning representations by back-propagating errors. Nature 323, 1986, pp. 533–536. https://doi.org/10.1038/323533a0.
[Russell, Norvig, 2020] Russell S., Norvig P. Artificial Intelligence: A Modern Approach, Pearson, 4th Edition, 2020. ISBN-13: 9780134671864.
[Salimans, 2016] Salimans I. et al. Improved techniques for training GANs. In Advances in Neural Information Processing Systems (NIPS), 2016, pp. 2234–2242.
[Salthouse, 2005] Salthouse T. A. Relations between Cognitive Abilities and Measures of Executive Functioning. Neuropsychology, V. 19, 2005, p. 532–545.
[Scarselli, 2008] Scarselli F., et al. The graph neural network model. IEEE Trans. Neural Netw., vol. 20, no. 1, 2008, pp. 61–80.
[Schmidt, 2019] Schmidt P., Biessmann F. Quantifying Interpretability and Trust in Machine Learning Systems. In Proceedings of the AAAI-19 Workshop on Network Interpretability for Deep Learning, Honolulu, HI, USA, 27 January–2 February 2019, p. 8
[Schwartz, 2021] Schwartz R et al. A Proposal for Identifying and Managing Bias in Artificial Intelligence. Draft. National Institute of Standards and Technology. Special Publication 1270. June 2021.
[Selvaraju, 2017] Selvaraju R.R., et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. Proceedings of the IEEE international conference on computer vision, 2017, pp. 618–626.
[Semiletov, 2021] Semiletov A. et al. Comparative Evaluation of Lung Cancer CT Image Synthesis with Generative Adversarial Networks // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) - Vol. 12744, 2012, pp. 593-608.
[Shafer, 1976] Shafer G. A Mathematical Theory of Evidence, Princeton University Press, 1976.
[Shamshad, 2022] Shamshad F. Transformers in Medical Imaging: A Survey. arXiv:2201.09873v1 [eess.IV] 24 Jan 2022.
[Sharkawy, 2020] Sharkawy A.-N. Principle of Neural Network and Its Main Types: Review. Journal of Advances in Applied & Computational Mathematics, 7, 2020, pp. 8-19.
[Shi, 2020] Shi S., Zhang X., Fan W. A modified perturbed sampling method for local interpretable model-agnostic explanation. arXiv preprint arXiv:2002.07434, 2020.
[Shortliffe, Buchanan,1975] Shortliffe E.H., Buchanan B.G. A model of inexact reasoning in medicine". Mathematical Biosciences. 23 (3–4), 1975, pp. 351–379. doi:10.1016/0025-5564(75)90047-4.
[Simonyan, 2013] Simonyan K., Vedaldi A., Zisserman A. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. 2nd International Conference on Learning Representations, ICLR 2014 - Workshop Track Proceedings, Dec 2013.
[Simonyan, 2015] Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition. ICLR 2015. arXiv:1409.1556.
[Strakhov, 2018] Strakhov M. Generative adversarial networks. 11 Apr. 2018. https://habr.com/ru/post/352794/ (дата обращения 26.06.2021).

[Sutton, 2015] Sutton R.S., Barto A.G. Reinforcement Learning: An Introduction. Second edition, in progress. 2015. A Bradford Book. The MIT Press. 352 p.
[Szegedy, 2014] Szegedy C. Going Deeper with Convolutions. arXiv:1409.4842 [cs.CV] 17 Sep 2014.
[Teney, 2017] Teney D., Liu L., Hengel A.V. D. Graph-structured representations for visual question answering. Proceedings of CVPR (2017), pp. 3233-3241.
[Tunali, 2020] Tunali O. Maximum mean discrepancy. V.7. Дата обращения 23.05.2022. https://www.kaggle.com/code/onurtunali/maximum-mean-discrepancy/notebook (дата обращения 26.06.2021).
[van der Maaten, 2008] van der Maaten L.J.P., Hinton G.E. Visualizing Data Using t-SNE. Journal of Machine Learning Research. 2008. Ноябрь (т. 9).
[Vaswani, 2017] Vaswani A. Attention is all you need. NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing SystemsDecember 2017 Pages 6000–6010.
[Vaswani, 2017] Vaswani A. et al. Attention Is All You Need. 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.
[Vatian, 2019] Vatian A., Gusarova N., Dobrenko N. et al. Fusing of Medical Images and Reports in Diagnostics of Brain Diseases // ACM International Conference Proceeding Series - 2019, pp. 102-108.
[Velickovic, 2018] Velickovic et al., Graph attention networks. Proceedings of ICLR, 2018.
[Wang, 2020] Wang H., et al. Score-cam: Score-weighted visual explanations for convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 24–25.
[Winograd, 1968] Winograd T. Procedures as a Representation for Data in a Computer Program for Understanding Natural Language. MIT AI Technical Report 235. 1971.
[Wooldridge, 2002] Wooldridge M. An Introduction to MultiAgent Systems. John Wiley & Sons. 2002, p. 366. ISBN 978-0-471-49691-5.
[Xu, 2015] Xu K. et al. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. Proceedings of the 32 nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP vol. 37
[Hubel, 1959] Hubel D. H., Wiesel T. N. Receptive fields of single neurones in the cat's striate cortex. J. Physiol. 148(3), 1959, pp. 574-91. doi: 10.1113/jphysiol.1959.sp006308.
[Xu, 2019] Xu et al. Graph wavelet neural network. Proceedings of ICLR, 2019.
[Yan, 2019] Yan Z. et al. Brain Tissue Segmentation based on Graph Convolutional Networks. 2019 IEEE International Conference on Image Processing (ICIP), 2019, pp. 1470-1474, doi: 10.1109/ICIP.2019.8803033.
[Yang, 2017] Yang H., Rudin C., Seltzer M. Scalable Bayesian rule lists. 34th International Conference on Machine Learning, ICML 2017, 2017
[Yang, 2019] Yang M., Kim B. Benchmarking Attribution Methods with Relative Feature Importance. CoRR, vol. abs/1907.09701, 2019.

[Yang, 2022] Yang G., Ye Q., Xia J. Unbox the black-box for the medical explainable AI via multi-modal and multi-centre data fusion: A mini-review, two showcases and beyond. Information Fusion 77, 2022, pp. 29–52.
[Zadeh, 1965] Zadeh L.A. Fuzzy sets. Information and Control. San Diego. 1965, pp. 338–353. doi:10.1016/S0019-9958(65)90241-X.
[Zafar, 2019] Zafar M. R., Khan N.M. Dlime: A deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems. arXiv preprint arXiv:1906.10263, 2019.
[Zeiler, 2014] Zeiler M.D., Fergus R. Visualizing and understanding convolutional networks. Computer Vision – ECCV 2014, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 818–833.
[Zhang, 2022] Zhang Y., Weng Y., Lund J. Applications of Explainable Artificial Intelligence in Diagnosis and Surgery. Diagnostics 2022, 12, 237. https://doi.org/10.3390/ diagnostics12020237.
[Zhaoping, 2019] Zhaoping L. A new framework for understanding vision from the perspective of the primary visual cortex. Current Opinion in Neurobiology. 58: 2019, pp. 1–10. doi:10.1016/j.conb.2019.06.001.
[Zheng, 2017] Zheng et al. Hybrid-augmented intelligence: collaboration and cognition. Front Inform Technol Electron Eng 18(2), 2017, pp. 153-179.
[Zheng, 2017] Zheng et al. Hybrid-augmented intelligence: collaboration and cognition. Front Inform Technol Electron Eng 18(2), 2017, pp. 153-179.
[Zhou, 2016] Zhou B., et al. Learning Deep Features for Discriminative Localization. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Jun 2016, pp. 2921–2929.
[Zhou, 2019] Zhou J. et al. Physiological Indicators for User Trust in Machine Learning with Influence Enhanced Fact-Checking. In International Cross-Domain Conference for Machine Learning and Knowledge Extraction; Holzinger A., Kieseberg P., Tjoa A.M., Weippl E., Eds.; Springer: Cham, Switzerland, 2019, pp. 94–113.
[Zhou, 2020] Zhou J. et al. Graph neural networks: A review of methods and applications. AI Open, Volume 1, 2020, pp. 57-81.
[Zhou, 2021] Zhou J. et al. A. Evaluating the Quality of Machine Learning Explanations: A Survey on Methods and Metrics. Electronics 2021, 10, 593.
[Zhu, 2019] D. Zhu et al. Robust graph convolutional networks against adversarial attacks. Proceedings of KDD 2019, pp. 1399-1407.
[Введение, 2007] Введение в математическое моделирование: Учеб. пособие / Под ред. П.В. Трусова. М.: Университетская книга, Логос, 2007. 440 с. ISBN 978-5-98704-037-X.
[Гаврилова, 2000] Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. СПб.: Питер, 2000. 384 с.
[ГОСТ 33707–2016] ГОСТ 33707–2016 (ISO/IEC 2382:2015) Информационные технологии (ИТ). Словарь. Введен 22 сентября 2016 г.
[ГОСТ Р 57309–2016] ГОСТ Р 57309–2016 (ИСО 16354:2013) Руководящие принципы по библиотекам знаний и библиотекам объектов. Введен 02 декабря 2016 г.

[ГОСТ Р 59276–2020] ГОСТ Р 59276–2020 Системы искусственного интеллекта. Способы обеспечения доверия. Общие положения. Дата введения 23 декабря 2020 г.
[ГОСТ Р 59277–2020] ГОСТ Р 59277-2020 Системы искусственного интеллекта. Классификация систем искусственного интеллекта. Дата введения 23 декабря 2020.
[ГОСТ Р ИСО 9000–2015] ГОСТ Р ИСО 9000–2015. Системы менеджмента качества. Основные положения и словарь. Дата введения 2015-11-01.
[ГОСТ Р ИСО/МЭК 9126-93] ГОСТ Р ИСО/МЭК 9126-93. Оценка программной продукции. Характеристики качества и руководства по их применению. Дата введения 1994-07-01.
[Дале, 2021] Дале Д. Маленький и быстрый BERT для русского языка. https://habr.com/ru/post/562064/ 10 июня 2021.
[Добров, 2020] Добров Б.В. и др.Онтологии и тезаурусы: модели, инструменты, приложения. М.: ИНТУИТ, Ай Пи Ар Медиа, 2020. 172 с. ISBN 978-5-4497-0668-3.
[Ивахненко, 1962] Ивахненко А.И. Техническая кибернетика. Системы автоматического управления с приспособлением характеристик. Государственное издательство технической литературы УССР, 1962. 422 с.
[Козлов, 2018] Козлов С. Курс о Deep Learning на пальцах. https://habr.com/ru/post/414165/ (дата обращения 26.06.2021).
[Косолапов, 2018] Косолапов К. Нейронные сети, фундаментальные принципы работы, многообразие и топология. 2018. https://habr.com/ru/post/416071/ (дата обращения 26.06.2021).
[Лурия, 1973] Лурия А.Р. Основы нейропсихологии. М., изд. МГУ, 1973.
[Наппельбаум, Поспелов, 1981] Наппельбаум Э. Л., Поспелов Д. А. Субъективное структурирование информации в задачах коллективного принятия решений// Нормативные и дескриптивные модели принятия решений / Под ред. Б. Ф. Ломова, В. Ю. Крылова, Н. В. Крыловой и др. М.: Наука, 1981. С. 191–205.
[Ной, 2007] Ной Н.Ф., МакГиннесс Д.Л. Разработка онтологий 101: руководство по созданию Вашей первой онтологии. http://www.labrate.ru/20181225/razrabotka_ontologiy_101_ruk.pdf (дата обращения 26.06.2021).
[Осинга, 2019] Осинга Д. Глубокое обучение. Готовые решения. Диалектика, 2019. 288 с. ISBN: 978-5-907144-50-7.
[Поспелов, 1969] Поспелов Д.А. «Сознание», «самосознание» и вычислительные машины. http://raai.org/about/persons/pospelov/pages/SR1_Pospelov_1969.pdf
[Поспелов, 1986] Поспелов Д.А. Ситуационное управление. Теория и практика. М.: Наука, 1986. 288 с.
[Программа, 2020] Перспективная программа стандартизации по приоритетному направлению «Искусственный интеллект» на период 2021–2024 годы https://www.economy.gov.ru/material/file/28a4b183b4aee34051e85ddb3da876225/20201222.pdf (дата обращения 26.06.2021).
[Стратегия, 2019] Указ Президента РФ от 10 октября 2019 г. N 490 "О развитии искусственного интеллекта в Российской Федерации" https://base.garant.ru/72838946/ (дата обращения 26.06.2021).

[Технический, 2022] Технический комитет по стандартизации (ТК 164) «Искусственный интеллект». Окончательные редакции. Дата обращения 21.05.2022. https://www.tc164.ru/окончательные редакции (дата обращения 26.06.2021).

Александра Сергеевна Ватьян
Наталия Федоровна Гусарова
Наталья Викторовна Добренко

СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Научно-учебное издание

Редакционно-издательский отдел Университета ИТМО
Зав. РИО Н.Ф. Гусарова
Дизайн обложки Н.А. Потехина
Вёрстка Н.Ф. Гусарова
Подписано к печати 29.09.2022
Заказ № 4678
Тираж 200

Печатается в авторской редакции

Отпечатано: Учреждение «Университетские коммуникации»
199034, Санкт-Петербург, В.О., Биржевая линия, 16