Содержание

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

ВС Вычислительная система

ГКС Глобальная компьютерная система

ИБ Информационная безопасность

ИИО Индекс информационного общества

ИПК Индекс прозрачности коммуникаций

ИТО Индекс технологической оснащенности

ИР Информационные ресурсы

МВС Многопроцессорные вычислительные системы

МЗ Механизм защиты

НВ Нечеткое высказывание

НК Нейрокомпьютер

НЛ Непрерывная логика

НМ Нечеткое множество

НП Непрерывная переменная

НС Нейронная сеть

НСД Несанкционированный доступ к информации

НЧС Нечеткая связь

ПО Программное обеспечение

СД Семантическое данное

СЗИ Средства защиты информации

СИБ Система информационной безопасности

ФН Формальный нейрон

ХС Хозяйствующий субъект

УПД Управление потоком данных

ВВЕДЕНИЕ

Эволюция средств обработки информации осуществляется в направлении создания систем информационных технологий (ИТ) *с* элементами самоорганизации, в которых присутствуют процессы зарождения, приспособления и развития [1]. На названных процессах основаны биологические системы, для которых характерны опыт эволюции, селективный отбор. Заимствование архитектурных принципов биосистем привело к разработке теорий нейронных сетей (НС), нечетких множеств, эволюционных методов, лежащих в основе *искусственных интеллектуальных систем*.

Для реализации названных процессов в технических системах совершенствуются методы нечетких вычислений, которые основываются на знаниях экспертов и хорошо зарекомендовали себя в условиях *неполной достоверности* и *неопределенности* информации. Задачи оптимизации решаются эволюционными методами, в том числе, с привлечением генетических алгоритмов. Нейросетевые технологии предоставляют *адаптивные* средства для реализации систем ИТ.

Эволюционный алгоритм можно рассматривать как итеративный алгоритм, который поддерживает популяцию индивидуумов. Первоначальная популяция создается в результате некоторого эвристического процесса. Новая популяция формируется с помощью отбора лучших индивидуумов путем отсеивания некоторых членов популяции в процессе эволюции. Каждый индивидуум - потенциальное решение задачи. При отборе решений используется критерий качества. После генерации ряда популяций можно получить индивидуум, наиболее полно соответствующий критерию качества. Эволюционные алгоритмы следуют принципу: популяция индивидуумов претерпевает преобразования, в процессе которых индивидуумы повышают свою выживаемость.

Нейронные сети получили распространение в многочисленных прикладных сферах распределенных вычислениях при решении нечетких и трудно формализуемых задач. Внимание разработчиков ИТ к НС можно объяснить естественным параллелизмом НС в противовес последовательному характеру управления ходом вычислений, свойственных большинству известных систем ИТ. Немаловажными факторами, способствующими распространению нейросетевых вычислений, являются такие свойства НС, как адаптивность, высокие информационная защищенность, способность выделения и классификации скрытых в информации знаний. Данный перечень качеств в большей мере присущ биосистемам, к которым НС существенно ближе, чем к современным системам ИТ.

Как известно [2], биосистемы обладают многоуровневой иерархической системой жизнеобеспечения, реализованной с использованием комплекса меха-

низмов информационной избыточности, защиты и иммунитета. Механизмы обеспечения информационной безопасности современных ИТ по возможностям далеки от биологических прототипов, в связи с чем разработка подхода к созданию адаптивных систем ИТ с встроенными функциями жизнеобеспечения, основанных на биосистемной аналогии, представляется *актуальной*.

Искусственным НС присуще свойство биологического подобия, как техническим моделям реальных биологических НС [3]. Нейросетевой базис можно рассматривать как основу для создания адаптивных командных пулов – аналога биологической ткани, в которых программно формируется иерархия функциональных устройств (комплекс взаимосвязанных органов) в соответствии с требованиями спецификации на разработку прикладной системы [4]. Механизмы информационной безопасности внутренне присущи, и адаптивным командным пулам, и функциональным компонентам системы ИТ, повторяя механизмы иммунной защиты организма.

НС свойственно *нечеткое представление данных*. Возможно представление данных в виде некоторой окрестности, нахождение значений в которой не вызывает изменения реализуемой НС функции. Информация в виде системы взвешенных межнейронных связей представляется в избыточной распределенной по НС форме, а искажение (снижение истинности) как оперативных, так и долговременных (системных) данных не приводит к утрате работоспособности НС. В процессах работы и адаптации НС участвует не локальная связь, а вся система межнейронных связей в форме *нечеткого избыточного распределенного информационного поля НС*.

Одним из перспективных направлений развития безопасных систем ИТ можно считать создание *адаптивных* СЗИ, удобных для технической реализации с привлечением современных наноэлектронных технологий [5] в виде СБИС, кремниевых пластин, ориентированных на высоконадежные механизмы жизнеобеспечения и информационной защиты биологических систем.

Высокая производительность систем ИТ при решении задач, характеризующихся нечеткой, недостоверной информацией, нерегулярными процессами обработки с изменяющимися в процессе эксплуатации системы составом и взаимосвязями компонентов, может обеспечиваться параллелизмом нейросетевых вычислений и управлением потоком данных (УПД). Подобные вычисления необходимы в задачах управления и обеспечения информационной безопасности сложных комплексов на основе адаптивных систем ИТ с защищенными процессами обработки и хранения больших объемов конфиденциальной информации.

Однако известные методы оказываются малопригодными для решения нечетких неформализуемых задач, где применимы *нечеткие вычисления* и *нейро-*

сетевые средства. Существующие методы распределенных вычислений, архитектуры и программное обеспечение систем ИТ не ориентированы на решение задач обеспечения информационной безопасности сложных технических комплексов в динамично изменяющихся условиях эксплуатации, не учитывают специфику нечетких и нейросетевых вычислений. Не разработаны методы и модели адаптивных СЗИ для построения информационно безопасных систем ИТ, способных приспосабливаться к изменению поля угроз.

Необходима разработка моделей систем ИТ с встроенными функциями информационной безопасности на основе биосистемной аналогии. Необходима разработка архитектуры и механизмов обеспечения информационной защиты иерархических технических комплексов, позволяющих в полной мере реализовать комплекс механизмов жизнеобеспечения и информационной защиты, присущий биологическим системам.

В учебном пособии рассмотрен подход к разработке модели адаптивной защиты, реализуемой на основе биосистемной аналогии с использованием интеллектуальных механизмов нейронных сетей и нечеткой логики. Целью настоящей работы является разработка модели и методики построения адаптивной системы информационной безопасности (СИБ), использующих адаптивные наборы (матрицы) экспертных оценок для информационно безопасных систем ИТ, ориентированных на нейросетевые вычисления, модели, учитывающей изменение поля угроз на этапах жизненного цикла системы ИТ.

Основными объектами исследований являются системы защиты информации, а предметом исследований – модели и методы построения адаптивных нейросетевых систем защиты информации с распределенной архитектурой, формами параллелизма, нечетким распределенным представлением информации.

Основными вопросами, рассматриваемыми в настоящем пособии, являются:

Разработка модели адаптивной информационной защиты систем ИТ на основе нейро-нечетких средств защиты информации, используя аналогию с защитными механизмами биологических систем.

Разработка системы оценок информационной защищенности систем ИТ, учитывающей структурные и экономические показатели адаптивной системы защиты информации.

Разработка методики построения адаптивной системы защиты информации на основе предложенных оценок и адаптивной модели СИБ.

Разработка архитектурных решений информационно защищенных командных пулов, учитывающих детализацию описания НС.

Разработка инструментальных средств для поддержки методики построения адаптивной СИБ.

Для изложения материала пособия использованы методы теории информационной безопасности систем, нейронных сетей, нечетких множеств, а также моделирование и исследование нейросетевых систем защиты информации.

1. ИНТЕЛЛЕКТУАЛЬНЫЕ СРЕДСТВА И МОДЕЛИРОВАНИЕ СИСТЕМ ЗАЩИТЫ ИНФОРМАЦИИ

В научных и научно-технических изданиях [6] активно обсуждается необходимость придания системам защиты информации в ИТ эволюционных качеств, присущих биосистемам, таких как *возможность развития и адаптивность*. Известные фирмы, например, Microsoft, заявляют о применении «технологии активной защиты» [7], основанной на *оценке поведения* программ с точки зрения их потенциальной опасности. В частности, СЗИ *корректируют* средства защиты компьютера при изменении его статуса или блокируют его, если возникает подозрение в заражении вирусом или проникновении злоумышленника [8].

1.1. Анализ использования интеллектуальных средств в системах защиты информации

Актуальна проблема *эволюционного развития* систем информационной безопасности (СИБ). Наряду с традиционными средствами защиты корпоративных сетей, такими как: антивирусы, детекторы уязвимостей, межсетевые экраны и детекторы вторжений используются средства автоматизации защиты, включающие корреляторы событий, программы обновлений, средства аутентификации, авторизации и администрирования (authentication, authorization, administration — ЗА) и системы управления рисками [9]. Корреляторы событий предназначены для анализа системных журналов СЗИ, операционных систем и приложений для выявления признаков нападения; программы обновления - автоматизации процедур установки исправлений для устранения выявленных уязвимостей (прежде всего, ошибок ПО) и поиска потенциальных уязвимостей системы; средства ЗА - управления идентификационной информацией и допуском пользователей к информационным ресурсам, а система управления рисками - моделирования и определения возможного ущерба от атаки на корпоративную сеть.

1.1.1. Интеллектуальные средства и задачи защиты информации

В основном публикации о применении интеллектуальных систем защиты информации посвящены системам обнаружения атак [10-19], в качестве интеллектуального инструмента в которых, как правило, используются нейронные сети (НС), системы нечеткой логики (НЛ) и экспертные системы (ЭС) [20-25].

Схемы обнаружения атак разделяют на две категории: 1) обнаружение злоупотреблений и 2) обнаружение аномалий. К первым относят атаки, которые используют известные уязвимости системы ИТ, а ко вторым - несвойственную пользователям системы ИТ деятельность. Для *обнаружения аномалий* выявляется деятельность, которая отличается от шаблонов, установленных для пользователей или групп пользователей. Обнаружение аномалий, как правило, связано с созданием базы данных, которая содержит профили контролируемой деятельности [26-28], а *обнаружение злоупотреблений* – со сравнением деятельности пользователя с известными шаблонами поведения хакера [29, 30] и использует методы на основе правил, описывающих сценарии атак. Механизм обнаружения идентифицирует потенциальные атаки в случае, если действия пользователя не совпадают с установленными правилами.

Большинство систем обнаружения злоупотреблений и аномалий основаны на модели, предложенной Деннингом [31]. Модель поддерживает набор профилей для легальных пользователей, согласовывает записи подсистемы аудита с соответствующим профилем, обновляет профиль и сообщает о любых обнаруженных аномалиях.

Для определения аномального поведения часто используют статистические методы для сравнения используемых пользователем команд с нормальным режимом работы. Поведение пользователя может быть представлено как модель на основе правил [32], в терминах прогнозируемых шаблонов [33] или анализа изменения состояния [22], а для выявления факта атаки используют методы сопоставления с образцом.

Можно выделить следующие варианты применения НС в системах обнаружения атак. *Дополнение нейронной сетью* существующих экспертных систем для фильтрации поступающих сообщений с целью снижения числа ложных срабатываний, присущих экспертной системе. Так как экспертная система получает данные только о событиях, которые рассматриваются в качестве подозрительных, чувствительность системы возрастает. Если НС за счет обучения стала идентифицировать новые атаки, то экспертную систему также следует обновить. Иначе новые атаки будут игнорироваться экспертной системой, прежние правила которой не способны распознавать данную угрозу.

Если НС представляет собой отдельную систему обнаружения атак, то она обрабатывает трафик и анализирует информацию на наличие в нем злоупотреблений. Любые случаи, которые идентифицируются с указанием на атаку, перенаправляются к администратору безопасности или используются системой автоматического реагирования на атаки. Этот подход обладает преимуществом в скорости по сравнению с предыдущим подходом, т. к. существует только один уровень анализа, а сама система обладает свойством адаптивности. НС приме-

няют также в системах криптографической защиты информации для хранения криптографических ключей в распределенных сетях [34].

Основным недостатком НС считают «непрозрачность» формирования результатов анализа [35]. Однако использование гибридных нейро-экспертных или нейро-нечетких систем позволяет явным образом отразить в структуре НС систему нечетких предикатных правил, которые автоматически корректируются в процессе обучения НС [36]. Свойство адаптивности нечетких НС позволяет решать не только отдельно взятые задачи идентификации угроз, сопоставления поведения пользователей с имеющимися в системе шаблонами, но и автоматически формировать новые правила при изменении поля угроз, а также реализовать систему защиты информации технической системы в целом.

1.1.2. Интеллектуальные средства для моделирования систем защиты информации

Для обнаружения и противодействия несанкционированным действиям используют различные математические методы и интеллектуальный инструментарий, как в нашей стране [37-67], так и за рубежом [68-74].

В [37] описан математический аппарат скрытых Марковских цепей для контроля принадлежности потоков к процессу, исполняемому в системе ИТ, и выявления несанкционированных процессов, а в [38] - задача идентификации вычислительных сетей (ВС) по набору доступных для наблюдения параметров и отнесение ВС к одному из известных классов.

В ряде работ [45 - 49, 72 - 74] рассматривается использование интеллектуальных мультиагентных систем для защиты информации. В частности дается обзор инструментов реализации атак, онтология предметной области, определяются структура команды агентов СЗИ, механизмы их взаимодействия и координации. Другая группа работ [50 - 52] посвящена проблеме применения мультиагентных и интеллектуальных технологий для обнаружения вторжений на Web-сервер, тестирования защищенности и обучения систем ИТ. Предложены подходы к построению систем моделирования атак на Web-сервер, основанные на использовании онтологии сетевых атак, стратегий их реализации, а также применении хранилища уязвимостей и программ реализации атак.

Работы [53, 54] посвящены обсуждению специфики применения НС для целей идентификации динамических объектов исходя из математического описания многослойной НС и мониторинга информационных систем.

Применение аппарата НС и генетических алгоритмов для защиты сетей от программных атак, направленных на нарушение доступности ресурсов, рассмотрена в [55 - 58]. Причем НС используются для обнаружения признаков атак в сетевом трафике, идентификации форматов передаваемых данных, динамической идентификации участников обмена, а генетические алгоритмы - получе-

ния близкого к оптимальному решения в задачах управления маршрутами и параметрами трафика при наличие нечеткости данных идентификации атаки в условиях дефицита информации или информационного «шума».

В [59, 60] использован аппарат нечетких множеств для реализации активного аудита безопасности работы системы ИТ. Для оценки защищенности сетей от угроз НСД, обнаружения злоупотреблений пользователей и программных атак применены методы интеллектуального анализа данных, работающие по принципу адаптивной защиты от НСД - «анализ – прогнозирование – предупреждение».

Идентификации и аутентификации пользователя по биометрическим, фонетическим параметрам посвящены ряд исследований [61 - 67], использующих математический аппарат нейронных сетей, комбинированные методы быстрой цифровой обработки сигналов и НС.

Из проведенного анализа следует вывод о необходимости решения не отдельных задач защиты информации с помощью НС, систем нечеткой логики, экспертных систем, а разработки *единого подхода* применения интеллектуальных средств для создания комплексной адаптивной защиты систем ИТ на основе биоаналогии [3]. Проектирование следует осуществлять как единый процесс построения адаптивной системы ИТ с внутренне присущими функциями защиты информации [75].

Наилучшим сочетанием свойств для достижения поставленной цели обладают нечеткие НС, которые сочетают достоинства НС и нечеткой логики, опирающейся на опыт экспертов информационной безопасности. Механизм *нечеткого логического вывода* позволяет использовать опыт экспертов, овеществленный в виде системы нечетких предикатных правил, для предварительного обучения нечеткой НС [76 - 78]. Последующее *обучение* НС на поле известных угроз предоставляет возможность анализа процесса логического вывода для коррекции существующей или синтеза новой системы нечетких предикатных правил СЗИ [76, 79, 80].

Перечислим свойства нечетких НС, необходимые для адаптивных СЗИ:

1) функциональная устойчивость и защищенность элементной базы,
2) возможность классификации угроз,
3) описание соответствия «угрозы – механизмы защиты» в виде системы нечетких предикатных правил,
4) адаптивность нейро-нечетких СЗИ (системы нечетких правил),
5) «прозрачность» для анализа структуры связей нейро-нечетких СЗИ и системы нечетких правил,
6) распределенный параллелизм вычислений.

1.2. Анализ методов защиты информации в биосистемах

Целью жизни является *самовоспроизведение* путем передачи генетической информации. Важно, чтобы за время жизни структуры она успевала построить хотя бы одну свою копию [2]. Копии содержат определенный процент "информационных дефектов" – мутаций, что является существенным *условием эволюционного процесса*. *Метаболизм* (обновление) *самовоспроизведение, мутабильность*, молекулярные механизмы *переноса информации и наследования* в организмах, обеспечивают совершенствование живых информационных систем [81].

Иерархия биосферы может быть подразделена на уровни системной организации генетического материала: нуклеотидный, триплетный, генный (образуют молекулярный уровень), хромосомный, клеточный уровень, тканевый уровень, органный уровень, организменный уровень, популяционный уровень, видовой уровень, биоценотический уровень, глобальный (биосферный уровень). Каждому уровню иерархии, начиная с молекулярного уровня, присуща генетическая преемственность и информационная защищенность структур.

1.2.1. Информационная основа биосистем

Биосфера – *иерархическая информационная система* с единым подходом к способам и методам преобразования, хранения и переноса информации, которые обладают высокой защищенностью. Многообразию видов и форм существования жизни можно поставить в соответствие многообразие специализированных системы информационных технологий, различающихся по сложности структурной организации и свойствам (табл. 1.1 [3]). Имеет место аналогия между свойствами, характерными для биосферы как биосистемы и как сложной информационной системы (табл. 1.2 [3]). То есть биосфера представляет собой сложную информационную систему, подсистемы которой обладают набором механизмов и свойств, придающим им высокую информационную защищенность. Обеспечение высокого уровня защищенности и жизнеспособности видов обусловлено надежностью способа кодирования, хранения и передачи информации (в процессе размножения) - *генетического кода* вида.

Придание системам ИТ качеств биосистем и, прежде всего, отвечающих за защищенность информационных процессов, связано с наличием:

- иерархии функционально разнородных подсистем с встроенными функциями защиты,
- защищенных механизмов сохранения и передачи информации,
- свойств сложной кибернетической системы,

- эволюционных качеств, а именно: способности к зарождению, росту и развитию, обучению и адаптации в динамической внешней среде.

Таблица 1.1

Уровень биосистемы	*Функция*	*Система ИТ*	*Нейросетевая система*
1. *Ядро*	Хранение, изменение кодирование и декодирование, передача, информации	Специализированные элементы и узловые схемы	Компоненты формальных нейронов (ФН); ROM для хранения параметров ФН
2. *Клетка*	Деление, рост, матричный синтез	Специализированные процессоры	Уровень ФН
3. *Ткань*	Среда межклеточных коммуникаций	Мультипроцессор; секции среды вычислений	Фрагмент слоя из ФН; слой ФН
4. *Орган*	Функциональная специализация	Функциональное устройство	Фрагмент НС (НС); НС
5. *Организм*	Законченная локализованная система	Среда вычислений, персональный компьютер	Специализированная НС; вычислительная машина - ВМ
6. *Популяция*	Воспроизводство видовой информации	Гомогенная локальная сеть ВМ	Универсальная нейросетевая вычислительная среда
7. *Вид*	Межвидовое разграничение	Гетерогенная локальная сеть ВМ	Локальная НС корпоративного уровня
8. *Биоценоз*	Локальное сосуществование видов	Отраслевая сеть	Отраслевая сеть
9. *Биогеоценоз*	Среда для локального сосуществования	Домен глобальной сети	Домен глобальной сети
10. *Биосфера*	Глобальная взаимосвязь	Глобальная сеть	Глобальная сеть

1.2.2. Защита информации в биосистемах

Защищенность биосистемы обеспечивается механизмами наследственности и изменчивости, которые носят информационный характер. Генетическим материалом биообъектов является ДНК - дезоксирибонуклеиновая кислота [2].

Популяции существуют благодаря размножению, которое сводится к передаче внутри вида генетической информации посредством ДНК. ДНК играет роль универсального и защищенного носителя информации. Специфика ДНК заключена в ее двойственном характере: с одной стороны, как *защищенного носителя информации*, а с другой - *самой информации* в виде генетического кода.

Таблица 1.2

Биологическая система	*Сложная информационная система*
Упорядоченность системы	Наличие иерархической организации
Самовоспроизведение	Процесс сохранения и передачи информации в системе
Специфичность организации	Отличие между системами различных уровней иерархии
Целостность и дискретность	Целостность и дискретность
Рост и развитие	Способность систем к наращиванию, самообучению и развитию
Обмен веществ и энергии	Открытость системы
Наследственность и изменчивость	Перенос информации и большой потенциал изменения как кода, так и передаваемых сообщений
Раздражимость	Наличие механизмов, обуславливающих поведение системы в зависимости от внешних воздействий
Движение	Способность систем к адаптации
Внутренняя регуляция	Наличие кибернетических механизмов и информационных потоков для внутреннего регулирования системы
Специфичность взаимодействия со средой	Специфичность реагирования на внешние воздействия каждой подсистемой.

Молекулы ДНК (рис. 1.1) – это линейные макромолекулы в виде двойных цепей полимеров, составленных из нуклеотидов, каждый из которых содержит по одной молекуле фосфорной кислоты (Ф) и сахара, а также одно из четырех азотистых оснований: аденин - А, гуанин - G, цитозин - С и тимин - Т. Аденин и гуанин – пуриновые основания, цитозин и тимин – пиримидиновые. Сочетания трёх рядом стоящих нуклеотидов в цепи ДНК (триплеты, или кодоны) составляют *генетический код*. Нарушения последовательности нуклеотидов в цепи ДНК приводят к наследственным изменениям в организме — мутациям. ДНК точно воспроизводится при делении клеток, что обеспечивает передачу в поколениях наследственных признаков и специфических форм обмена веществ [2].

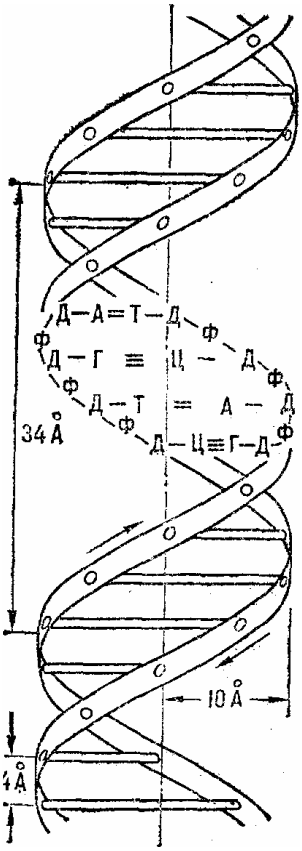

Рис. 1.1. Схема молекулы ДНК

Надежность структуры ДНК обуславливается силой водородных связей между цепями, а уникальность - тем, что разнообразие видов в природе основано на 20 аминокислотах - АМК, входящих в генный код.

Исследования ДНК выявили ряд закономерностей (правила Чаргаффа):

- число нуклеотидов, содержащих пуриновые основания равно числу нуклеотидов, содержащих пиримидиновые основания $A+G = T+C$;
- в ДНК содержание аденина равно содержанию тинина, а содержание гуанина равно содержанию цитозина $A = T$, $G = C$, $G + T = A + C$;
- ДНК разных видов могут иметь различия из-за преобладанием аденина над гуанином и тимина над цитозином ($A+T>C+G$), и наоборот ($C+G>A+T$);
- отношение $(C+G)$ / $(A+T)$ видоспецифично: во всех клетках организма отношение $(C+G)$ / $(A+T)$ одинаково.

Кодирование аминокислот - избыточное вырожденное кодирование. Число комбинаций $4^3 = 64$ втрое превышает разнообразие аминокислот, каждой из которых соответствует несколько кодонов (табл. 1.3).

Таблица 1.3

АМК	Кодон	Мольная масса (Мк)	Антикодон	Мольная масса (Ма)	Σ (Мк + Ма)
ПРО	CCC	330	GGG	450	780
	CCT	345	GGA	434	779
	CCA	354	GGT	429	779
	CCG	370	GGC	410	780
ЛЕЙ	CTC	345	GAG	434	779
	CTT	360	GAA	418	778
	CTA	369	GAT	409	778
	CTG	385	GAC	394	779
	TTA	384	AAT	393	777
	TTG	400	AAC	378	778
ГИС	CAC	354	GTG	425	779
	CAT	369	GTA	409	778
ГЛУ-NH_2	CAA	378	GTT	400	778
	CAG	394	GTC	385	779
АРГ	CGC	370	GCG	410	780
	CGT	385	GCA	394	779
	CGA	394	GCT	385	779
	CGG	410	GCC	370	780
	AGA	418	TCT	360	778
	AGG	434	TCC	345	779
СЕР	TCC	345	AGG	434	779
	TCT	360	AGA	418	778
	TCA	369	AGT	409	778
	TCG	385	AGC	394	779
	AGC	394	TCG	385	779
	AGT	409	TCA	369	778

Окончание таблицы 1.3

ФЕН	TTC	360	AAG	418	778
	TTT	375	AAA	402	777
ТИР	TAC	369	ATG	409	778
	TAT	384	ATA	393	777
НОН	TAA	393	ATT	384	777
	TAG	409	ATC	369	778
	TGA	409	ACT	369	778
ЦИС	TGC	385	ACG	394	779
	TGT	400	ACA	378	778
ТРИ	TGG	425	ACC	370	779
ТРЕ	ACC	354	TGG	425	779
	ACT	369	TGA	409	778
	ACA	378	TGT	400	778
	ACG	394	TGC	385	779
ИЛЕЙ	ATC	369	TAG	409	778
	ATT	384	TAA	393	777
	ATA	393	TAT	384	777
МЕТ	ATG	409	TAC	369	778
АСП-NH_2	AAC	378	TTG	400	778
	AAT	393	TTA	384	777
ЛИЗ	AAA	402	TTT	375	777
	AAG	418	TTC	360	778
АЛА	GCC	370	CGG	410	780
	GCT	385	CGA	394	779
	GCA	394	CGT	385	779
	GCG	410	CGC	370	780
ВАЛ	GTC	385	CAG	394	779
	GTT	400	CAA	378	778
	GTA	409	CAT	369	778
	GTG	425	CAC	354	779
АСП	GAC	394	CTG	385	779
	GAT	409	CTA	369	778
ГЛУ	GAA	418	CTT	360	778
	GAG	434	CTC	345	779
ГЛИ	GGC	410	CCG	370	780
	GGT	425	CCA	354	779
	GGA	434	CCT	345	779
	GGG	450	CCC	330	780

Правило вырожденности: если два кодона имеют два одинаковых первых нуклеотида и их третьи нуклеотиды принадлежат к одному классу (пуриновому или пиримидиновому), то они кодируют одну и ту же аминокислоту.

В ДНК двойные цепи полимеров соединены между собой водородными связями, в соответствии с *правилом комплементарности*: каждый кодон имеет только один антикодон, способный связаться с ним по всем водородным связям.

Устойчивость структуры ДНК обуславливается *силой водородных связей* между цепями полимеров: аденин и тимин образуют между собой две водородные связи (*A=T*), а гуанин и цитозин – три (*C≡G*). То есть связь *A=T* слабее связи *C≡G*. Чем больше в геноме вида отношение (*C+G)* / (*A+T*), тем вид устойчивее к внешним воздействиям. Увеличение отношения ограничивает количество кодов. Чем меньше вариантов, тем проще закодированная в геноме организация вида. Если в пределах периода цепи из 10 нуклеотидов необходимо обеспечить равномерность количества водородных связей между парами *A=T* и *C≡G*, то количество пар *A=T* д. б. равно 6, а *C≡G* – 4, так как 6*2 = 4*3.

Имеет место *уравновешенность распределения массы* ДНК. Число вариантов мольных масс системы «кодон+антикодон» равно 4 (табл. 1.4).

Таблица 1.4

Количество	Суммарная масса Мк + Ма
8	777
23	778
23	779
10	780

Суммарная масса М = Мк + Ма изменяется незначительно, что объясняется близостью мольных масс пар: для *A=T* М = 134+125 = 259 и для *C≡G* М = 110+150 = 260.

Для 64 возможных комбинаций кодонов существует 20 вариантов различных мольных масс (табл. 1.5), т. е. их количество равно числу различных аминокислот. Мольная масса по длине полинуклеотидных цепей распределена равномерно: при любом чередовании нуклеотидов в спирали ДНК структура молекулы будет уравновешенной. Максимальная равномерность масс и связей между спиралями ДНК наблюдается у позвоночных: видоспецифичное отношение $\frac{C+G}{A+T} = \frac{4}{6} = 0{,}67$.

Таблица 1.5

Количество кодонов	Мольная масса	Количество кодонов	Мольная масса
1	330	3	393
3	345	6	394
3	354	3	400
3	360	1	402
6	369	6	409
3	370	3	410
1	375	3	418
3	378	3	425
3	384	3	434
6	385	1	450

Для обеспечения информационной защищенности процесса передачи и хранения информации в ДНК используется *принцип избыточности*, как при размножении (передача информации), так и при хранении генома.

Чем сложнее организм, тем большая избыточность кода в геноме. Наиболее простая организация молекулы ДНК (без повторяющихся отрезков и пропусков в коде) у вирусов и бактерий. Простота молекулы ДНК компенсируется высокой скоростью размножения (избыточностью при передаче информации). В ядрах клеток высших организмов много избыточной ДНК - геном состоит из тысяч повторяемых участков, чередующихся с уникальными последовательностями оснований.

Прослеживается тенденция: *чем сложнее организм, тем сложнее способы размножения*. Разделение особей на мужские и женские, внутривидовое разнообразие также являются гарантом повышения защищенности существования вида.

У многоклеточных организмов хранение генетической информации осуществляется в ядрах клеток, где находится *удвоенное* количество наследственной информации - диплоидный набор хромосом. Это объясняется процессами деления клетки - одна “копия” остается в родительской клетке, а вторая передается дочерней и в последствии также удваивается.

Таким образом, основные особенности кода ДНК, обеспечивающие информационную защищенность и функциональную устойчивость биосистем, можно свести к следующему: информационная избыточность и комплементарность кодирования, равномерность распределения масс и уравновешенность системы связей по молекуле ДНК.

Клеточный принцип построения биосистем – один из основных для обеспечения информационной защищенности генома из-за значительной избыточности: достаточно одной клетки, чтобы на основе наследственной информации восстановить организм с его видовыми и индивидуальными особенностями. Биосистема - сложная система, состоящая из иерархии специализированных автономных компонентов, которые выполняют общесистемные функции по хранению всей наследственной информации и обрабатывают, декодируют только определенную часть общей информации, связанную с функциями данных компонентов.

Существует градация организмов по степени сложности - видовое разнообразие. Чем проще система (меньше структурная избыточность и защищенность), тем интенсивнее процесс передачи информации, т. е. б**о**льшая избыточность за счет высокой скорости размножения. Большие объемы компенсируют возможную потерю или модификацию информации при передаче. Обратно, чем сложнее система, тем большая структурная избыточность и меньше скорость размножения.

Используется избыточность и самих информационных сообщений - большое число повторяющихся последовательностей нуклеотидов в кодах. Процесс передачи информации становится более защищенным - половое размножение. В процессе трансляции сообщений осуществляется избыточная передача информации с одновременным увеличением периода между трансляциями.

Клетка является наименьшей структурой, которая осуществляет хранение и декодирование информации. Общая организация процессов декодирования информации внутри клетки обладает повышенной информационной защищенностью: декодирование триплетного кода ведется по *принципу сопоставления* - каждый кодон имеет только один антикодон, способный связаться с ним по всем водородным связям. В процессе декодирования ДНК используются свойства комплементарности и близости мольных масс пар нуклеотидов: 260 ($C+G$) и 259 ($A+T$). Важно также, что декодирование в клетке ведется не самой ДНК, а с ее копии - *i*РНК.

Ядро можно представить как компонент системы, в котором осуществляются только процессы хранения информации и копирования ее частей (аналог режима Read only). То есть оригинальная генетическая информация не покидает ядра и не претерпевает изменений (свойство стабильности), а дубликат информации подлежит дальнейшим преобразованиям с возможностью фиксации изменений во вновь созданных компонентах системы (свойство пластичности).

Особенности клетки как защищенной системы по хранению и обработке информации состоят в следующем:

- генетическая (системная) информация хранится в обособленной структуре - ядре, защищающем ее от внешних воздействий;
- декодирование генома производится над дублем системной информации вне ядра специальными обрабатывающими структурами, которые используют при декодировании принцип сопоставления при соблюдении комплементарности кода.

Таким образом, отдельные клетки и биологические организмы в целом являются информационными системами, которые благодаря иерархической организации, методам и принципам хранения, кодирования и декодирования информации являются информационно защищенными системами.

1.3. Моделирование систем защиты информации и оценки защищенности систем ИТ

Моделирование СЗИ и оценки уровня защищенности систем ИТ – необходимый этап для автоматизации процедур анализа уязвимостей и выявления атак на корпоративную систему с целью придания ИТ эволюционных свойств адаптивности и развития [82-84].

1.3.1. Моделирование систем защиты информации

В печати встречаются сообщения о разработке эффективных методик, способных снизить расходы от внедрения СЗИ, например, использующих *имитационные модели* [85]. Методики ориентированы на решение задачи создания экономически оптимальной СЗИ в разрезе инвестиций, *минимизирующих общий ущерб* при нарушениях ИБ. Применение относительно недорогих способов и средств обеспечения ИБ (антивирусные программы, организационные ограничения и т. п.) существенно снижает общий ущерб. Поэтому инвестиции СЗИ в сравнительно малых размерах эффективны в небольших организациях, не подвергающихся специальным компьютерным атакам. Для динамичных компаний, функционирующих в конкурентной изменяющейся среде, рост затрат на СЗИ не всегда ведет к снижению ущерба от атак на корпоративную систему.

Часто модели защиты являются частью *системы управления рисками* и учитывают такие параметры, как актуальные угрозы, имеющиеся ошибки в программном обеспечении, важность, интервал и время простоя различных ресурсов, вероятность атаки, варианты защиты и возможная величина *ущерба*. Система управления рисками в системе ИТ позволяет просчитывать существующие риски, моделировать оптимальный комплекс контрмер, автоматически разрабатывать профиль защиты и оценивать остаточные риски [9].

Биосистемная аналогия в структуре защиты систем ИТ базируется на *иерархии СЗИ*, встроенных механизмах иммунной защиты и накопления опыта.

Известные СЗИ, как правило, ограничиваются реализацией функций нижнего уровня системы защиты и антивирусной направленностью средств иммунной защиты. Согласно [86] около 70% вирусных атак осуществляется извне через точку входа в защищаемую сеть и только около 30 % изнутри. Первые можно отнести к внешним угрозам жизнеобеспечению системы, вторые - внутренним. В обоих случаях задействуется *иммунная защита* биосистемы. Реализация идеи информационной иммунной системы состоит в том, что в случае обнаружения в сети признаков заражения отправляют образец нового вируса в антивирусный центр, откуда, спустя некоторое время, получают обновление антивирусной базы, которое распространяют по корпоративной сети прежде, чем успеет распространиться вирус.

Названный подход входит в противоречие с биосистемной аналогией, в частности, с *внутрисистемной* реализацией иммунной защиты, т. к. в рассмотренной системе антивирусной защиты (в отличие от биосистемы) большая часть механизмов иммунной защиты находится в антивирусном центре, расположенным за пределами корпоративной сети.

Размещение антивирусного центра *вне* защищаемой системы ИТ позволяет злоумышленникам: во-первых, под видом обновления антивирусной базы сформировать канал для загрузки вирусов и троянских коней, во-вторых, в случае автоматической отправки на анализ подозреваемых на наличие вируса файлов получить доступ к конфиденциальной информации. Кроме того, время реакции подобной иммунной защиты в лучшем случае измеряется часами, что, наряду с перечисленными возможностями реализации каналов НСД, мало приемлемо для большинства критических приложений. Следовательно, сфера применения подобного подхода ограничена только восстановлением выведенной из строя корпоративной сети (аналог процесса реанимации больной биосистемы с помощью инъекций).

В биосистемах функции иммунной защиты реализуются через

- *внутренние* механизмы *оперативной реакции* на угрозы и дестабилизирующие воздействия, распределенные по уровням *иерархии* СЗИ,
- *долговременные процессы* накопления жизненного опыта, носящие эволюционный характер [81, 87].

Биосистемная аналогия систем ИТ в *эволюционных процессах* основана на реализации совокупности механизмов наследования, развития, адаптации и отбора, свойственных биосистемам. В то время как разрабатываемые интеллектуальные средства выявления атак и несанкционированных информационных процессов в корпоративной сети основное внимание уделяют лишь свойству адаптивности при построении перспективных СЗИ [21, 47, 55]. Причем СЗИ уровня почтовых шлюзов и межсетевых экранов в большей мере ориентирова-

ны на выявление внешних атак, а СЗИ серверного уровня - нейтрализацию внутренних угроз в корпоративной системе.

Известные интеллектуальные СЗИ [20-68], как правило, реализуют только механизмы *оперативной реакции* и нейтрализации угроз жизнедеятельности системы ИТ, практически не уделяя внимание координирующей роли, которую играет нервная система - верхний уровень иерархии защиты биологических систем в реализации эволюционного процесса накопления жизненного опыта системы (*долговременного запоминания* системной информации). В биосистемах имеют место процессы постепенной адаптации иерархической системы жизнеобеспечения и защиты с использованием всего арсенала средств эволюционных процессов.

В ИТ-системах помимо иммунного уровня СЗИ необходима *иерархия уровней защиты* и, прежде всего, наличие верхних уровней СЗИ (например, рецепторного уровня средств защиты), выполняющих функции нервной системы биологического организма по накоплению жизненного опыта, координации и установлению ассоциативных (долговременных) связей между процессами, происходящими на нижних уровнях СЗИ - атаками и изменением поля угроз. Другими словами, в системах ИТ, в частности, корпоративной или локальной сети необходим иерархический уровень накопления жизненного опыта по нейтрализации атак, представленного в форме *структурированных информационных полей*, удобных для наследования в последующих реализациях системы.

1.3.2. Методы оценки защищенности систем ИТ

Известны оценки защищенности системы ИТ, исходящие из наличия определенного набора средств и механизмов защиты, методик изготовления, эксплуатации и тестирования, позволяющие отнести то или иное устройство или систему ИТ к одному из дискретных уровней защищенности в соответствии с используемыми в данной стране стандартами [88].

В работе [89] предложено в качестве оценки защищенности использовать рейтинговый показатель, который учитывает распределение механизмов защиты по эшелонам многоуровневой модели системы информационной безопасности и изменение вероятности достижения злоумышленником объекта защиты в зависимости от эшелона многоуровневой модели СЗИ. К недостаткам модели следует отнести статичный характер оценки защищенности системы ИТ, не учитывающей такие параметры как ущерб от реализации угроз ИБ и частоту осуществления атак.

В работе [90] защищенность оценивается исходя из ущерба от реализации в системе ИТ угроз, носящих случайный характер, который оценивается через коэффициенты опасности угроз. Причем коэффициенты опасности представ-

ляются нечеткими величинами, а показатель защищенности системы ИТ определяется посредством формируемой методом экспертных оценок матрицы нечетких отношений между коэффициентом опасности совокупности угроз и степенью защищенности системы ИТ. Недостатком подобного оценивания является отсутствие привязки показателей защищенности к местоположению МЗ в структуре СЗИ. Как и в предыдущем случае сохраняется статичность оценки защищенности ИБ системы ИТ.

В работе [85] предлагается для проведения инвестиционного анализа СЗИ и оценки ущерба в случае реализации угроз ИБ учитывать ущерб, как в стоимостном исчислении, так и "нематериальный" ущерб, нанесенный репутации, конкурентным возможностям хозяйствующего субъекта (табл.1.6) [85].

Таблица 1.6

Величина ущерба	Характеристика показателя "величина нематериального ущерба"
Ничтожный	Ущербом можно пренебречь
Незначительный	Ущерб легко устраним, затраты на ликвидацию последствий реализации угрозы невелики
Умеренный	Ликвидация последствий реализации угрозы не связана с крупными затратами и не затрагивает критически важные задачи, но положение рынке ухудшается, часть клиентов теряется
Серьезный	Затрудняется выполнение критически важных задач. Утрата на длительный период положения на рынке. Ликвидация последствий реализации угрозы связана со значительными инвестициями
Критический	Реализация угрозы приводит к невозможности решения критически важных задач. Организация прекращает существование

В последнем случае вводят семантические показатели "величина нематериального ущерба" и «вероятность нанесения ущерба», которая связана с частотой реализации угрозы за определенный период времени (табл.1.7) [85].

Таблица 1.7

Частота реализации угрозы	Значение вероятности	Семантическая характеристика реализации угрозы
Нулевая	Около нуля	Угроза практически никогда не реализуется
1 раз за несколько лет	Очень низкая	Угроза реализуется редко
1 раз за год	Низкая	Скорее всего, угроза не реализуется
1 раз в месяц	Средняя	Скорее всего, угроза реализуется
1 раз в неделю	Выше средней	Угроза почти обязательно реализуется
1 раз за день	Высокая	Шансов на положительный исход нет

По мнению [85], нет приемлемых методик для нахождения нужного оптимума для *динамичных* компаний, функционирующих в конкурентной изменяющейся среде. Анализ различных вариантов обеспечения ИБ по критерию "стоимость/эффективность" предлагается осуществлять, учитывая соображения:

- стоимость СЗИ не должна превышать определенную сумму (как правило, не более 20% от стоимости системы ИТ);
- уровень ущерба не должен превышать некоторое значение, например, "незначительный".

Известные [88-90] оценки отражают *статическое* состояние объекта защиты, исходя из наличествующих механизмов защиты, не учитывают действительную загруженность механизмов защиты по нейтрализации последствия атак, динамику изменения поля угроз, возможность адаптации СЗИ к изменению поля угроз, не дают указаний на изменение состава механизмов защиты и структуры многоуровневой СЗИ.

Выводы по главе 1

1. Показана необходимость придания системам защиты информации в ИТ эволюционных качеств, присущих биосистемам, и, прежде всего, *возможность развития и адаптивность*. Наряду с традиционными средствами защиты корпоративных сетей, такими как: антивирусы, детекторы уязвимостей, межсетевые экраны и детекторы вторжений используются средства *автоматизации защиты*, включающие коррелляторы событий, программы обновлений, средства ЗА (аутентификации, авторизации и администрирования) и системы управления рисками.

Анализ показал, что для обнаружения атак применяют системы защиты информации, в качестве интеллектуального инструмента в которых, как правило, используются нейронные сети, системы нечеткой логики и основанные на правилах экспертные системы; что необходимо решать не отдельные задачи защиты информации, а разрабатывать единый подход применения интеллектуальных средств для создания комплексной адаптивной защиты систем ИТ на основе биоаналогии.

2. Биологические системы образуют *иерархию информационных систем* с единым подходом к способам и методам преобразования, хранения и переноса информации, которые обладают высокой защищенностью. Защищенность биосистемы обеспечивается механизмами наследственности и изменчивости, которые носят информационный характер.

Особенности кода ДНК, обеспечивающие информационную защищенность биосистем:

- информационная избыточность и комплементарность кодирования,
- равномерность распределения масс и уравновешенность системы связей по молекуле ДНК,

а особенности клетки:

- генетическая информация хранится в обособленной структуре - ядре, защищающем ее от внешних воздействий;
- декодирование генома производится над дублем системной информации вне ядра специальными обрабатывающими структурами, которые используют при декодировании принцип сопоставления при соблюдении комплементарности кода.

Показано что, отдельные клетки и биологические организмы в целом являются информационными системами, которые благодаря иерархической организации, методам и принципам хранения, кодирования и декодирования информации являются информационно защищенными системами.

3. Моделирование СЗИ и разработка показателей защищенности систем ИТ – необходимый этап для автоматизации процедур анализа уязвимостей и выявления атак на корпоративную систему с целью придания ИТ эволюционных качеств адаптивности и развития

Анализ показал, что для разработки эффективных методик, способных снизить расходы от внедрения СЗИ, используют имитационные модели, модели системы управления рисками, которые учитывают актуальность угроз, имеющиеся ошибки в программном обеспечении, важность, интервал и время простоя различных ресурсов, вероятность атаки, варианты защиты и возможная величина ущерба, что позволяет моделировать оптимальный комплекс контрмер и автоматически разрабатывать профиль защиты.

Отмечено, что биосистемная аналогия в структуре защиты систем ИТ основана на иерархии СЗИ, встроенных механизмах иммунной защиты и накопления опыта. Известные СЗИ, как правило, ограничиваются антивирусной направленностью средств иммунной защиты и реализацией функций нижнего уровня в иерархии СЗИ.

Существующие показатели защищенности системы ИТ отражают *статическое* состояние объекта защиты, исходя из наличия механизмов защиты, и не учитывают активность механизмов защиты по нейтрализации последствия атак, динамику изменения поля угроз, возможность адаптации СЗИ к изменению поля угроз, не дают указаний на изменение состава механизмов защиты и структуры многоуровневой СЗИ.

4. Показано, что перспективными задачами обеспечения ИБ являются:

- Разработка модели адаптивной информационной защиты систем ИТ на ос-

нове нейро-нечетких средств защиты информации с использованием аналогии с иерархией защитных механизмов биологических систем.

- Разработка системы оценок информационной защищенности систем ИТ, учитывающей структурные и экономические показатели адаптивной системы защиты информации.
- Разработка метода построения адаптивной системы защиты информации на основе предложенных оценок и иерархической адаптивной модели СЗИ.
- Разработка архитектурных решений нейросетевых СЗИ.
- Разработка инструментальных средств для поддержки метода построения адаптивной СЗИ.

2. РАЗРАБОТКА АДАПТИВНОЙ МОДЕЛИ СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ

Наблюдается тенденция использования в создаваемых человеком сложных технических системах элементов организации живой природы. В частности, в области информационных технологий данная тенденция проявляется в искусственных нейронных сетях, топология которых ближе к организации нервной системы биологических систем, чем к архитектуре современных систем ИТ.

2.1. Иерархия уровней системы защиты информации

Как следует из предыдущей главы, биологическим системам свойственна иерархическая организация системы защиты информации. Биосистемная аналогия в структуре защиты систем ИТ базируется на иерархии СЗИ: механизмах иммунной защиты и механизмах накопления опыта в информационных полях нейронных сетей нервной системы.

Особую роль в эволюции биосистем играет *нервная система* как адаптивный инструмент взаимодействия с внешней средой. Нервная система - феномен самоорганизации возникла для формирования элементарных рефлексов в ответ на внешние воздействия. Т. е. *рефлексия* является продуктом верхних уровней информационной защиты биосистемы в результате внешнего раздражения. Информация о рефлексах сохраняется в генетической памяти на нижних уровнях информационной защиты и передается по наследству [81].

Феномен самоорганизации обусловливает *целенаправленность поведения* биосистемы, приводит к необходимости в *системе воспитания* развивает новую форму памяти в виде *адаптивного информационного поля НС* [91].

Переход нервной системы в качественно новую форму связан с появлением в биосистеме *поведенческих реакций*, свидетельствующих о развитии сложной связи между внешними воздействиями и реакцией организма. Происходит

разделение информации между носителями различной природы: *ДНК* и *нервными клетками*. Поведенческая информация формируется на основе генетически передаваемых посредством ДНК поведенческих реакций, фиксируемых в информационном поле нервной системы. Однако поведенческие реакции биосистемы не ограничиваются только передаваемыми по наследству. Для них характерно накопление *жизненного опыта* и передача его потомкам *через обучение*. Результаты обучения фиксируются в ДНК для передачи в поколениях.

Построение безопасных интеллектуальных систем ИТ основано на иерархической организации информационной защиты, а также:

- биосистемной аналогии в архитектуре систем ИТ,
- известных механизмах информационной защиты биосистем, а именно:
  - иерархия уровней защиты в биосфере: нуклеотид - кодон – ген – хромосома – ДНК -…- организм - … - биосфера,
  - на нижних уровнях иерархии (кодон – ген – хромосома – ДНК) организовано сохранение генетической информации, реализация механизма мутаций, кодирование и декодирование информации, разделение сообщений по критерию «свой/чужой»,
  - на верхних уровнях иерархии – реализована связь системы с внешней средой через органы чувств – рецепторы и накопление опыта в НС нервной системы,
  - изменение генетической информации связано не с изменением формы представления, а содержания информации – жизненного опыта,
  - информационная безопасность биосистемы обеспечивается за счет адаптивности - приобретения жизненного опыта, позволяющего успешно оперировать смысловыми ситуациями, в частности, распознавать своих и чужих, выбирать поведение в сложной и постоянно изменяющейся окружающей среде,
- наличии иерархии уровней информационной защиты систем ИТ:
  - информация в адаптивных СЗИ хранится в виде информационных полей на 2-х уровнях иерархии: внизу, как поля идентифицирующего угрозы и вверху иерархии, как поля жизненного опыта, ставящего в соответствие полю известных угроз механизмы защиты информации,
  - нижний уровень адаптивных СЗИ – иммунный, на котором осуществляется проверка соответствия передаваемых сообщений в системе по критерию «свой/чужой», проверяется форма представления информации (контейнер),

- идентифицирующая информация - своя для каждой системы и связана с формой, но не содержанием информации,
- верхний уровень СЗИ – рецепторный необходим для связи с внешней средой и накопления опыта в виде информационного поля адаптивных СЗИ,
- перенос и наследование информации в адаптивных СЗИ – это передача информационных полей НС иммунного и рецепторного уровней, сформированных в процессе жизненного цикла некоторой системы ИТ, в последующие реализации системы (потомкам),

• свойствах НС, необходимых для реализации функций информационной защиты:

- возможность наследования ранее накопленного опыта подобных систем в виде информационных полей нижнего и верхнего уровней адаптивных СЗИ,
- способность к кластеризации (расширению классификации) угроз - адаптация информационного поля уровней иерархии адаптивных СЗИ,
- коррекция жизненного опыта адаптивных СЗИ - адаптация информационного поля уровней иерархии СЗИ,
- возможность анализа, коррекции и переноса (наследование) информации в СЗИ других систем.

2.2. Методика проектирования адаптивной СЗИ

Метод проектирования адаптивных систем защиты информации базируется на основных свойствах НС и нечетких систем, связанных с адаптивностью, обучаемостью, возможностью представления опыта специалистов информационной безопасности (ИБ) в виде системы нечетких правил, доступных для анализа.

Возможность *обучения* рассматривается как одно из наиболее важных качеств нейросетевых систем, которое позволяет адаптироваться к изменению входной информации. Обучающим фактором являются избыточность информации и скрытые в данных закономерности, которые видоизменяют информационное поле НС в процессе адаптации. НС, уменьшая степень избыточности входной информации, позволяет выделять в данных *существенные признаки*, а соревновательные методы обучения - классифицировать поступающую информацию за счет *механизма кластеризации*: подобные вектора входных данных группируются нейронной сетью в отдельный кластер и представляются конкретным формальным нейроном - ФН-прототипом. НС, осуществляя кластеризацию данных, находит такие усредненные по кластеру значения функциональ-

ных параметров ФН-прототипов, которые минимизируют ошибку представления сгруппированных в кластер данных.

Метод проектирования адаптивной защиты систем ИТ включает:

1) решение задачи *классификации* а) угроз по вектору признаков атак и б) механизмов защиты (МЗ) по вектору угроз; производится соотнесение *посылок* (на нижних уровнях защиты - нечеткого вектора признаков атак, на верхних уровнях защиты - нечеткого вектора угроз) с классификационными *заключениями* (на нижних уровнях - выявленными угрозами, на верхних уровнях – механизмами защиты, необходимыми для нейтрализации поля известных угроз);

2) решение задачи *кластеризации* угроз по признакам атак и МЗ по вектору угроз как саморазвитие классификации при расширении поля угроз; производится разбиение входных векторов на группы (на нижних уровнях защиты - векторов признаков атак, на верхних уровнях защиты - векторов угроз) и отнесение вновь поступающего входного вектора к одной из групп либо формирование новой группы (на нижних уровнях - группы угроз, на верхних уровнях – группы механизмов защиты, необходимых для нейтрализации поля известных угроз);

3) формирование матриц *экспертных оценок* для определения степени соответствия на нижних уровнях защиты - угроз признакам атаки и, на верхних уровнях защиты – механизмов защиты полю угроз;

4) представление в виде *систем нечетких правил* результатов решения задач П. 1 и 3, полученных в процессе нечеткого логического вывода классификационных заключений по нечетким посылкам (на нижних уровнях защиты - соотношения «признаки атаки - угрозы», на верхних уровнях защиты – соотношения «угрозы - МЗ»);

5) реализацию систем нечетких правил в виде специализированных структур - *нейро-нечетких классификаторов* (на нижних уровнях защиты - классификаторов «признаки атаки - угрозы», на верхних уровнях защиты – классификаторов «угрозы - МЗ»);

6) реализацию результатов решения задачи п.2 в виде *четких классификаторов* на основе самообучающейся НС (на нижних уровнях защиты - классификаторов «признаки атаки - угрозы», на верхних уровнях защиты – классификаторов «угрозы - МЗ»);

7) *наследование* (передачу) *опыта* адаптивной СЗИ по обеспечению информационной безопасности, приобретенного в процессе эксплуатации подобной ИТ-системы, в проектируемую СЗИ путем перенесения информационных полей четких и нейро-нечетких классификаторов (на нижних уровнях защиты - классификаторов «признаки атаки - угрозы», на верхних уровнях защиты – классификаторов «угрозы - МЗ»);

8) *обучение классификаторов* по П. 5, 6 на обучающей выборке – подмножестве входных векторов (на нижних уровнях защиты - векторов признаков атак, на верхних уровнях защиты - векторов угроз) с целью формирования информационных полей четких и нейро-нечетких классификаторов;

9) *адаптацию* в процессе эксплуатации ИТ-системы *информационных полей* четких и нейро-нечетких классификаторов (на нижних уровнях защиты - классификаторов «признаки атаки - угрозы», на верхних уровнях – классификаторов «угрозы - МЗ»);

10) *коррекцию* адаптируемых матриц *экспертных оценок* (п. 3) и систем *нечетких правил* (п. 4) по результатам адаптации;

11) *формулирование новых* нечетких *правил* в случае расширения классификации по результатам выполнения П. 2 и 9 (на нижних уровнях защиты - классификации «признаки атаки - угрозы», на верхних уровнях – классификации «угрозы - МЗ»);

12) *формирование* комплекса *оценок защищенности* ИТ-системы исходя из результатов выполнения п. 10 и распределения механизмов защиты по иерархии СЗИ;

13) *анализ структуры связей* нейро-нечетких классификаторов, «прозрачной» *системы нечетких правил* и комплекса *оценок защищенности* по п. 12 для выявления наиболее используемых или отсутствующих в ИТ-системе механизмов защиты;

14) *формирование спецификации* на разработку отсутствующих МЗ;

15) *коррекция структуры* системы информационной безопасности за счет расширения перечня используемых МЗ и их размещения в иерархии адаптивной СЗИ.

Порядок действий согласно методу проектирования адаптивных СЗИ может изменяться, но обязательными являются:

1) формирование матриц *адаптируемых экспертных оценок* и на их основе создание исходных *систем нечетких правил* и *классификаторов* (на нижних уровнях защиты - классификаторов «признаки атаки - угрозы», на верхних уровнях защиты – классификаторов «угрозы - МЗ»);

2) *идентификация* выявленной *угрозы* и при расширении поля известных угроз - *кластеризация угроз* с последующей адаптацией информационных полей путем обучения НС уровней защиты;

3) *кластеризация* вследствие изменения поля угроз сопровождается *коррекцией* или *расширением системы нечетких правил*;

4) *изменение поля угроз* вызывает *модификацию систем нечетких правил* и матриц *экспертных оценок* в результате обучения классификаторов уровней защиты;

5) при расширении системы нечетких правил формируется *описание* нового (отсутствующего) *механизма защиты*;

6) «прозрачность» системы нечетких правил позволяет сформулировать *спецификацию на создание* отсутствующего МЗ;

7) на основании анализа комплекса оценок защищенности ИТ-системы (в случае экономической целесообразности) включают новый МЗ в состав СЗИ.

2.3. Разработка иерархической модели адаптивной системы защиты информации

Модель адаптивной защиты использует принцип биосистемной аналогии, в частности, иерархию системы защиты информационных процессов и ресурсов в биологической системе, согласно которой на нижних уровнях иерархии задействованы механизмы иммунной системы, а на верхних - механизмы адаптивной памяти и накопления жизненного опыта нервной системы [84].

Модель адаптивной защиты в системах ИТ характеризуется следующими атрибутами: СЗИ - многоуровневая иерархическая, использует экспертные оценки для привнесения априорного опыта в СЗИ в виде системы нечетких предикатных правил, эволюционный характер СЗИ обеспечивается, прежде всего, адаптивными свойствами нейро-нечетких сетей, реализующих систему нечетких предикатных правил.

Внизу иерархии СЗИ решается задача *классификации/кластеризации атак* по совокупности признаков, носящих неполный и не вполне достоверный характер. Т. е. нейронная сеть нижнего уровня СЗИ, исходя из опыта экспертов ИБ, реализует систему нечетких правил, которая описывает процесс логического вывода получения заключения (тип атаки), используя в качестве нечетких посылок векторы входных признаков.

На нижних уровнях иерархии используют аппаратно-программные средства идентификации атак, в том числе и нейросетевые [58]. Задача нечеткой классификации успешно решается с применением нейро-нечетких сетей [76, 101].

Если достоверность классификации по известным угрозам меньше некоторого уровня, то при наличии признаков атаки классификация расширяется за счет введения новой градации в классификацию – решается задача кластеризации угроз.

Кластеризация расширяет систему нечетких правил соответствующих уровне СЗИ, т. к. классифицируется ранее неизвестная угроза.

На верхних уровнях иерархии защиты для каждого эшелона многоуровневой СЗИ средства защиты информации используют результаты классификации нижних уровней иерархии в виде посылок системы нечетких предикатных правил для формирования заключений - соответствий «угрозы-механизмы защи-

ты». То есть решается задача *классификации механизмов защиты* (нечеткие заключения) по вектору нечетких признаков угроз, для нейтрализации последствий которых данные МЗ предназначены.

Другими словами, для каждого эшелона многоуровневой СЗИ, используя результаты нечеткой классификации (тип атаки) в качестве посылок, системой нечетких правил описывается соответствие «угрозы – механизмы защиты», исходя из опыта экспертов ИБ. НС данного уровня СЗИ после обучения будет отражать достоверность нейтрализации заданного в отдельном правиле набора угроз соответствующим механизмом защиты рассматриваемого эшелона многоуровневой СЗИ.

Если при увеличении размерности вектора признаков угроз после обучения НС достоверность классификации по механизмам защиты (активность механизмов защиты отдельных эшелонов) меньше некоторого уровня, то при наличии признаков атаки классификация МЗ расширяется за счет введения новой градации в классификацию – задача кластеризации механизмов защиты.

После обучения нечеткой НС соответствующего эшелона анализ нечеткого правила по вновь введенному МЗ позволяет сформулировать спецификацию на отсутствующий механизм защиты.

Для эшелонов многоуровневой СЗИ на основе экспертных оценок целесообразно сформировать лингвистические переменные «частота реализации угрозы» и «потенциальный ущерб» (например, табл. 1.6 и 1.7).

Верхний уровень иерархии СЗИ также необходим для обобщения результатов (посылок) в виде активности МЗ, частоты реализации и ущерба от угрозы с целью формирования системы нечетких предикатных правил - заключений о целесообразности расширения состава *активированных* механизмов защиты по отдельным эшелонам СЗИ. Активация МЗ производится, если *интегральные оценки*, учитывающие величину потенциального ущерба, частоту реализации угроз и достоверность нейтрализации угроз данным механизмом защиты, превышают заданные пороговые значения.

2.3.1. Структура иерархической модели адаптивной СЗИ

При проектировании адаптивной системы защиты информации следует учитывать комплексный характер решаемой задачи (рис. 2.1).

Связующим звеном адаптивной модели СЗИ является методика оценки защищенности ИТ-системы, которая координирует взаимосвязь классификаторов угроз и механизмов защиты (в виде НС, нечетких НС, систем нечетких предикатных правил), структурной модели СИБ, инструментальных средств расчета показателей защищенности и рейтинга ИТ-системы.

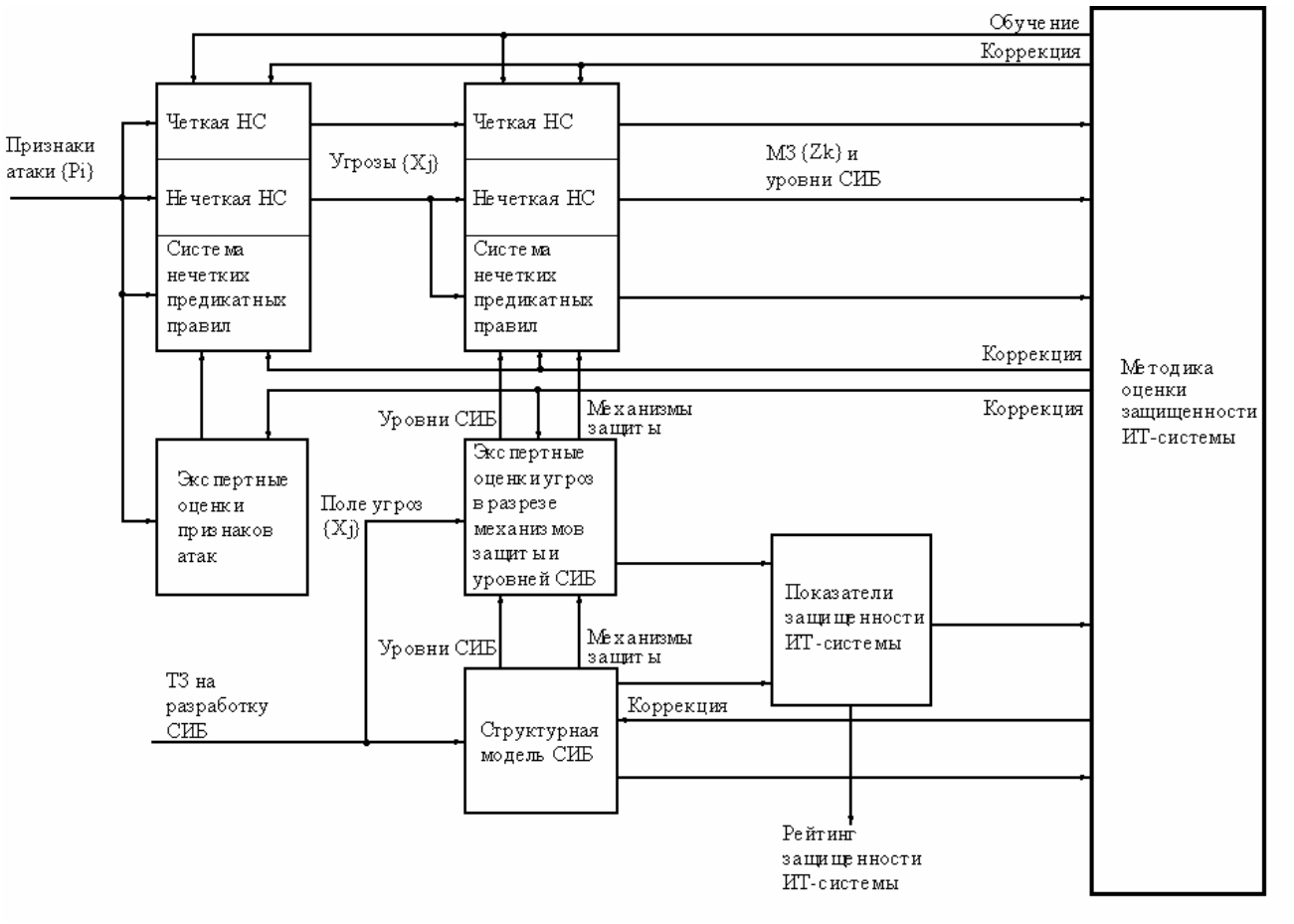

Рис. 2.1. Структура модели адаптивной СЗИ

Динамичный характер поля угроз выдвигает свойство адаптивности ИТ-систем в разряд первоочередных качеств, необходимой СЗИ. С другой стороны, не менее важным качеством является возможность реализации в СЗИ *накопленного опыта*, который овеществляется в виде информационно-полевой компоненты иерархии механизмов защиты. Однако нецелесообразно в объекте информатизации использовать всевозможные МЗ, а ограничиваются минимальным комплектом, достаточным для отражения угроз, оговоренных в спецификации на проектирование ИТ-системы.

В соответствии с заданием на проектирование системы защиты информации выбирается структурная модель СИБ в виде иерархии уровней механизмов защиты, а априорный опыт экспертов представляется массивами экспертных оценок, на базе которых формируются системы нечетких предикатных правил для классификации 1) угроз по признакам атак и 2) МЗ на поле угроз.

Системы нечетких предикатных правил для последующей адаптации и анализа представляются в виде нечетких НС, которые обучают на некотором подмножестве входных векторов признаков атаки. Одновременно обучают классификаторы в виде обычных НС таким образом, чтобы число образуемых кластеров равнялось числу правил в системе нечетких предикатных правил.

Аналогично обучают нейросетевые классификаторы механизмов защиты по векторам известных угроз.

Для исходных массивов экспертных оценок производят расчет показателей защищенности и рейтинга ИТ-системы, которые используются методикой оценки защищенности ИТ-системы для анализа и коррекции, как массивов экспертных оценок, так и функциональных параметров нейросетевых классификаторов и систем нечетких предикатных правил.

Информация в адаптивной СЗИ хранится и может передаваться в поколениях (тиражирование и последующие модификации ИТ-систем) в виде распределенных адаптивных информационных полей НС: 1) *поля известных угроз* иммунных уровней защиты и 2) поля *жизненного опыта* рецепторных уровней защиты. Процесс адаптации первых связан с решением задач классификации, кластеризации, приводящих к расширению информационного поля известных угроз на нижних уровнях иерархии СЗИ. Изменение перечня известных угроз ИБ отражается на верхних уровнях иерархии СЗИ в соответствующей модификации информационного поля жизненного опыта, реализованного в виде специализированных структур нечетких НС, которые, в свою очередь, описывается системами нечетких предикатных правил. Процесс адаптации вторых связан с обучением нечетких НС (конструктивные алгоритмы обучения), которое адекватно видоизменяет систему нечетких предикатных правил, ставящую в соответствие известным угрозам механизмы защиты информации.

2.3.2. Механизмы реализации модели адаптивной СЗИ

Основным механизмом реализации *адаптивных свойств* СЗИ следует считать способность нечеткого распределенного информационного поля нейронной сети к *накоплению знаний* в процессе обучения.

Вторым по важности механизмом с точки зрения адаптивной модели СЗИ является *нечеткий логический вывод*, который базируется на нечетком представлении информации в НС и позволяет использовать опыт экспертов в области информационной безопасности, овеществленный в виде системы нечетких предикатных правил, для *предварительного обучения* нейро-нечеткой сети).

Возможность отображения системы нечетких предикатных правил на структуру СЗИ и последующее ее *обучение* на поле известных угроз позволяют не только устранить противоречивость исходной системы нечетких предикатных правил, но и дает возможность проанализировать сам процесс логического вывода с целью дальнейшего уточнения существующей или синтеза новой системы нечетких предикатных правил адаптивной СЗИ.

Третьим механизмом, необходимым для реализации адаптивных СЗИ, является способность нейронных и нейро-нечетких сетей к *классификации и кластеризации*.

Нечеткий логический вывод

Нечеткие НС сочетают достоинства нейросетевых ВС и нечетких логических систем, опирающихся на априорный опыт в виде заданной системы *нечетких предикатных правил*. Механизм нечетких выводов основан на базе знаний, формируемой специалистами предметной области (экспертами) в виде системы нечетких предикатных правил вида:

Π_1 : если x есть A_1 , то y есть B_1,

Π_2 : если x есть A_2 , то y есть B_2,

...

Π_n : если x есть A_n , то y есть B_n,

где x и y, соответственно, входная переменная (например, угроза) и переменная вывода (к примеру, механизм защиты), а A_i и B_i - функции принадлежности семантических данных.

Нечеткое отношение $R = A \to B$ отражает знания эксперта $A \to B$ в виде причинного отношения предпосылки (угрозы) и заключения (механизма защиты), где операция $\to$ соответствует нечеткой импликации. Отношение R можно рассматривать как нечеткое подмножество прямого произведения $X \times Y$ полного множества угроз X и механизмов защиты Y, а процесс получения нечеткого результата вывода B' по предпосылке A' и знаниям $A \to B$ - в виде композиционного правила: $B' = A' \bullet R = A' \bullet (A \to B)$, где $\bullet$ - операция, например, max-min-композиции [102].

Логический вывод, как правило, включает следующие этапы (рис. 2.2) [76]:

1) *Введение нечеткости (fuzzification)*: по функциям принадлежности, заданным на области определения входных НП, исходя из фактических значений НП, назначается степень истинности каждой угрозы для каждого правила;
2) *Логический вывод*: по степени истинности угроз формируются заключения по каждому из правил, образующие нечеткое подмножество для каждого механизма защиты;
3) *Композиция*: полученные на этапе 2 нечеткие подмножества для каждого механизма защиты объединяются с целью формирования нечеткого подмножества для всех механизмов защиты (по всем правилам);

4) *Приведение к четкости (defuzzification)*: сводится к преобразованию нечеткого набора выводов по всем правилам в четкое значение итоговой защищенности системы.

Этапы логического вывода для системы нечетких правил:

Π_1 : если x есть A , то w есть D,

Π_2 : если y есть B , то w есть E,

Π_3 : если z есть C , то w есть F,

проиллюстрированы на рис. 2.2 [102], где *x, y* и *z* – входные НП, соответствующие известным угрозам, *w* - НП вывода, соответствующая итоговой защищенности системы, а *A, B, C, D, E, F*- функции принадлежности семантических данных.

1) на основании значений непрерывных переменных по семантикам *A, B, C* находятся степени истинности $\alpha(x_0) = A(x_0),\ \alpha(y_0) = B(y_0),$ и $\alpha(z_0) = C(z_0)$ угрозы для каждого из нечетких предикатных правил; 2) операцией min в соответствии со степенью истинности $\alpha(x_0),\ \alpha(y_0),$ и $\alpha(z_0)$ удаляются верхние части семантик *D, E* и *F,* формируются заключения по каждому из правил, образующие нечеткое подмножество для каждого механизма защиты; 3) операцией max производится объединение усеченных семантик и формирование комбинированного нечеткого подмножества, описываемого семантикой $\mu_\Sigma(w)$ и соответствующего логическому выводу для выходной переменной *w* итоговой защищенности системы ИТ; 4) определяется значение выходной непрерывной переменной, например, с использованием центроидного метода находится центр тяжести w_0 для кривой $\mu_\Sigma(w)$.

Нечеткая классификация

В механизме классификации адаптивных СЗИ целесообразно использовать сочетание возможностей НС и нечеткой логики. Нейронные сети и системы с нечеткой логикой имеют специфические особенности: с одной стороны, возможность обучения НС, а с другой, процесс решения задач системами с нечеткой логикой прозрачен для объяснения получаемых выводов. Объединение двух подходов *в нечетких НС* позволяет сочетать достоинства нейросетевых и нечетких логических систем, опирающихся на априорный опыт специалистов в области информационной безопасности.

Как следует из опыта разработки нечетких НС (таб. 2.1) [76] для целей классификации реализуют нейро-нечеткие сети типа 1, которые решают задачу отнесения нечеткого входного вектора к четкому классу, а нейро-нечеткие сети

типов 2, 3 и 4 применяют для построения нечетких систем, основанных на системе нечетких правил вывода.

Таблица 2.1

Fuzzy neural net	*Weights*	*Inputs*	*Targets*
Type 1	crisp	fuzzy	crisp
Type 2	crisp	fuzzy	fuzzy
Type 3	fuzzy	fuzzy	fuzzy
Type 4	fuzzy	crisp	fuzzy
Type 5	crisp	crisp	fuzzy
Type 6	fuzzy	crisp	crisp
Type 7	fuzzy	fuzzy	crisp

Рассмотрим подход к организации нейро-нечеткого классификатора, использующего механизм нечеткого логического вывода для классификации МЗ по нечетким векторам угроз нейронной сетью с нечеткими связями [101].

Механизм нечеткого логического вывода основан на базе знаний, формируемой экспертами информационной безопасности в виде системы нечетких предикатных правил вида:

Π_1 : если $\tilde{x}_1$ есть A_{11} и ... $\tilde{x}_n$ есть A_{1n}, то $\tilde{y} = B_1$,

Π_2 : если $\tilde{x}_1$ есть A_{21} и ... $\tilde{x}_n$ есть A_{2n}, то $\tilde{y} = B_2$,

...

Π_k : если $\tilde{x}_1$ есть A_{k1} и ... $\tilde{x}_n$ есть A_{kn}, то $\tilde{y} = B_k$,

где $\tilde{x}_i$ и $\tilde{y}_j$ - нечеткие входные переменные и переменные вывода, соответствующие угрозам и МЗ, а A_{ij} и B_i, $i = \overline{1,k}$, $j = \overline{1,n}$ - функции принадлежности.

Пусть задано полное пространство угроз (предпосылок) $X = \{\tilde{x}_1,..., \tilde{x}_m\}$ и полное пространство механизмов защиты (заключений) $Y = \{\tilde{y}_1,..., \tilde{y}_n\}$. Между $\tilde{x}_i$ и $\tilde{y}_j$, $i = 1...m$, $j = 1...n$, существуют нечеткие причинные отношения $\tilde{x}_i \to \tilde{y}_j$, которые можно представить в виде матрицы R с элементами r_{ij}, $i = 1...m$, $j = 1...n$, а предпосылки и заключения - как нечеткие множества A и B на пространствах X и Y, отношения которых можно представить в виде: $B = A \bullet R$, где $\bullet$ - операция композиции, например, max-min-композиция.

Для реализации системы нечетких предикатных правил нейро-нечеткий классификатор механизмов защиты по нечетким векторам угроз должен выполнять следующие действия:

- *введение нечеткости* - по функциям принадлежности, заданным на области определения входных НП, в соответствии со значением НП назначается степень истинности для каждой угрозы;
- *логический вывод* - по степени истинности угроз формировать заключения по каждому из правил, образующие нечеткое подмножество для каждой переменной вывода - МЗ;
- *композиция* - полученные на предыдущем этапе нечеткие подмножества для каждой переменной вывода по всем правилам объединять с целью формирования нечеткого подмножества для всех переменных вывода.

В полном пространстве угроз $X = \{\tilde{x}_1,..., \tilde{x}_m\}$ максимально число входных нечетких векторов задается всевозможными сочетаниями координат $\tilde{x}_i, \quad i = 1...m$. Каждому входному вектору из пространства *X* можно поставить в соответствие нечеткий ФН нейро-нечеткого классификатора, выполняющий операцию логического вывода, например, min. Отображение множества результатов логического вывода в полное пространство заключений $Y = \{\tilde{y}_1,..., \tilde{y}_n\}$ можно реализовать посредством операции композиции, и каждому выходному вектору из пространства *Y* можно поставить в соответствие нечеткий ФН нейро-нечеткого классификатора, выполняющий операцию, к примеру, *max*.

Нейро-нечеткий классификатор *m*-мерных нормализованных векторов угроз ***X*** с нечеткими координаторами $(\tilde{x}_1,..., \tilde{x}_m)$ будем представлять в виде трехслойной нечеткой НС (рис. 2.3), в которой:

- первый слой содержит *m*, по числу координат входного вектора угроз, нечетких ФН с комплементарными нечеткими связями, формирующих *m* пар нечетких высказываний вида: $\tilde{x}_i$ *есть* S, $\tilde{x}_i$ *есть* L, $\quad i = 1...m$;
- средний слой содержит до 2^m нечетких ФН, выполняющих операцию логического вывода (например, *min*) над сочетаниями НВ 1-го слоя НС для формирования системы нечетких классификационных заключений;
- выходной слой содержит *n*, по числу координат выходного вектора, нечетких ФН, выполняющих операцию композиции (например, *max*) над классификационными заключениями 2-го слоя НС для формирования *n*-мерных векторов ***Y*** выходных нечетких заключений $(\tilde{y}_1,..., \tilde{y}_n)$.

Нечеткие ФН 1-го слоя формируют комплементарные пары значений истинности для входных НП $\tilde{x}_i$ координат входного вектора угроз ***X***.

При заданном значении координаты вектора угроз ***X*** на отрезке области определения каждому значению НП соответствует значение ординат функций принадлежности *S* (small) и *L* (large), которые в сумме дают 1 (рис. 2.4).

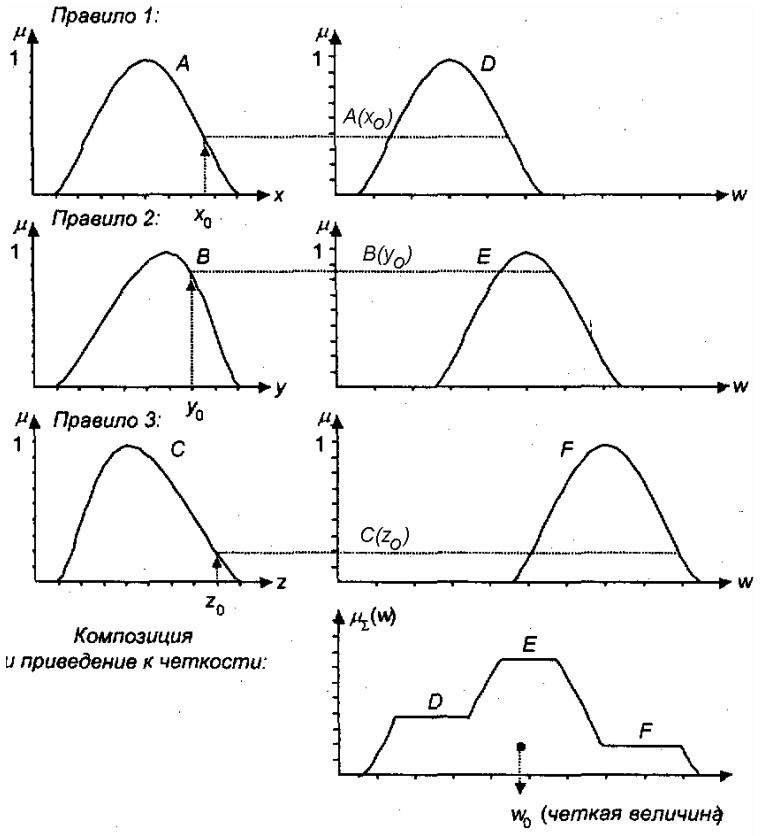

Рис. 2.2. К иллюстрации процедуры логического вывода

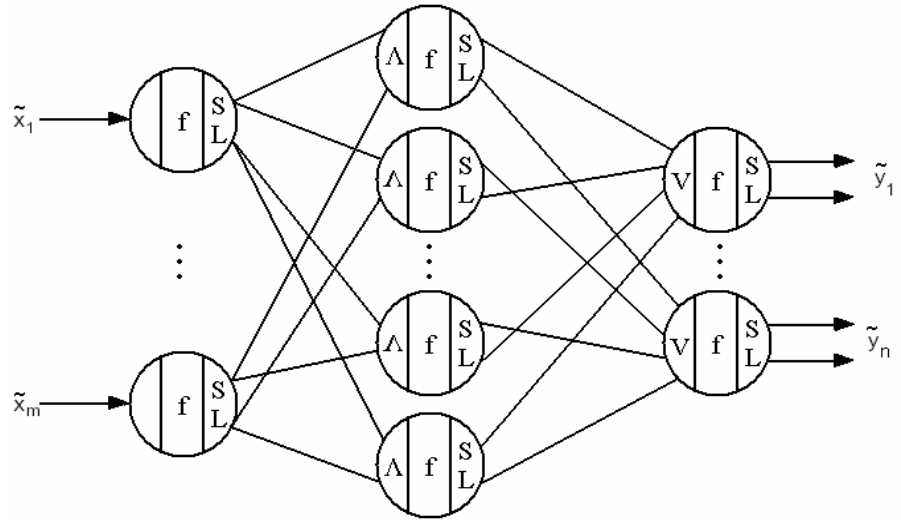

Рис. 2.3. Нейро-нечеткий классификатор

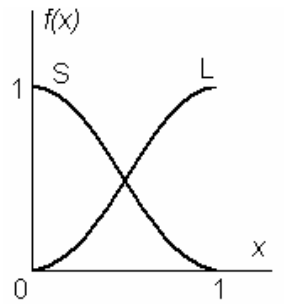

Рис. 2.4. Функции принадлежности комплементарной нечеткой связи

Пара функций принадлежности, например *S* и *L,* образуют две нечеткие связи, составляющие одну комплементарную нечеткую связь.

Если во 2-м слое нечеткой НС содержится максимальное число нечетких ФН «И», то промежуточный вектор нечетких заключений будет содержать всевозможные нечеткие классификационные заключения, которые могут следовать из всех возможных векторов угроз.

Третий слой нечеткой НС образован из нечетких нейронов «ИЛИ» (по числу нечетких заключений $\tilde{y}_j, \quad j = 1...n$) и формирует вектор выходных нечетких заключений в соответствии с заданной экспертами информационной безопасности системой нечетких предикатных правил.

Последующее обучение нейро-нечеткого классификатора МЗ по нечетким векторам угроз может производиться по алгоритмам адаптации нейро-нечетких сетей, в частности с использованием механизма нечеткой связи. Обучение нейро-нечеткого классификатора на наборе векторов известных угроз (обучающая выборка) позволит выявить возможную противоречивость системы нечетких предикатных правил и устранить из структуры НС незначащие связи (неточные заключения в системы нечетких правил) [76, 102].

Введение избыточности в информационные поля нейросетевых классификаторов СЗИ

Обратной стороной специализации слоев нечеткой НС, обеспечивающей структурную «прозрачность» информационного поля нейро-нечеткого классификатора СЗИ, является отсутствие информационной избыточности, что негативно сказывается на функциональной устойчивости и защищенности информационных полей НС к деструктивным воздействиям.

При сохранении специализации отдельных слоев нейро-нечеткоих сетей в соответствии с правилами нечеткого логического вывода, удобной для последующего анализа, необходимо ввести избыточность в информационное поле нейро-нечеткого классификатора. Избыточность информационного поля создает предпосылки для распределенного хранения информации в структурированных полях нечеткой НС в виде системы комплементарных нечетких связей [118], а структурная специализация слоев ФН в нечетких НС позволяет анализировать результаты обучения информационных полей НС.

Систему нечетких правил логического вывода можно отождествлять с формальным описанием логических преобразований нечетких высказываний (НВ). В качестве аппарата для формализации преобразований над НВ можно использовать аналог нормальных форм для четких высказываний в виде дизъюнктивной (ДНФ) и конъюнктивной (КНФ) нормальных форм. Причем НВ на выходе функции *S* соответствует инверсному значению некоторой нечеткой переменной, а *L* – прямому значению той же переменной (рис. 2.5).

Если применить подобный подход комплементарного дублирования и к скрытым слоям нейро-нечеткой сети, то можно добиться формирования взаи-

мопротивоположных результатов, как для этапа логического вывода, так и этапа композиции, что позволяет увеличить избыточность информационных полей нейро-нечеткого классификатора (рис. 2.6).

Представляется целесообразной также следующая форма введения избыточности в информационное поле нейро-нечеткого классификатора – увеличение размерности входных данных путем добавления к входному вектору X вектора Z текущего состояния СИБ (рис. 2.7).

Подобная коррекция структуры СЗИ вызывает не только увеличение размерности входных данных классификатора, но и расширяет систему нечетких правил логического вывода, которая учитывает не только координаты входного вектора X, но и координаты вектора Z текущего состояния СИБ, что, в свою очередь, также приводит к возрастанию избыточности информационного поля нейро-нечеткого классификатора. В процессе работы классификатора производится не только идентификация вектора Y по векторам X и Z, но и формируются предложения C по изменению состояния системы.

Рассмотрим модель (рис. 2.8) адаптивной системы информационной безопасности, в которой отражены изменения в структуре уровней защиты, аналогичные приведенным на рис. 2.7.

Для иммунного уровня защиты координаты вектора Z могут отражать системные характеристики ИТ-системы, к примеру, такие как:

- тип установленного программного обеспечения и обновлений к нему,
- работающие сервисы,
- поддержка многозадачности,
- поддержка многопользовательского режима,
- наличие в ИТ-системе таких устройств ввода/вывода информации, как дисководы, CD, DVD-приводы, USB-порты и пр.,
- наличие устройств «горячей» замены, к примеру, RAID массивов, других средств резервного копирования информации,
- возможность беспроводного доступа в систему и пр.

Для рецепторного уровня защиты координаты вектора Z могут отражать структурные характеристики СЗИ, к примеру, такие как:

- множество используемых в СЗИ механизмов защиты,
- распределение МЗ по иерархии СЗИ,
- активность уровней иерархии СЗИ,
- активность используемых в СЗИ механизмов защиты,
- показатели защищенности ИТ-системы, включая рейтинговые и пр.

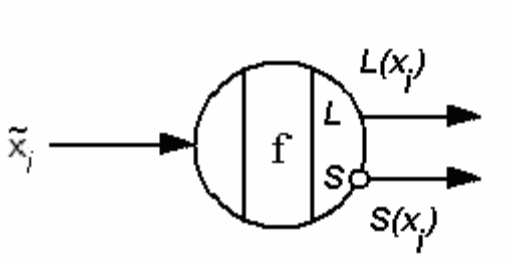

Рис. 2.5. Входной узел нейро-нечеткого классификатора

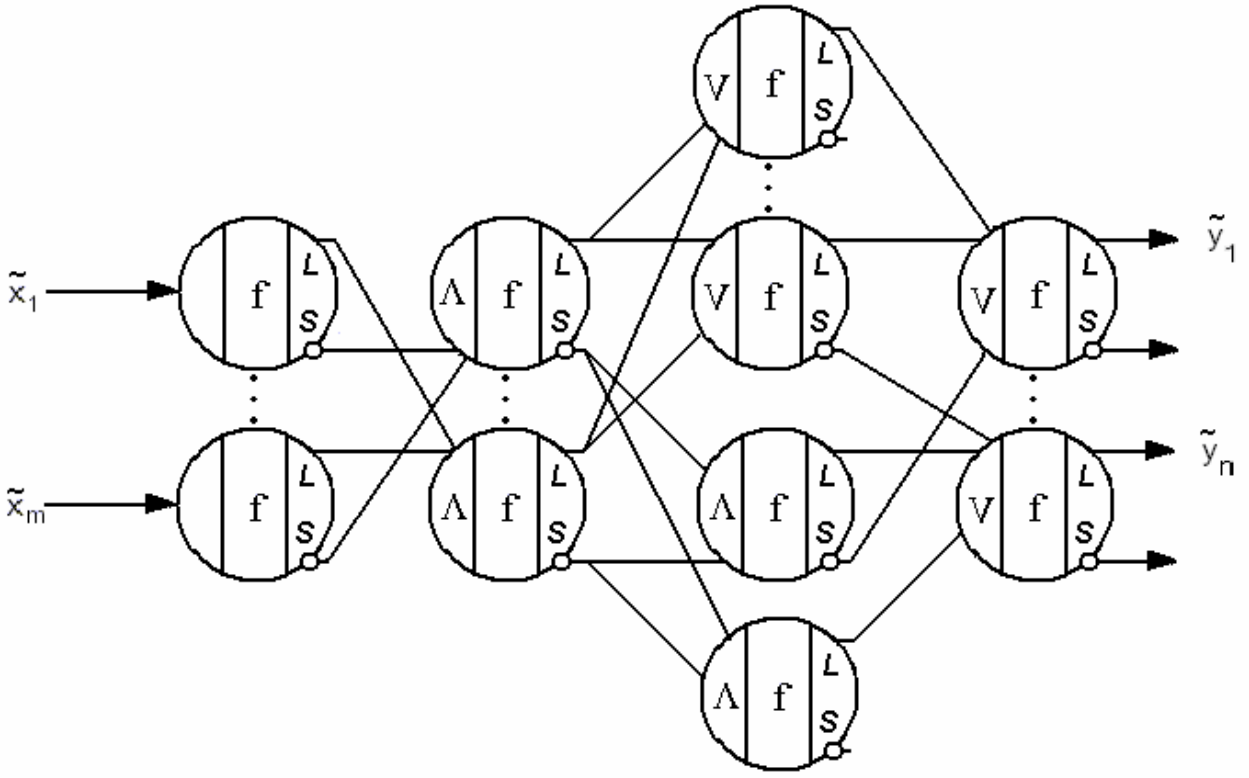

Рис. 2.6. Структура избыточного нейро-нечеткого классификатора

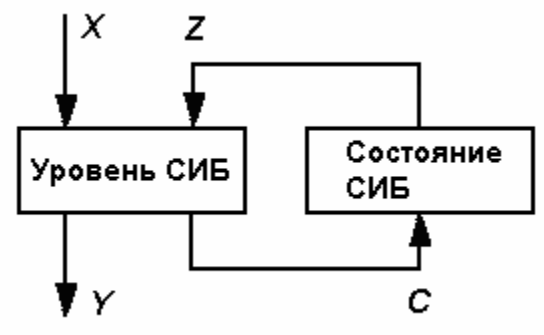

Рис. 2.7. Коррекция иерархического уровня нейро-нечеткого классификатора

Наличие регистров состояния в составе иерархических уровней СЗИ приводит к существенному возрастанию избыточности информационных полей НС, как за счет увеличения размерности входных векторов классификаторов, так и последующего приведения формальной записи системы нечетких правил логического вывода к аналогу совершенной формы (например, СовДНФ).

Анализ информационных полей обученных нечетких НС классификаторов уровней адаптивной защиты, сформированных с учетом текущего состояния ИТ-системы и СЗИ, позволяет оценивать влияние отдельных координат векторов *X* и *Z* на вектор *Y* (например, на идентификацию угроз).

В частности, на иммунном уровне защиты целесообразно при идентификации угроз учитывать состояние ИТ-системы, включая аппаратно-программную составляющую СЗИ, а на рецепторном уровне защиты - активность отдельных МЗ, уровней СЗИ, показатели защищенности ИТ-системы.

Накопление опыта в адаптивных СЗИ

Накопление опыта в информационных полях нейро-нечетких классификаторов, входящих в состав иерархических уровней адаптивных СЗИ, происходит в процессе обучения нейросетевых средств защиты информации.

Рассмотрим вариант алгоритма обучения нейро-нечетких классификаторов, основанных на системе нечетких предикатных правил. Предположим, что нейро-нечеткий классификатор с размерностью входного вектора *N* (, например, число заданных угроз) по обучающему множеству $\{(\boldsymbol{x}^1, y^1),...,(\boldsymbol{x}^n, y^n)\}$ должен реализовать некоторое отображение: $y^k = f(\boldsymbol{x}^k) = f(x_1^k, x_2^k,..., x_N^k), \quad k = 1,...,n,$ где *k* – размерность обучающего множества.

Для описания отображения *f* используем один из алгоритмов нечеткого вывода, применяя следующую систему предикатных правил для всех *i* = 1,.... *m,* где *i* – количество используемых механизмов защиты:

Π_i : если x_1 есть v_{i1} и если x_2 есть v_{i2} ... и если x_n есть v_{in}, то $y = z_i$,

где v_{ij} - семантическое данное, соответствующее *j*-й уязвимости для *i*-го механизма защиты, z_i - вещественное число, определяющее степень использования *i*-го механизма защиты в формировании значения итоговой защищенности системы.

Степень истинности *i*-го правила α_i определяется с помощью моделирования логического оператора «И», например, операцией умножения:

$\alpha_i = \prod_1^n v_{ij}(x_j^k)$. В соответствии с центроидным методом выход системы определяется как: $o^k = \left(\sum_{i=1}^m \alpha_i z_i\right) / \left(\sum_{i=1}^m \alpha_i\right)$, а функции ошибки для *k-го* предъявленного образца, например, как: $E_k = 0.5(o^k - y^k)^2$.

Для подстройки параметров системы исходных предикатных правил в нечеткой НС можно использовать градиентный метод и как в обычных НС корректировать величины $z_i := z_i - \eta \frac{\partial E_k}{\partial z_i} = z_i - \eta(o^k - y^k)\frac{\alpha_i}{\alpha_1 + ...\alpha_m}$, $i = 1,..., m$, где η - константа скорости обучения.

Изначально в адаптивном уровне СЗИ формируется система нечетких предикатных правил для всех *известных* механизмов защиты информации $\{z_k, k = \overline{1,K}\}$, также как нейросетевые средства идентификации угроз обучены на всем поле *известных угроз* $\{x_p, p = \overline{1,P}\}$. *Незаданным* угрозам во входном векторе $\boldsymbol{x}$ соответствуют нулевые значения координат, а *деактивированным* механизмам защиты – близкие к 0 значения степени использования данного механизма защиты в формировании значения итоговой защищенности системы.

Задавая пороговые значения для величин $z_k, k = \overline{1,K}$, можно определять, как наименее задействованные, так и эффективно используемые механизмы в обеспечении безопасности защищаемой системы.

После активации всех потенциальных механизмов защиты информации и введения дополнительных ФН в последний скрытый слой НС, соответствующий размерности вектора известных механизмов защиты, происходит *расширение системы нечетких предикатных правил*. Таким образом, СЗИ самостоятельно формируют правило, описывающее *отсутствующий* механизм защиты информации в защищаемой системе. При последующей адаптации произойдет обучение СЗИ под отсутствующий механизм защиты информации, направленный на нейтрализацию неспецифицированной угрозы x_p. Анализ дополнительного нечеткого предикатного правила позволяет сформировать спецификацию на проектирование отсутствующего в системе средства или механизма защиты информации.

2.3.3. Модель адаптивной СЗИ и этапы жизненного цикла систем ИТ

Целью этапов проектирования и создания системы жизненного цикла является формирование корректной (без несанкционированных возможностей) при-

кладной информационно безопасной системы ИТ. На начальном этапе жизненного цикла в соответствии с требованиями спецификации на проектирование системы осуществляется формирование системы ИТ и СЗИ с заданной совокупностью свойств.

Для реализации функции СЗИ, соответствующих системе нечетких предикатных правил (например, для классификации механизмов защиты), формируются адаптивные информационные поля адаптивных уровней защиты прикладной системы ИТ. Производится предэксплуатационное обучение нейронечетких классификаторов и нейронных кластеризаторов с применением корректных алгоритмов, т. е. выполняется адаптация информационных полей НС под задачи информационной защиты.

Процессы настройки (обучения) производятся в режиме адаптации системы при непосредственном участии и под контролем доверенных лиц, в частности, администратора системы ИТ. Процесс настройки завершается блокировкой режима адаптации и переводом сформированной системы в режим работы.

Многоуровневая модель информационной безопасности системы на первом этапе соответствует минимальной активации потенциальных механизмов защиты и полноте информационного поля известных угроз.

Целью этапа эксплуатации жизненного цикла системы является корректное исполнение системой заданных функций. Основной режим – работа. Предусмотрен режим адаптации функций системы защиты информации, который использует механизм адаптации для реагирования на изменение внешних факторов - происходит дальнейший рост, самообучение системы и изменение информационных полей СЗИ. Как и на предыдущем этапе, процессы коррекции функций СЗИ производятся в режиме адаптации системы при непосредственном участии администратора системы ИТ. Процесс настройки завершается блокировкой режима адаптации и переводом системы в режим работы.

Многоуровневая модель адаптивной СЗИ на втором этапе динамически пополняется путем перевода механизмов защиты из статуса «потенциальный» в статус «активированный» и привязки активированного механизма к соответствующему эшелону модели СЗИ. Увеличивается число элементов в подмножестве заданных угроз, как за счет включения элементов из множества известных угроз, так и за счет пополнения самого множества известных угроз ранее неизвестными угрозами. Возможно расширение множества потенциальных механизмов защиты за счет описания в виде нечетких предикатных правил и последующей реализации ранее отсутствующих механизмов защиты.

Целью этапа вывода системы из эксплуатации является постепенное сворачивание прикладных функций системы при корректной работе СЗИ и сохранении основных системных функций.

Модель информационной безопасности прикладной системы достигает максимального насыщения, как механизмами защиты, так и содержанием информационного поля известных угроз. Накопленный опыт СЗИ подлежит анализу и использованию (наследованию) в создаваемых ИТ-системах.

2.4. Разработка комплекса показателей для систем ИТ

Применение модели адаптивной защиты, основанной на принципе биологической аналогии [3] позволяет [92]:

- обеспечить близкое к оптимальному соотношение "стоимость/ эффективность" СЗИ за счет постепенного наполнения многоуровневой модели ИБ только необходимыми механизмами защиты,
- в динамике отслеживать наиболее задействованные механизмы защиты при изменении поля угроз,
- формировать спецификацию требований на отсутствующие механизмы защиты,
- оценивать защищенность системы ИТ через величины относительного ущерба и интегральные показатели активности распределенных по структуре СЗИ механизмов защиты.

2.4.1. Показатели защищенности системы ИТ

Решение о расширении классификаций атак и механизмов защиты производится в соответствии с системой *оценок достоверности* нейтрализации угроз в разрезе отдельных механизмов защиты или отдельных эшелонов СЗИ и аналогичных *оценок потенциального ущерба*, также соотносимых с отдельными механизмами защиты или отдельными эшелонами СЗИ. Далее по тексту потенциальный ущерб будем рассматривать в относительных величинах, к примеру, по отношению к значению максимально допустимого ущерба в информационной системе хозяйствующего субъекта.

Можно использовать распределение используемого в системе ИТ подмножества механизмов защиты по эшелонам многоуровневой модели СЗИ, аналогичное изображенному на рис. 2.9 [89], учитывая, что количество механизмов защиты и требований безопасности, оговоренных в действующих стандартах информационной безопасности, превышает 250 (см., например, [100]).

Результаты экспертных оценок, а также последующего обучения нечетких НС могут быть представлены в виде матрицы достоверности «угрозы – механизмы защиты» *ME*

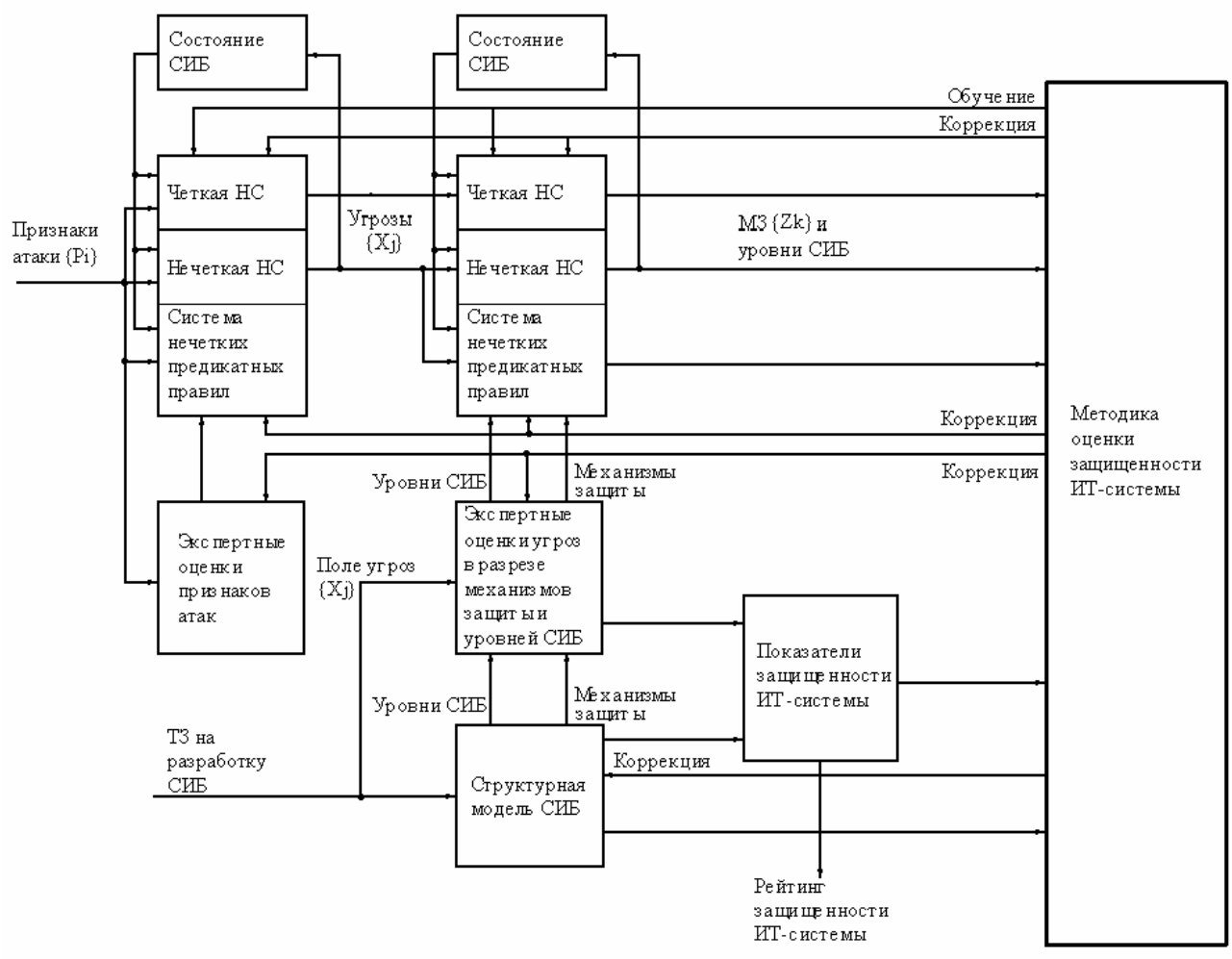

Рис. 2.8. Модель адаптивной системы информационной безопасности

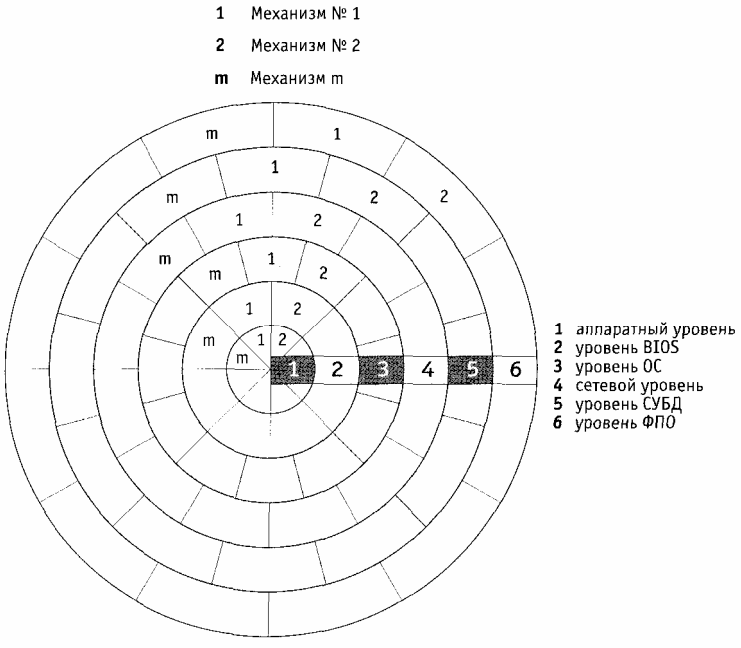

Рис. 2.9. Иллюстрация распределения механизмов защиты по эшелонам

$$ME_{m\times n} = \left(\begin{array}{ccc} me_{11} & me_{12} & me_{1\,n} \\ me_{21} & me_{22} & me_{2\,n} \\ \hline me_{m\,1} & me_{m\,2} & me_{m\,n} \end{array}\right),$$

где m – число механизмов защиты, n - число эшелонов СЗИ.

Активность эшелона СЗИ по нейтрализации угроз, входящих в систему предикатных правил в качестве посылок, определяется строкой интегральных показателей, представленных, например, строкой показателей *значимости* (как в [93]) *эшелона* в многоуровневой СЗИ

$$x_j = \sqrt[m]{\prod_{i=1}^{m} me_{ij}}\ ,\ j = 1,...,n\ , \qquad (1)$$

нормированных, например, по значению максимального из $x_j,\ j = 1,...,n$ или по значению суммы элементов строки показателей значимости $\sum_{j=1}^{m} x_j$, $j = 1,...,n$.

Сопоставление интегральных показателей в пределах строки позволяет выявить наиболее задействованные эшелоны в многоуровневой модели СЗИ по нейтрализации поля действующих на систему ИТ угроз.

Аналогично по матрице достоверности использования механизмов защиты для нейтрализации угроз можно получить столбец интегральных показателей *активности* использования отдельного *механизма защиты* во всех эшелонах многоуровневой СЗИ для нейтрализации последствий действующего поля угроз

$$x_i = \sqrt[n]{\prod_{i=1}^{n} me_{ij}}\ ,\ i = 1,...,m\ . \qquad (2)$$

Сопоставление интегральных показателей в пределах столбца позволяет выявить наиболее задействованные механизмы защиты в многоуровневой СЗИ.

Анализ интегральных показателей матрицы достоверности «угрозы – механизмы защиты» дает возможность обосновать целесообразность использования механизма защиты в составе соответствующего эшелона многоуровневой СЗИ.

Использование экспертных оценок и последующее отражение в структуре нейро-нечеткой сети априорного опыта экспертов ИБ сопровождается проверкой на непротиворечивость результатов опроса экспертов. Непротиворечивость оценок экспертов ИБ может быть обеспечена применением, например, метода экспертных оценок матрицы нечетких отношений [90] или метода на основе расчета максимального собственного значения матрицы парных сравнений [93].

Приведенные выше показатели будут более информативными, если учитывать не только достоверность использования механизмов защиты в структуре СЗИ, но и показатели *потенциального ущерба*, возникающего в результате реализации атак на систему ИТ и который может быть предотвращен системой информационной безопасности. С этой целью по аналогии с [90] *оценку защищенности* можно косвенно связать с *предотвращением ущерба* системе ИТ, и, кроме того, использовать экспертные оценки для сопоставления, с одной стороны, поля угроз ИБ с потенциальным ущербом от их реализации, с другой стороны, размера потенциального ущерба с местом реализации угрозы в структуре ИТ.

2.4.2. Методика оценки защищенности системы ИТ

Для каждого эшелона многоуровневой СЗИ формируется экспертная оценка *достоверности нейтрализации* поля известных угроз известными механизмами защиты и *потенциального ущерба*, исходя из опыта экспертов ИБ. Ущерб от реализации угроз в системе ИТ следует оценивать в относительных величинах, например, по отношению к максимально допустимой для данного хозяйствующего субъекта величине. Расчет потенциального ущерба производится за определенный промежуток времени с учетом частоты активации угроз (например, табл.1.7).

1. Исходные данные (*экспертные оценки)* представляют в матричной форме.

Для каждого эшелона многоуровневой СЗИ оценивается достоверность нейтрализации угроз механизмами защиты с последующим формированием матрицы достоверности «МЗ-угрозы» *MT*

$$MT_{m\times p} = \left(\begin{array}{cc:c} mt_{11} & mt_{12} & mt_{1p} \\ mt_{21} & mt_{22} & mt_{2p} \\ \hdashline mt_{m1} & mt_{m2} & mt_{mp} \end{array}\right),$$

i = 1,..., m – число механизмов защиты, j = 1,..., p - число известных угроз, и матрицы достоверности «угрозы-эшелоны» *TE*

$$TE_{p\times n} = \left(\begin{array}{cc:c} te_{11} & te_{12} & te_{1n} \\ te_{21} & te_{22} & te_{2n} \\ \hdashline te_{p1} & te_{p2} & te_{pn} \end{array}\right),$$

i = 1,..., p - число известных угроз, j = 1,..., n - число эшелонов СЗИ.

Для каждого эшелона многоуровневой СЗИ оценивается уровень потенциального ущерба и формируются матрицы «эшелоны-ущерб» *ET*

$$ET_{n\times p}=\begin{pmatrix} et_{11} & et_{12} & et_{1\,p} \\ et_{21} & et_{22} & et_{2\,p} \\ \hline et_{n\,1} & et_{n\,2} & et_{n\,p} \end{pmatrix},$$

$i = 1,..., n$ - число эшелонов СЗИ, $j = 1,..., p$ - число известных угроз, и матрицы «ущерб-МЗ» *TM*

$$TM_{p\times m}=\begin{pmatrix} tm_{1\,1} & tm_{1\,2} & tm_{1\,m} \\ tm_{2\,1} & tm_{2\,2} & tm_{2\,m} \\ \hline tm_{p1} & tm_{p2} & tm_{p\,m} \end{pmatrix},$$

$i = 1,..., p$ - число известных угроз, $j = 1,..., m$ – число механизмов защиты.

2. Для каждого эшелона многоуровневой СЗИ экспертные оценки в виде системы нечетких предикатных правил отображают в структуре нейро-нечетких сетей. В процессе последующей адаптации нечетких НС в составе иерархических СЗИ на обучающей выборке, соответствующей некоторому подмножеству поля известных угроз производится *автоматическая коррекция* системы нечетких предикатных правил, а также показателей потенциального ущерба и достоверности (истинности) нейтрализации набора угроз соответствующим эшелоном или МЗ многоуровневой СЗИ. Корректность исходных экспертных оценок может быть проверена сопоставлением элементов вышеперечисленных матриц либо сопоставлением интегральных оценок защищенности до и после процесса обучения нейро-нечетких СЗИ.

3. *Интегральные оценки защищенности* получают в результате операций над матрицами. В частности умножение матриц достоверности «МЗ-угрозы» *MT* и «угрозы-эшелоны» *TE* позволяет получить матрицу «МЗ-эшелоны» *ME* – матрицу достоверности активации известных механизмов защиты, распределенных по эшелонам многоуровневой СЗИ, для нейтрализации известных угроз

$$ME_{m\times n}=\begin{pmatrix} me_{11} & me_{12} & me_{1\,n} \\ me_{21} & me_{22} & me_{2\,n} \\ \hline me_{m\,1} & me_{m\,2} & me_{m\,n} \end{pmatrix},$$

$i = 1,..., m$ – число механизмов защиты, $j = 1,..., n$ - число эшелонов СЗИ, а умножение матриц потенциального ущерба «эшелоны-ущерб» *ET* и «ущерб-МЗ» *TM* - матрицу потенциального ущерба «эшелоны-МЗ» *EM,* отражающую распределение потенциального ущерба от реализации известных угроз по механизмам защиты и эшелонами многоуровневой СЗИ

$$EM_{n\times m} = \left(\begin{array}{ccc} em_{11} & em_{12} & em_{1m} \\ em_{21} & em_{22} & em_{2m} \\ \hline em_{n1} & em_{n2} & em_{nm} \end{array}\right),$$

$i = 1,..., n$ - число эшелонов СЗИ, $j = 1,..., m$ – число механизмов защиты.

Промежуточные оценки в виде строки (1) и столбца (2) интегральных показателей характеризуют *активность* использования отдельного *механизма защиты* либо отдельного *эшелона* в рамках многоуровневой СЗИ, а также позволяют оценить потенциальный *ущерб* в разрезе механизмов защиты и эшелонов системы информационной безопасности.

4. Дальнейшие операции над матрицами *ME* и *EM* дают возможность обобщить в диагональных элементах *итоговой матрицы* как показатель достоверности активации механизма защиты в результате атаки, так и потенциального ущерба от ее реализации.

Умножением матрицы достоверности *ME* и матрицы потенциального ущерба *EM* получают квадратную *матрицу достоверности потенциального ущерба* «МЗ-МЗ» *MM*

$$MM_{m\times m} = \left(\begin{array}{ccc} mm_{11} & mm_{12} & mm_{1m} \\ mm_{21} & mm_{22} & mm_{2m} \\ \hline mm_{m1} & mm_{m2} & mm_{mm} \end{array}\right),$$

$i = j = 1,..., m$ – число МЗ, а умножением матрицы *EM* и матрицы *ME* получают квадратную матрицу достоверности потенциального ущерба «эшелоны-эшелоны» *EE*

$$EE_{n\times n} = \left(\begin{array}{ccc} ee_{11} & ee_{12} & ee_{1n} \\ ee_{21} & ee_{22} & ee_{2n} \\ \hline ee_{n1} & ee_{n2} & ee_{nn} \end{array}\right),$$

$i = j = 1,..., n$ - число эшелонов СЗИ.

Для матрицы *MM* в качестве обобщающего показателя можно рассматривать вектор, образованный диагональными элементами $mm_{ij} = p_i$, $i = j = 1,\ldots, m$, матрицы - *вектор достоверности распределения потенциального ущерба по механизмам защиты* СЗИ

$$P_{1xm} = (p_1, p_2, ..., p_m),$$

а для матрицы *EE* – вектор из ее диагональных элементов $ee_{ij} = d_i$, $i = j = 1,\ldots, n$, - *вектор достоверности распределения ущерба по эшелонам СЗИ*

$$D_{1xn} = (d_1, d_2, ..., d_n).$$

5. В качестве интегральных оценок защищенности системы ИТ в разрезе механизмов защиты можно использовать рейтинговый показатель R_M - длину m-мерного вектора P_{1xm}

$$R_M = \left|P_{1xm}\right| = \sqrt{\sum_{i=1}^{m} p_i^2}\ ,\ i = 1,..., m, \tag{3}$$

а в разрезе эшелонов СЗИ - рейтинговый показатель R_E - длину n-мерного вектора D_{1xn}

$$R_E = \left|D_{1xn}\right| = \sqrt{\sum_{i=1}^{n} d_i^2}\ ,\ i = 1,..., n. \tag{4}$$

Предельными значениями $R_{M\,lim}$ и $R_{E\,lim}$ рейтинговых показателей являются, соответственно, $m\sqrt{m}$ и $n\sqrt{n}$, где m – число известных МЗ, а n - число эшелонов СЗИ, получаемые при достоверной активации во всех эшелонах многоуровневой СЗИ всех МЗ, предотвращающих по каждому из МЗ нанесение ущерба, равного максимально допустимому.

Текущую эффективность СЗИ целесообразно оценивать в относительных величинах, используя в качестве пороговых значений максимальные значения рейтинговых показателей $R_{M\,max}$ и $R_{E\,max}$, учитывающие достоверной активации во всех эшелонах многоуровневой СЗИ только *активированных* МЗ, предотвращающих по каждому из механизмов защиты нанесение ущерба, равного максимально допустимому

$$\eta_M = \frac{R_M}{R_{M\max}}, \tag{5}$$

$$\eta_E = \frac{R_E}{R_{E\max}}. \tag{6}$$

2.4.3. Оценки информационных ресурсов и безопасности глобальных компьютерных систем

Для оценки уровня информационно-коммуникационных технологий (ИКТ) глобальных компьютерных систем (ГКС) применима система показателей, аналогичная используемой для оценки защищенности систем ИТ. Группа показателей развития информационного общества (ИО) детализируется за счет введения дополнительных экспертных оценок в разрезе иерархии ГКС для различных групп стран и учета материальных затрат, необходимых для эксплуатации и модернизации инфраструктуры ГКС. Формируется группа показателей информационной безопасности, учитывающих распределение механизмов защиты по иерархии ГКС и величину предотвращенного ущерба на заданном поле уг-

роз. Рейтинговые показатели уровня ИКТ получаются путем объединения двух названных групп показателей.

Разработке количественных показателей и методик анализа движения стран мира к ИО уделяется большое внимание [95]. Известны следующие показатели для оценки информационных ресурсов (ИР) различных стран мира: индекс технологической оснащенности (ИТО) [96], индекс прозрачности коммуникаций (ИПК) [97], и индекс информационного общества (ИИО) [98]. Известны также показатели информационной безопасности (ИБ), исходящие из наличия определенного набора средств и механизмов защиты (МЗ), методик изготовления, эксплуатации и тестирования, позволяющих отнести систему информационно-коммуникационных технологий к одному из дискретных уровней защищенности в соответствии с используемыми стандартами [99, 100].

Названные показатели ИР и ИБ не учитывают иерархию ГКС отдельных стран и материальные затраты, необходимые для поддержания инфраструктуры глобальных информационных и коммуникационных систем. В этой связи необходима разработка комплекса показателей оценки информационных ресурсов и информационной безопасности глобальных компьютерных систем для различных групп стран, отличающихся уровнем развития ИТ. Известные показатели ИТО, ИПК, ИИО можно представить в виде матрицы экспертных оценок распределения ИР по странам мира - «страны-ИР» и дополнить экспертными оценками распределения ИР по иерархии ГКС внутри страны (группы стран с аналогичным уровнем развития ИТ) в виде матриц «ИР-иерархия». Аналогично для ГКС показатели ИБ можно представить в виде матрицы достоверности нейтрализации угроз механизмами защиты «МЗ-угрозы» и матрицы достоверности «угрозы-эшелоны».

Показатели информационных ресурсов

Рассмотрим первую группу показателей, связанных с *уровнем развития информационного общества*. Исходные данные – результаты *экспертных оценок* распределения ИР по странам (группам стран) представляют в виде матрицы *CR* «страны-ИР» размерностью *m*x*p*

$$CR_{m\times p} = \begin{pmatrix} cr_{11} & cr_{12} & \vdots & cr_{1p} \\ cr_{21} & cr_{22} & \vdots & cr_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ cr_{m1} & cr_{m2} & \vdots & cr_{mp} \end{pmatrix},$$

$i = 1,..., m$ – число анализируемых стран, $j = 1,..., p$ - число анализируемых показателей ИР. Аналогично формируют матрицу *RH* «ИР-иерархия» распределения ИР по иерархии ГКС стран, относящихся к одной группе по уровню развития ИКТ размерностью *p*x*n*

$$RH_{p\times n} = \begin{pmatrix} rh_{11} & rh_{12} & \vdots & rh_{1n} \\ rh_{21} & rh_{22} & \vdots & rh_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ rh_{p1} & rh_{p2} & \vdots & rh_{pn} \end{pmatrix},$$

i = 1,..., p - число анализируемых показателей ИР, j = 1,..., n - число уровней иерархии в ГКС страны.

Формируют матрицу материальных затрат на эксплуатацию и модернизацию ГКС страны «иерархия-ИР» *HR* размерностью *n*x*p*

$$HR_{n\times p} = \begin{pmatrix} hr_{11} & hr_{12} & \vdots & hr_{1p} \\ hr_{21} & hr_{22} & \vdots & hr_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ hr_{n1} & hr_{n2} & \vdots & hr_{np} \end{pmatrix},$$

i = 1,..., n - число уровней иерархии в ГКС страны, j = 1,..., p - число анализируемых показателей ИР, и матрицу «ИР-страны» *RC* размерностью *p*x*m*

$$RC_{p\times m} = \begin{pmatrix} rc_{11} & rc_{12} & \vdots & rc_{1m} \\ rc_{21} & rc_{22} & \vdots & rc_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ rc_{p1} & rc_{p2} & \vdots & rc_{pm} \end{pmatrix},$$

i = 1,..., p - число анализируемых показателей ИР, j = 1,..., m – число подлежащих анализу стран.

Интегральные показатели получают в результате операций над матрицами. В частности умножение матриц «страны-ИР» *CR* и «ИР-иерархия» *RH* позволяет получить матрицу «страны-иерархия» *CH* размерностью *m*x*n* – матрицу распределения ИР по странам и иерархии ГКС

$$CH_{m\times n} = \begin{pmatrix} ch_{11} & ch_{12} & \vdots & ch_{1n} \\ ch_{21} & ch_{22} & \vdots & ch_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ ch_{m1} & ch_{m2} & \vdots & ch_{mn} \end{pmatrix},$$

i = 1,..., m – число подлежащих анализу стран, j = 1,..., n - число уровней иерархии в ГКС страны, а умножение матриц материальных затрат *HR* и *RC* - матрицу материальных затрат «иерархия-страны» *HC* размерностью *n*x*m,* отражающую распределение затрат на эксплуатацию и модернизацию ГКС

$$HC_{n\times m} = \begin{pmatrix} hc_{11} & hc_{12} & \vdots & hc_{1m} \\ hc_{21} & hc_{22} & \vdots & hc_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ hc_{n1} & hc_{n2} & \vdots & hc_{nm} \end{pmatrix},$$

i = 1,..., n - число уровней иерархии в ГКС страны, j = 1,..., m – число подлежащих анализу стран.

Промежуточные оценки в виде строки и столбца интегральных показателей значимости [93] характеризуют распределение *ИР* по иерархии ГКС и по группе стран, а также позволяют оценить материальные затраты на поддержание информационной инфраструктуры в разрезе стран и иерархии ГКС.

Дальнейшие операции над матрицами *CH* и *HC* дают возможность обобщить в диагональных элементах *итоговой матрицы* как показатель распределения ИР, так и материальных затрат на поддержание информационной инфраструктуры стран.

Умножением матрицы распределения ресурсов *CH* и материальных затрат *HC* получают квадратную *матрицу* затрат на поддержание информационной инфраструктуры ГКС при данном распределения ИР – матрицу «страны-страны» *CC*

$$CC_{m\times m} = \left(\begin{array}{cc:c} cc_{11} & cc_{12} & cc_{1m} \\ cc_{21} & cc_{22} & cc_{2m} \\ \hdashline cc_{m1} & cc_{m2} & cc_{mm} \end{array}\right),$$

$i = j = 1,..., m$ – число анализируемых стран, а умножением матрицы *HC* и матрицы *CH* получают квадратную матрицу затрат на поддержание информационной инфраструктуры ГКС при данном распределения ИР – матрицу «иерархия-иерархия» *HH*

$$HH_{n\times n} = \left(\begin{array}{cc:c} hh_{11} & hh_{12} & hh_{1n} \\ hh_{21} & hh_{22} & hh_{2n} \\ \hdashline hh_{n1} & hh_{n2} & hh_{nn} \end{array}\right),$$

$i = j = 1,..., n$ - число уровней иерархии в ГКС страны.

Для матрицы *CC* в качестве обобщающего показателя можно рассматривать вектор, образованный диагональными элементами $cc_{ij} = c_i$, $i = j = 1,\ldots, m$, матрицы - вектор затрат на поддержание информационной инфраструктуры ГКС при данном распределения ИР

$$C_{1xm} = (c_1, c_2, ..., c_m),$$

а для матрицы *HH* – вектор из ее диагональных элементов $hh_{ij} = h_i$, $i = j = 1,\ldots, n$, - вектор затрат на поддержание информационной инфраструктуры ГКС при данном распределения ИР

$$H_{1xn} = (h_1, h_2, ..., h_n).$$

В качестве интегральных показателей информационной оснащенности в разрезе группы стран можно использовать рейтинговый показатель R_C - длину m-мерного вектора C_{1xm}

$$R_C = |C_{1xm}| = \sqrt{\sum_{i=1}^{m} c_i^2}\ ,\ i = 1,..., m, \qquad (7)$$

а в разрезе иерархии ГКС - рейтинговый показатель R_H - длину n-мерного вектора H_{1xn}

$$R_H = |H_{1xn}| = \sqrt{\sum_{i=1}^{n} h_i^2}\ ,\ i = 1,..., n. \qquad (8)$$

Выводы по главе 2

Рассмотрены иерархическая модель и метод проектирования адаптивной системы защиты информации, основанные на принципах биологического подобия. Модель адаптивной защиты использует принцип биосистемной аналогии, в частности, иерархию системы защиты информационных процессов и ресурсов в биологической системе, согласно которой на нижних уровнях иерархии задействованы механизмы иммунной системы, а на верхних - механизмы адаптивной памяти и накопления жизненного опыта нервной системы.

1. Показано, что построение безопасных систем ИТ основано на иерархической организации информационной защиты, а также:

- биосистемной аналогии в архитектуре систем ИТ,
- известных механизмах информационной защиты биосистем,
- наличии иерархии уровней информационной защиты систем ИТ,
- свойствах НС, необходимых для реализации функций информационной защиты.

2. Предложен метод проектирования адаптивной СЗИ, включающий следующие этапы:

1) формирование матриц *адаптируемых экспертных оценок* и на их основе создание исходных *систем нечетких правил* и *классификаторов* (на нижних уровнях защиты - классификаторов «признаки атаки - угрозы», на верхних уровнях защиты – классификаторов «угрозы - МЗ»);

2) *идентификация* выявленной *угрозы* и при расширении поля известных угроз - *кластеризация угроз* с последующей адаптацией информационных полей путем обучения НС уровней защиты;

3) *кластеризация* вследствие изменения поля угроз сопровождается *коррекцией* или *расширением системы нечетких правил*;

4) *изменение поля угроз* вызывает *модификацию систем нечетких правил* и матриц *экспертных оценок* в результате обучения классификаторов уровней защиты;

5) при расширении системы нечетких правил формируется *описание* нового (отсутствующего) *механизма защиты*;

6) «прозрачность» системы нечетких правил позволяет сформулировать *спецификацию на создание* отсутствующего МЗ;

7) на основании анализа комплекса оценок защищенности ИТ-системы (в случае экономической целесообразности) включают новый МЗ в состав защиты.

3. Отмечено, что при проектировании адаптивной системы защиты информации следует учитывать комплексный характер модели, связующим звеном которой является методика оценки защищенности системы ИТ, которая координирует взаимосвязь классификаторов угроз и механизмов защиты (в виде НС, нечетких НС, систем нечетких предикатных правил), структурной модели системы информационной безопасности, инструментальных средств расчета показателей защищенности и рейтинга системы ИТ.

Вначале выбирается структурная модель СИБ в виде иерархии уровней механизмов защиты, а априорный опыт экспертов представляется массивами экспертных оценок, на базе которых формируются системы нечетких предикатных правил для классификации 1) угроз по признакам атак и 2) МЗ на поле угроз. Системы нечетких предикатных правил для последующей адаптации и анализа представляются в виде нечетких НС, которые обучают на некотором подмножестве входных векторов признаков атаки. Одновременно обучают классификаторы в виде обычных НС таким образом, чтобы число образуемых кластеров равнялось числу правил в системе нечетких предикатных правил. Аналогично обучают нейросетевые классификаторы механизмов защиты по векторам известных угроз.

Для исходных массивов экспертных оценок производят расчет показателей защищенности и рейтинга системы ИТ, которые используются методикой оценки защищенности системы ИТ для анализа и коррекции, как массивов экспертных оценок, так и функциональных параметров нейросетевых классификаторов и систем нечетких предикатных правил.

4. В качестве основных элементов адаптивной модели СЗИ разработаны методика и комплекс показателей для оценки уровня защищенности системы ИТ, учитывающие *достоверность* нейтрализации угроз, а также *потенциальный ущерб* и частоту активации угроз.

Показана возможность применения разработанного комплекса показателей защищенности для оценки уровня развития информационных технологий глобальных компьютерных систем ГКС. Детализирована группа показателей развития информационного общества за счет введения дополнительных эксперт-

ных оценок в разрезе иерархии ГКС для различных групп стран и учета материальных затрат, необходимых для эксплуатации и модернизации инфраструктуры ГКС. Сформирована группа показателей информационной безопасности, учитывающих распределение механизмов защиты по иерархии ГКС и величину предотвращенного ущерба на заданном поле угроз. Получены рейтинговые показатели уровня ГКС путем объединения двух названных групп показателей.

Предложено, в качестве основных механизмов реализации *адаптивных свойств* СЗИ использовать:

- способность нечеткого распределенного информационного поля нейронной сети к *накоплению знаний* в процессе обучения;
- *нечеткий логический вывод*, который позволяет использовать опыт экспертов в области информационной безопасности, овеществленный в виде системы нечетких предикатных правил, для *предварительного обучения* нейро-нечеткой сети. Возможность отображения системы нечетких предикатных правил на структуру СЗИ и последующее ее *обучение* на поле известных угроз позволяют проанализировать процесс логического вывода для уточнения существующей или синтеза новой системы нечетких предикатных правил СЗИ;
- способность нейронных и нейро-нечетких сетей к *классификации и кластеризации*.

Показано, каким образом адаптивная модель отражает развитие системы информационной безопасности в процессе жизненного цикла систем ИТ.

3. АСПЕКТЫ ОРГАНИЗАЦИИ АДАПТИВНЫХ СИСТЕМ ЗАЩИТЫ ИНФОРМАЦИИ

Реализация адаптивных СЗИ базируется на принципах подобия архитектуры и механизмов защиты системы ИТ архитектуре и механизмам защиты биологических систем. Принципы подобия нашли отражение в модели адаптивной СЗИ, рассмотренной в предыдущей главе. В настоящей главе рассматриваются вопросы реализации и инструментальные средства для моделирования адаптивных СЗИ.

В качестве базы для построения адаптивных СЗИ может быть использован технический аналог ткани биологической системы, представленный в виде взаимосвязанных интерфейсом нейросетевых командных пулов, управляемых потоком данных. В соответствии с принципами монолитности исполнения и многофункциональности памяти обработку данных следует проводить непо-

средственно в локальных пулах команд путем выполнения последовательных операций чтения - модификации – записи.

Определенной альтернативой монолитности исполнения можно считать секционирование, которое позволяет, используя базовые блоки (секции) в качестве элементов структуры, программно формировать нейросетевую СЗИ в соответствии с предъявляемыми требованиями. Секционирование позволяет усложнить операционную зону и использовать параллельную арифметику для реализации основных операций нейросетевого базиса [5, 103]. Секционирование не противоречит принципу монолитности особенно при реализации нейросетевого вычислителя по технологии «нейрокомпьютер на кремневых пластинах» [5]. С другой стороны, секционирование позволяет воплотить архитектурные решения базовых блоков с учетом возможностей отечественной микроэлектронной промышленности в виде наращиваемых СБИС.

Задачи, подлежащие решению в системах защиты информации, можно подразделять на формализуемые и неформализуемые. Первый класс задач как более широкий и исследованный реализуется с помощью программных средств на универсальных машинах. Однако традиционный подход к управлению вычислениями критикуется из-за последовательного характера вычислительного процесса [109, 110].

Заслуживает внимания метод решения формализуемых задач, в котором управление вычислительным процессом осуществляется с помощью потока данных – УПД [104, 111]. УПД отказывается от принудительного задания порядка выполнения машинных операций. Неформализуемые задачи – область применения нейросетевых методов, где иное управление вычислениями не приемлемо из-за невозможности алгоритмического описания хода вычислительного процесса.

Программно настраиваемый нейросетевой командный пул способен решать оба класса задач, представленных в виде пакетных нейросетевых программ [4, 106].

Адаптивность СЗИ обеспечивается использованием элементной базы, способной к обучению, и, прежде всего, нейронных сетей. Нейросетевые СЗИ согласно принципу биосистемной аналогии следует представлять в виде структурированных информационных полей иммунного и рецептороного уровней защиты.

3.1. Разработка алгоритма адаптации нейросетевых СЗИ

Для представления процессов адаптации нейросетевых систем защиты информации удобно использовать язык графического описания объектов, подобный предложенному Дж. Деннисом и Д. Мисунасом [112 - 114].

3.1.1. Описание нейросетевых СЗИ

Описание нейросетевых СЗИ на графическом языке УПД сведется к воспроизведению одной из стандартных топологий, где в качестве исполнительных элементов могут быть использованы либо ФН, либо слой из формальных нейронов. Программы потоков данных, согласно [114, 115], могут быть представлены в форме последовательности операторов, подчиняющихся определенному синтаксису языка, либо в виде функционально завершенной совокупности КП, размещаемых в командных ячейках пула команд.

Определим основные понятия, которые использованы ниже по тексту.

Пул команд – многофункциональная безадресная память для размещения ПНП; получает пакеты данных; формирует командные пакеты или пакеты данных.

Пакетная нейросетевая программа – функционально завершенная совокупность взаимосвязанных командных пакетов.

Командный пакет - структурный компонент ПНП, образованный совокупностью специализированных полей и задающий, как операцию нейросетевого базиса, так и номера командных пакетов-приемников результата.

Пакет данных (ПД) – средство доставки (контейнер) значений данных от одного КП (источника) к другому КП (приемнику результата).

Командная ячейка – часть пула команд для размещения КП.

Нейросетевой базис включает в себя функции и компоненты, которые рассматривают как язык представления НС [107]. Каждому из компонентов базиса ставят в соответствие КП, из которых формируют функционально полные наборы КП и используют в качестве элементарных программных и структурных единиц [116, 117].

Для иллюстрации на рис. 3.1 представлена межнейронная связь (синапс), выполняющая операцию взвешивания входного сигнала.

Синапсу соответствует КП, состоящий из поля приемника результата (коммуникационное поле) D, поля функциональных параметров W_1 (помечено вентильным кодом C - const), поля входного порта X_1 и поля готовности данных R_1 (оба с вентильным кодом N - not).

Иногда функцию взвешивания сигнала передают репликатору и получают выходную звезду (рис. 3.2), КП которой содержит поля весов, например, $\alpha_1,\ldots$, α_n. Выходная звезда соответствует отдельной нечеткой связи, если нечеткое множество $\{\alpha_1,\ldots, \alpha_n\}$ соответствует некоторой семантике СД.

Формальный нейрон (рис. 3.3) получается последовательным соединением адаптивного сумматора, нелинейного преобразователя и нечеткой связи. КП формального нейрона содержит полный набор вышеназванных полей обра-

зующих его структурных компонентов. Функциональная универсальность позволяют рассматривать ФН в качестве базового элемента ПНП.

Слой ФН (рис. 3.4) - следующий уровень абстрагирования. Роль элемента структуры НС играет ранг идентичных по функциональным возможностям ФН. КП слоя ФН в отличие от КП отдельного нейрона содержит матрицу параметров W_{ij} вместо вектора весов и порога срабатывания ФН W0 ($0 \le i \le r;\ 0 \le j \le n$; где r - число ФН в слое НС, n - число входов отдельного ФН).

Для описания НС с помощью ПНП необходимо, чтобы набор командных пакетов удовлетворял требованиям функциональной полноты. Если в качестве единственного базового элемента выбрать ФН, то ПНП будет представляться ограниченной совокупностью (по числу ФН НС) однотипных КП, различающихся только содержимым коммуникационных (связи) и функциональных (веса) полей. Если же в качестве базового элемента выбрать слой формальных нейронов, то ПНП будет более компактной, а следовательно снизятся затраты времени, связанные с транспортировкой готовых КП из пула команд к PU и ПД в обратном направлении.

Для решения неформализуемых задач может быть использован стандартный нейросетевой подход: в зависимости от типа задачи выбирается одна из известных сетевых конфигураций, соответствующая ей парадигма обучения НС, а в качестве базового элемента – ФН, представленный командным пакетом. Информация о межнейронных связях сети записывается в коммуникационные поля командного пакета, а параметры НС, полученные в результате обучения - в функциональные поля той же совокупности КП.

Формализуемые задачи могут быть описаны на графическом языке, где в качестве исполнительных элементов, информации и связей будут использоваться, соответственно, блоки нейросетевого базиса, токены данных и управляющие токены (в формализуемых задачах появятся условные вершины), а также сигнальные линии для передачи значений данных и управляющей информации в виде пакетов данных [111].

Представление формализуемых задач в виде ПНП потребует использования специальных КП для описания условных вершин реализуемого алгоритма (рис. 3.5). В отличие от фон-неймановских машин, в которых ветвление в алгоритме организуется модификацией содержимого счетчика команд, изменяющего порядок выборки команд из памяти, в подходе УПД, где отсутствует заранее обусловленный порядок выборки и обработки КП, необходимо управляемо перенаправлять потоки данных. С этой целью можно либо отдельным полям КП придать вентильные свойства и управлять этими полями посредством управляющих токенов [111], либо в состав набора КП ввести управляющий КП (рис. 3.5).

*	D
C	W_1
N	X_1
N	R_1

$X_1 \xrightarrow{W_1} X_1 W_1$

Рис. 3.1. Синапс

Выходная звезда

Str	D_1	D_2	···	D_n
C	α_1	α_2	···	α_n
N	X			
N	R			

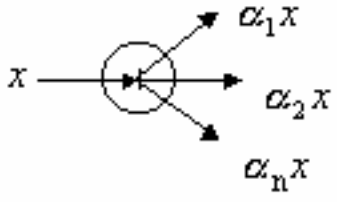

Рис. 3.2. Выходная звезда

FN	D_1	D_2	···	D_m
C	W_0	W_1	···	W_n
N	X_1	X_2	···	X_n
N	R_1	R_2	···	R_n

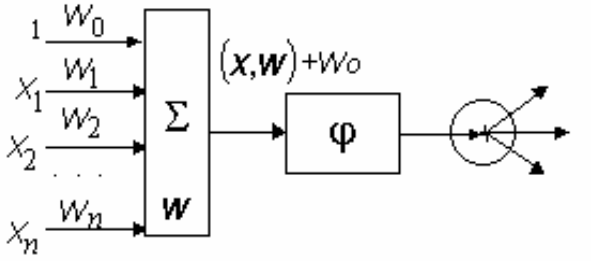

Рис. 3.3. Формальный нейрон

LN	D_1	D_2	···	D_m
C	W_{10}	W_{11}	···	W_{1n}
C	⋮	W_{ij}	···	
C	W_{r0}	W_{r1}	···	W_{rn}
N	X_1	X_2	···	X_n
N	R_1	R_2	···	R_n

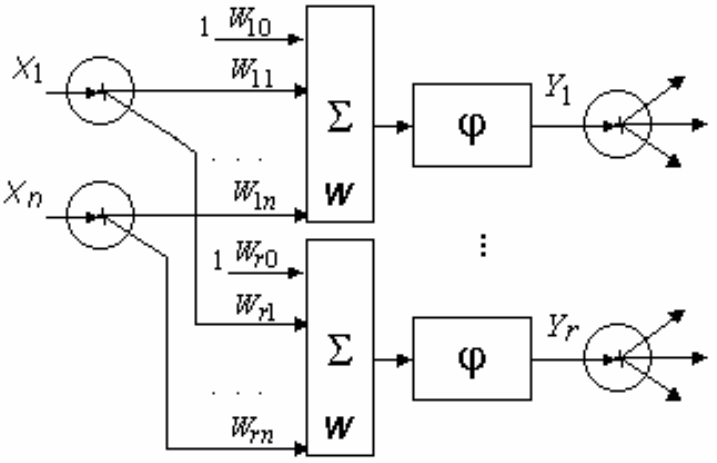

Рис. 3.4. Слой формальных нейронов

Boolean		DT_1	···	DT_n
		DF_1	···	DF_n
N	X_1	X_2		X_3
N	R_1	R_2		R_3
C	ε			

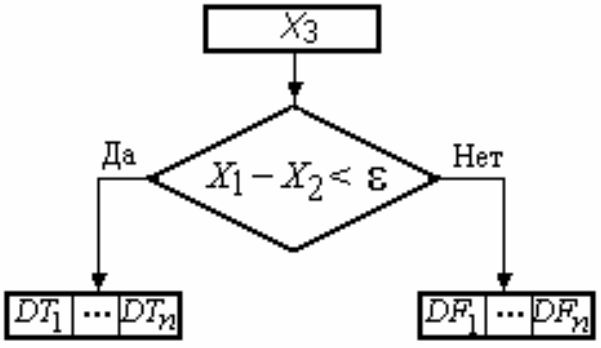

Рис. 3.5. Управляющий командный пакет

Управляющий КП будет фиксировать предназначенные ему данные в своих входных портах (X_1, X_2, X_3), устанавливая соответствующие биты готовности в полях (R_1, R_2, R_3).

При поступлении операндов КП обрабатывается таким образом, что часть данных используется для формирования результата отношения Boolean (к примеру, $X_1 - X_2 < \varepsilon$), а другая часть поступивших данных (X_3) в качестве значения будет передана либо адресатам $DT_1, DT_2, \ldots, DT_n$, либо адресатам $DF_1, DF_2, \ldots, DF_n$ в зависимости от результата отношения TRUE или FALSE, соответственно.

При разработке языковых средств для описания нейросетевых СЗИ следует руководствоваться следующими положениями. 1) в командных пулах информация, представленная в виде пакетов, преобразуется и перемещается в соответствии с графом обработки информации. 2) нейросетевая СЗИ представляется в виде размещенной в пуле команд ограниченной функционально завершенной совокупности КП, образующих ПНП. 3) КП - функционально-структурная единица языка описания НС. 4) Система КП должна удовлетворять требованиям функциональной полноты, т.е. набор операций, реализуемых КП системы, должен составлять нейросетевой базис. 5) КП в зависимости от реализуемой функции может соответствовать: a) отдельным элементам ФН или комбинации элементов; b) отдельному ФН; c) части слоя, содержащего ФН; d) отдельному слою ФН; e) НС. 6) КП может быть операционным, т. е. реализующим функцию нейросетевого базиса, или управляющим, переключающим потоки данных в НС. 7) нейросетевая СЗИ формируется путем заполнения командных, коммуникационных и функциональных полей КП либо на этапе обучения, либо, в случае обученной сети, занесением готовой к выполнению ПНП в пул команд. 8) Данные представляют собой контейнер или квант информации, дополненный коммуникационной и служебной информацией. 9) НС самоуправляется механизмом готовности данных и передачей результатов обработки КП-источников в операндные поля (входные порты) КП- приемников с помощью пакетов данных. 10) нейросетевая СЗИ начинает функционировать после занесения данных в операндные поля КП и срабатывания механизма готовности: КП извлекается из пула команд после поступления операндов во входные порты в количестве, достаточном для корректного выполнения операции.

Для обеспечения информационной защищенности и сохранения работоспособности при разрушении части НС целесообразно использовать механизм частичной готовности. КП извлекается из пула команд в случаях: 1) заполнения всех операндных полей; 2) поступления заданной совокупности операндов; 3) поступления определенного количества операндов; 4) истечения допустимого времени нахождения КП в пуле команд.

Использование механизма частичной готовности данных при реализации нейросетевых приложений в виде ПНП позволяет задействовать механизм нейросетевой избыточности и повысить информационную защищенность и функциональную устойчивость НС. В пакетах, играющих роль транспорта для передачи сообщений, в основном содержится коммуникационная информация и значения данных, к искажению которых НС в достаточной мере устойчива.

Для обучения нейросетевых СЗИ, построенных на базе логарифмической структурной модели формального нейрона, предложен алгоритм по методу обратного распространения ошибки [119].

3.1.2. Алгоритм логарифмического обратного распространения ошибки

Метод обратного распространения ошибки при вычислении поправок к весам НС многократно использует операцию умножения. В логарифмической модели формального нейрона умножение в процессе взвешивания заменено суммированием.

В данной модели ФН использованы два нелинейных преобразователя (рис. 3.6). Первый из них $\varphi = a \ln bx$, $a<1$, $b>1$, $x>0$ размещен на синаптических входах ФН (соответствующих выходах предыдущего слоя сети) и выполняет функцию масштабирования сигналов в НС. Второй преобразователь реализует функцию $\psi^{-1}(x) = p\, e^{\,mx}$, $p<1, m>1,$ для потенцирования значений логарифма взвешенных сигналов перед их суммированием в теле ФН и реализации дополнительного нелинейного преобразования $\eta(x) = k\, x^{ma}$, $k = p\, b^{\,ma}$, $ma<1$. То есть роль функции активации в структуре ФН модели *B* играет степенная функция $\eta(x)$, неявно реализуемая в синапсах при суперпозиции функций $\psi^{-1}(\varphi(x))$ и соответствующем подборе значений коэффициентов *a, m, p*. Выходом ФН является выход сумматора *Σ*, на котором реализуется функция $q = \sum_i s_i$, где s_i - *i*-й синаптический вход тела ФН, $1 \le i \le n$.

Обратный логарифмический просчет начинается с вычисления значения ошибки $\delta = y - q$, где *y* – целевое значение одной из координат выходного вектора $\boldsymbol{Y}_u$, заданное на множестве обучающих пар $\{(\boldsymbol{X}_u, \boldsymbol{Y}_u)\}$, *u = 1, 2 ,..., M*, а *q* – реальное значение сигнала соответствующей координаты выходного вектора на выходе данного ФН при подаче на его входы вектора $\boldsymbol{X}_u$. Пусть значение ошибки δ распределено по входам тела ФН пропорционально величинам взвешенных входных сигналов s_i, то есть $\delta_i = {\delta \cdot s_i}/{q}$, где δ_i - ошибка, приведенная к *i*-у синаптическому входу. Для исключения ошибки по *i*-у входу необходимо скор-

ректировать сигнал *i*-го синапса на величину δ_i, $s_i' = s_i + \delta_i = s_i + \frac{\delta \cdot s_i}{q} = \frac{y \cdot s_i}{q}$. Т. е. s_i' получается путем умножения исходного значения s_i на величину y/q.

Значение сигнала *i*-го синапса s_i при подаче на вход ФН сигнала x_i определяется прямым просчетом распространения сигналов в структуре ФН. Сигнал x_i одновременно является выходным сигналом q_j нейрона *j* предыдущего слоя НС, где он подвергается преобразованию $\varphi = a \ln bx$. Поэтому на репликатор рассматриваемого ФН поступает значение логарифма входного сигнала $X_i = a \ln b x_i$. Двухвходовой сумматор *i*-го синапса выполняет сложение значений $X_i = a \ln b x_i$ и $W_i = a \ln b w_i$. Причем значение веса может храниться в локальной памяти ФН в виде логарифма. Результат – значение логарифма взвешенного сигнала $S_i = X_i + W_i$ подвергается преобразованию $\psi^{-1}(x)$ в значение сигнала *i*-го синапса s_i: $s_i = p\, e^{m(X_i + W_i)} = p\, e^{m(a \ln(b x_i \cdot b w_i))} = p\left(b x_i \cdot b w_i\right)^{ma}$.

Корректное значение логарифма взвешенного сигнала *i*-го синапса равно $S_i' = \frac{1}{m}\ln\left(\frac{1}{p} s_i'\right) = \frac{1}{m}\ln\left(\frac{s_i\, y}{p\, q}\right) = \frac{1}{m}\ln\left(\frac{y}{pq}\, p\left(bx_i \cdot bw_i\right)^{ma}\right) = a \ln bx_i + a \ln bw_i + \frac{1}{m}\ln\frac{y}{q} = X_i + W_i + \frac{\Delta}{ma}$.

Так как $S_i' = S_i + \Delta_i$, то $\Delta_i = \frac{\Delta}{ma}$. Последнее соотношение показывает, что все значения логарифмов синаптических сигналов данного ФН получают одинаковое приращение $\Delta_i = \frac{\Delta}{ma}$ *вне зависимости* от значений входных сигналов, где Δ - разность значений логарифмов целевого и реального выходных сигналов ФН: $\Delta = Y - Q = a \ln\left(\frac{y}{q}\right)$.

Значение логарифма приращения синаптических сигналов ФН Δ_i в *ma* раз ($m>1$, $a<1$) меньше разности Δ значений логарифмов целевого и реального выходных сигналов нейрона. Пусть значение логарифма ошибки синаптического сигнала поровну распределено по входам сумматора синаптического взвешивания, т. е. $\Delta W_i = \Delta X_i = \frac{\Delta_i}{2} = \frac{\Delta}{2ma}$. Подобное распределение позволяет технически просто (за счет операции сдвига) осуществить вычисление приращений и коррекцию, как логарифмов значений синаптических сигналов, так и логарифмов значений весов в процессе обучения. Возможны иные подходы к распределению сигнала ошибки в синапсе между значениями входного сигнала и веса.

Значения логарифмов синаптических сигналов S', входных сигналов X' и весов W' могут быть связаны соотношением: $S_i' = X_i' + W_i' =$

$= X_i + \frac{\Delta}{2ma} + W_i + \frac{\Delta}{2ma}$. Для простоты расчетов значения $\frac{\Delta}{2ma}$ величину *ma* следует выбирать кратной степени 2. В этом случае величины ΔW_i и ΔX_i несложно получить из Δ выполнением операции сдвига.

Для реализации процедуры обратного просчета в каждом синапсе модели *B* ФН необходимы два функциональных преобразователя: преобразователь $\varphi(x)$ на входе ФН x_i и преобразователь $\psi^{-1}(x)$ на соответствующем входе тела ФН. Преобразователи реализуют комплементарную пару функций – в рассматриваемом случае $\varphi(x) = a \ln bx$, $a<1$, $b>1$, $x>0$ и i , $1 \le i \le m$, , таких, что значения их коэффициентов при суперпозиции функций $\psi^{-1}(\varphi(x))$ позволяют неявно реализовать в синаптических связях функцию активации $\eta(x) = k\, x^{ma}$, $k = p\, b^{ma}$, $ma<1$.

Отметим следующие этапы в алгоритме обучения НС по процедуре логарифмического обратного распространение ошибки:

1. При начальной инициализации НС весам и смещениям присвоить малые случайные величины из диапазона изменения значений логарифмов.

2. Выбрать очередную обучающую пару из множества обучающих пар $\{(\boldsymbol{X}_u, \boldsymbol{Y}_u)\}$, $u = 1, 2, ..., M$; подать логарифм значений входного вектора $\boldsymbol{X}_u$ на вход НС.

3. Вычислить логарифм значений выходного вектора $\boldsymbol{Q}_u$ нейронной сети.

4. Вычислить разность между логарифмами значений одноименных координат вектора $\boldsymbol{Q}_u$ сети и целевого вектора $\boldsymbol{Y}_u$ из обучающей пары векторов ($\boldsymbol{X}_u$, $\boldsymbol{Y}_u$), то есть определить значение логарифма координат вектора ошибки $\boldsymbol{\Delta}_u$, $u = 1, 2, ..., M$.

5. Для настройки весов используется процедура логарифмического обратного просчета: логарифмы весов НС корректируются на множестве обучающих пар $\{(\boldsymbol{X}_u, \boldsymbol{Y}_u)\}$путем прибавления к логарифмам функциональных параметров каждого *k*-го слоя НС ($1 \le k \le l$, l – число слоев сети) поправок Δ_k таких, чтобы для каждой координаты вектора $\boldsymbol{\Delta}_u$ значение $\sum_k \Delta_k$ равнялась значению данной координаты вектора $\boldsymbol{\Delta}_u$. Алгоритм циклически выполняется на множестве обучающих пар $\{(\boldsymbol{X}_u, \boldsymbol{Y}_u)\}$до достижения необходимой степени близости векторов $\boldsymbol{Q}_u$ к $\boldsymbol{Y}_u$.

Шаги 2 и 3 соответствуют прямому просчету (распространение сигналов от входов к выходам нейронной сети), а шаги 4 и 5 – обратному просчету: логарифм значения сигнала ошибки передается по НС в обратном направлении и используется для подстройки значений логарифмов функциональных параметров ФН.

Прямой просчет НС выполняется послойно, начиная с входного слоя, куда поступает вектор $\boldsymbol{X}_u$ из множества обучающих пар $\{(\boldsymbol{X}_u, \boldsymbol{Y}_u)\}$. Входные преобразователи $\varphi(x)$ формируют значения логарифмов координат вектора входных сигналов, а двухвходовые сумматоры каждого i-го синапса ($1 \le i \le r$, r – число входов нейрона) – соответствующие суммы значений $X_i = a \ln b\, x_i$ и $W_i = a \ln b\, w_i$. Значения логарифмов взвешенных сигналов $S_i = X_i + W_i$ подвергаются преобразованию $\psi^{-1}(x)$ в значения сигналов синапсов s_i, причем функция активации $\eta(x) = k\, x^{ma}$ «сжимает» каждый из сигналов s_i, а величина q каждого ФН в слое вычисляется как взвешенная сумма его входов. Выходной вектор входного слоя является входным для следующего слоя. Процесс повторяется послойно, пока не будет получен выходной вектор НС - один из векторов $\boldsymbol{Q}_u$.

Подстройка весов выходного слоя. Для каждого из векторов $\boldsymbol{X}_u$ из множества обучающих пар $\{(\boldsymbol{X}_u, \boldsymbol{Y}_u)\}$, $u = 1, 2, ..., M$ заданы значения целевого вектора $\boldsymbol{Y}_u$. Если обеспечена необходимая степень близости целевого вектора и полученного в результате прямого просчета выходного вектора $\boldsymbol{Q}_u$, то НС обучена. В противном случае необходимо обучение НС. На рис. 6 показан фрагмент взаимодействия оперативных данных с информационным полем НС, соответствующей модели B формального нейрона, при обучении межнейронных связей между скрытым j-м и выходным k-м слоями нейронной сети.

Для подстройки весов выходного слоя НС вычисляются значения логарифмов координат выходного вектора $Q_i^k, 1 \le i \le m$, соответствующего целевого вектора $Y_i^k, 1 \le i \le m$ и определяются значения координат вектора логарифмической ошибки Δ_i^k, k – номер выходного слоя, а m – размерность выходного вектора НС. Для каждой i-й координаты вектора ошибки (выход i-го ФН выходного слоя НС): $\Delta_i^k = Y_i^k - Q_i^k = a \ln \dfrac{y_i}{q_i}, 1 \le i \le m$.

На рис. 3.7 описанным преобразованиям соответствуют вершины $v_1 - v_4$: вершины v_1, v_2, v_4 сопоставлены с вершинами ОД, содержащими в полях данных значения логарифмов координат выходного, целевого векторов и вектора логарифмической ошибки, а в коммутационных полях – d_i^k - указатель на i-й нейрон k-го слоя в качестве источника ошибки. ОВ v_3 задают операцию над вершинами ОД – вычитание. Вершины $v_4 - v_6$ описывают процесс выбора минимальной из координат вектора ошибки k-го слоя $\Delta_{\min}^k$ (вершина v_5) и фор-

мирования вершин ОД (вершины v_6), содержащих исходные данные для коррекции функциональных параметров выходного слоя НС.

Процесс коррекции синаптической силы межнейронных связей детализирован на рис. 3.7 для *i*-го ФН выходного слоя (вершины $v_6 - v_{10}$). В ОВ v_7 формируется значение логарифма приращения для всех весов *i*-го ФН *k*-го слоя НС $\Delta W^k{}_i = \Delta_i^k - \frac{\Delta_{\min}^k}{2}$ и помещается в одно из полей данных вершины ОД v_8. В остальные поля данных вершины v_8 заносятся значения логарифмов функциональных параметров, считанные из локальной памяти весов (*Weights Memory*) *i*-го ФН *k*-го слоя в соответствии с коммутационным полем d_i^k вершины v_6. Непосредственно коррекция весов межнейронных связей осуществляется в ОВ v_9 путем сложения логарифмов предыдущих значений весов и соответствующих приращений $W_i^k{}' = W_i^k + \Delta W_i^k = W_i^k + \Delta_i^k - \frac{\Delta_{\min}^k}{2}$, а новые значения логарифмов весов фиксируются в полях данных вершины ОД v_{10}. Сохранение откорректированных значений производится в долговременной памяти весов (информационном поле нейросетевой СЗИ) в соответствии с коммуникационным полем d_i^k вершины ОД v_{10}.

По оговоренному выше порядку распределения значения логарифмической ошибки $\Delta W^k{}_i = \Delta_i^k - \frac{\Delta_{\min}^k}{2}$ в качестве логарифма приращений для проведения коррекции в скрытых слоях НС, в частности следующего, *j*-го слоя, выбирается значение $\Delta X^j{}_i = \frac{\Delta_{\min}^k}{2}$ единое для всех i, $1 \le i \le m$, где *m* – число ФН в *j*-м слое НС. Формирование $\Delta X^j{}_l$ описывается вершинами v_{11}, v_{12}: ОВ v_{11} соответствует операции масштабирования приращения $y = kx$, $k = 1/2$; вершина ОД v_{12} в поле данных содержит вновь сформированное значение логарифма приращения для очередного, *j*-го слоя НС, а в коммуникационных полях – указание на приемники логарифма приращения $t_p^j, 1 \le p \le n$, где *n* – число ФН в *j*-м слое НС.

Подстройка весов скрытого слоя. Так как из слоя k на все ФН скрытого j-го слоя подается одно и то же значение логарифма приращений (ошибки) $\Delta X^j{}_l = \frac{\Delta_{\min}^k}{2}$, то упрощается дальнейший обратный просчет.

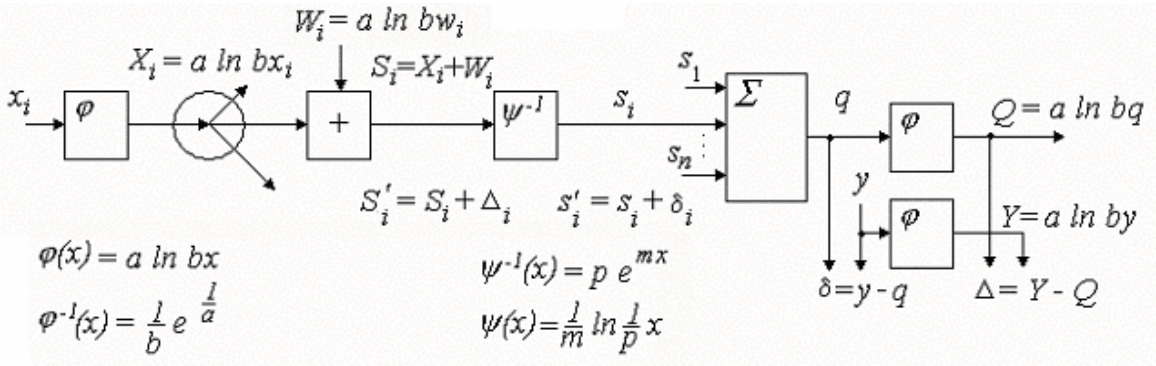

Рис. 3.6. Структурная модель логарифмического ФН

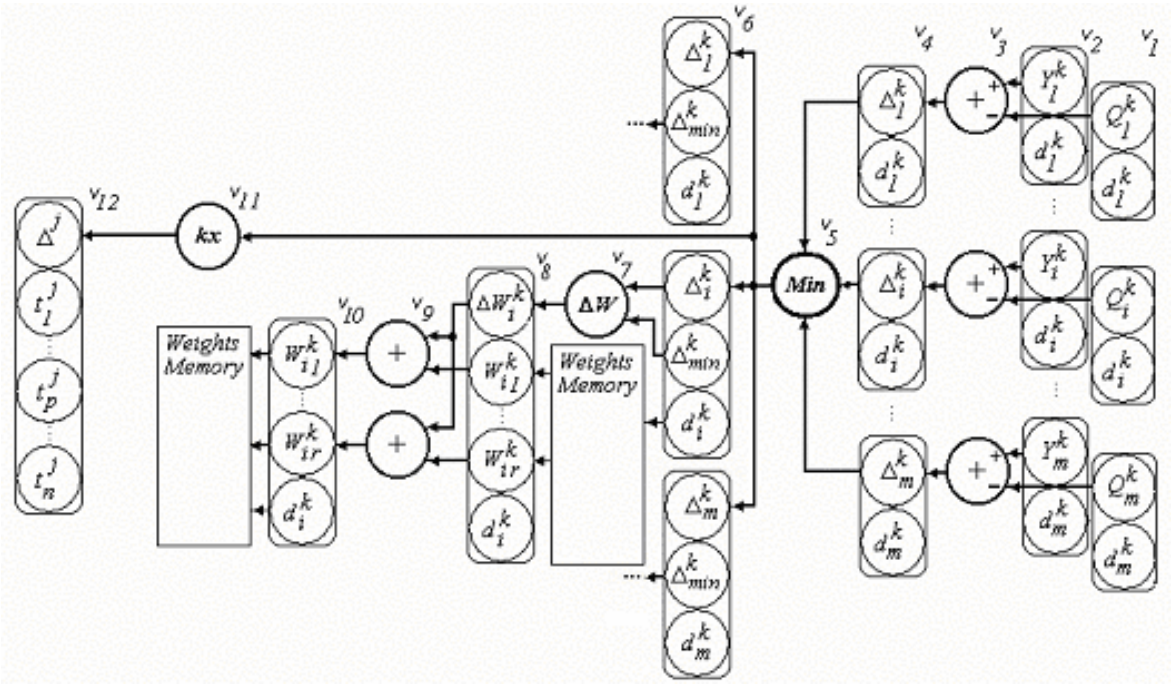

Рис. 3.7. Фрагмент ПНП для обучения логарифмической НС

Передаваемое к следующему (j-1)-у скрытому слою значение логарифма приращения будет меньше входного $\Delta X^{j}{}_{l}$ вдвое. То есть $\Delta X^{j-1}{}_{i} = \frac{\Delta_i^j}{2} = \frac{\Delta_{\min}^k}{4}$ для всех i, $1 \le i \le m$, где m – число ФН в (j-1)-м слое НС, а $\Delta W^{j}{}_{i} = \Delta_i^j - \frac{\Delta_{\min}^k}{4} = \frac{\Delta_{\min}^k}{4}$ для всех $i, 1 \le i \le m$, m – число ФН в j-м слое нейронной сети.

Таким образом, в каждом последующем скрытом слое значение логарифма ошибки уменьшается вдвое, а значение логарифма приращения функциональных параметров для всех ФН слоя составляет половину от значения логарифма поступившей в данный слой ошибки. Для (j-v)-го скрытого слоя будет справедливо соотношение:

$$\Delta X^{j-v}{}_{l} = \frac{\Delta_l^{j-v+1}}{2} = \frac{\Delta_{\min}^k}{2^{v+1}}.$$

Подстройка весов входного слоя. Оставшееся после коррекции функциональных параметров первого из скрытых слоев значение логарифма ошибки полностью суммируется с логарифмами весов всех ФН входного слоя обучаемой НС.

Рассмотренная процедура повторяется для всех пар векторов обучающей выборки до тех пор, пока не будет достигнута заданная степень близости всех пар одноименных векторов ***Q*** к ***Y***.

3.1.3. Оценки времени обучения по методам логарифмического и обратного распространения ошибки

Алгоритм обратного распространения ошибки. Подстройка весов ФН *выходного слоя НС.* Если входной сигнал *i*-го входа ФН обозначим x_i, формируемый выходной сигнал – q, а целевой выходной сигнал – y, то при использовании дельта-правила приведенная к входу ошибка составит величину $\delta = q(1 - q)(y - q)$. Затем δ умножается на величину x_i и на коэффициент скорости обучения η, а результат прибавляется к значению веса связи.

$$\Delta w_i = \eta\, x_i\, q(1 - q)(y - q) = \eta\, x_i\, \delta.$$
$$w_{pq,k}(n+1) = w_{pq,k}(n) + \Delta w_{pq,k}$$

Т. е. вычисляется производная функции активации - в случае сигмоидальной функции выполняются операции вычитания и умножения $q(1 - q)$, определяется ошибка выхода $(y - q)$, и полученные значения умножаются на скорость обучения

$\Delta = \eta\, q(1 - q)(y - q)$.

Данная последовательность операций производится *один раз в пересчете на один ФН*, т. е. 2 вычитания и 3 умножения, требуя затрат времени $2a+3m$. Для коррекции каждого веса связи дополнительно выполняется одна операция умножения и одно сложение $w_i(n+1) = w_i(n) + x_i \Delta$.

Итого для n_{k-1}-входового ФН в выходном k-м слое НС суммарные затраты времени на коррекцию всех весов связей составят

$$t = 2a+3m + n_{k-1}\ (a+m) = (n_{k-1}+2)\ a + (n_{k-1}+3)\ m,$$

где a – время выполнения операции сложения/вычитания; m – время выполнения операции умножения.

При подстройке весов ФН скрытого слоя. Веса связей выхода ФН умножается на величину δ соответствующего ФН в выходном слое. Величина δ для ФН скрытого слоя, получается суммированием полученных произведений и умножением на производную сжимающей функции $\delta = q(1-q)\left[\sum_i \delta_i w_i\right]$.

То есть полученные значения ошибки δ_i i-го ФН $(j+1)$-го слоя умножаются на значение веса w_i и суммируются по числу n_{j+1} ФН в $(j+1)$-м слое НС. Полученная суммарная ошибка умножается на значение производной функции активации текущего j-го слоя НС. Итого для определения ошибки, приведенной к выходу ФН j-го скрытого слоя, необходимо выполнить $(n_{j+1}+2)$ операций умножения и n_{j+1} операций сложения для каждого ФН в скрытом слое $n_{i+1}\ a + (n_{j+1}+2)\ m$.

Затем для коррекции весов ФН j-го скрытого слоя, как и для случая выходного слоя НС, следует выполнить n_{j-1} операцию умножения $x_i \Delta$ и столько же операций сложения $w_i + \Delta w_i$, т.е. затратить время, равное $n_{j-1}\ (a+m)$.

Таким образом, для одного цикла коррекции всех весов k-слойной НС с размерностью входного вектора n_0 и числом ФН в слоях n_i, где $i=1,\ldots,k$ потребуется время

$$t_{NN} = n_k((n_{k-1}+2)a + (n_{k-1}+3)m) + \sum_{i=1}^{k-1} n_i(n_{i-1}(a+m) + n_{i+1}a + (n_{i+1}+2)m). \quad (9)$$

Алгоритм логарифмического обратного распространения ошибки

Подстройка весов выходного слоя. Для подстройки весов выходного слоя НС вычисляются значения логарифмов координат выходного вектора Q_i^k, $1 \le i \le n_k$, соответствующего целевого вектора Y_i^k, $1 \le i \le n_k$ и определяются значения координат вектора логарифмической ошибки Δ_i^k. Здесь k – номер выходного слоя НС, а n_k – размерность выходного вектора НС. Для каждой i-й координаты вектора ошибки (выход i-го ФН выходного слоя НС):

$\Delta_i^k = Y_i - Q_i = a \ln \frac{y_i}{q_i}$, $1 \le i \le n_k$, где n_k – число ФН в выходном слое НС. Выполняемые операции – табличное логарифмирование и вычитание.

Выбирается *отличная от нуля* минимальная из координат вектора ошибки *k*-го слоя $\Delta^k_{\min}$, для чего необходимо выполнить n_k -1 операцию сравнения.

В процессе коррекции веса связей для *i*-го ФН выходного слоя НС формируется значение логарифма приращения $\Delta W^k{}_i = \Delta_i^k - \frac{\Delta^k_{\min}}{2}$ для всех весов *i*-го ФН *k*-го слоя НС , для чего необходимы 1 операция сдвига и 1 операция вычитания. Коррекция весов осуществляется путем сложения логарифмов предыдущих значений весов и соответствующих приращений $W_i^k{}' = W_i^k + \Delta W_i^k$.

Итого для ФН в выходном *k*-м слое НС затраты времени на коррекцию весов составят $t_{out} = n_k(2l + a) + (n_k - 1)a + s + (n_{k-1} + 1)\, n_k a$,

где *a* – время выполнения операции сложения/вычитания/сравнения; *l* – время выполнения операции логарифмирования; *s* – время выполнения операции сдвига.

Для коррекции в скрытых слоях НС В качестве логарифма приращений $\Delta W^k{}_i = \Delta_i^k - \frac{\Delta^k_{\min}}{2}$, в частности, *j*-го слоя, выбирается *отличное от нуля* значение $\Delta X^j{}_i = \frac{\Delta^k_{\min}}{2}$ единое для всех i, $1 \le i \le n_j$, где n_j – число ФН в *j*-м слое НС.

Подстройка весов скрытого слоя. Так как из слоя *k* на все ФН скрытого *j*-го слоя подается одно и то же значение логарифма ошибки $\Delta X^j{}_l = \frac{\Delta^k_{\min}}{2}$, то передаваемое к следующему (*j*-1)-у скрытому слою НС значение логарифма приращения будет меньше входного $\Delta X^j{}_l$ вдвое. То есть $\Delta X^{j-1}{}_i = \frac{\Delta_i^j}{2} = \frac{\Delta^k_{\min}}{4}$ для всех i, $1 \le i \le n_{j-1}$, где n_{j-1} – число ФН в (*j*-1)-м слое НС, а $\Delta W^j{}_i = \Delta_i^j - \frac{\Delta^k_{\min}}{4} = \frac{\Delta^k_{\min}}{4}$ для всех i, $1 \le i \le n_j$, n_j – число формальных нейронов в *j*-м слое нейронной сети.

Таким образом, в каждом *j*-м скрытом слое для коррекции весов необходима одна операция сдвига и $n_{j-1}\, n_j$ операций алгебраического сложения.

Подстройка весов входного слоя. Оставшееся после коррекции весов первого из скрытых слоев значение логарифма ошибки полностью суммируется с

логарифмами весов всех ФН входного слоя обучаемой НС, что потребует $n_0\, n_1$ операций алгебраического сложения.

Итого для одного цикла обучения логарифмической НС суммарные затраты времени на коррекцию весов связей составят

$$t_{NN} = n_k(2l + a) + (n_k - 1)a + (k-1)s + (n_{k-1} + 1)\, n_k a + \sum_{i=1}^{k-1} n_{i-1} n_i a\,. \qquad (10)$$

Последнее выражение не содержит операций умножения, операция логарифмирования выполняется табличным преобразованием, а для задания параметра скорости обучения НС используется однотактная операция сдвига.

Для проведения сравнения полученных соотношений (9) и (10) отдельно определим относительный выигрыш во времени обучения выходного слоя НС и соотношение времени обучения внутренних и входного слоев нейронной сети.

Отношение времени обучения *выходного слоя* НС в случае обратного распространения ошибки к времени обучения выходного слоя логарифмической нейронной сети описывается выражением, независящим от числа ФН в выходном слое НС

$$\frac{t_{NN\_B\_P}}{t_{NN\_Ln}}_{out} \approx 1 + \frac{m}{a}\,. \qquad (11)$$

То есть цикл обучения выходного слоя НС по методу логарифмического обратного распространения ошибки экономичнее примерно во столько раз, во сколько быстрее выполняется операция сложения по отношению к операции умножения в конкретной ИТ-систем.

Аналогично определим отношение времени обучения *входного и внутренних слоев* НС для случаев классического обратного распространения ошибки к времени обучения входного и внутренних слоев логарифмической нейронной сети:

$$\frac{t_{NN\_B\_P}}{t_{NN\_Ln}}_{in} = 1 + \sum_{i=1}^{k-1}\left(\left(1 + \frac{n_{i+1} + 2}{n_{i-1}}\right)\frac{m}{a} + \frac{n_{i+1}}{n_{i-1}}\right). \qquad (12)$$

Из выражения (12) следует, что и при обучении внутренних слоев НС соотношение (11) не ухудшается, а, напротив, эффективность обучения логарифмической сети возрастает с увеличением числа слоев НС. В частности, если предположить постоянство количества ФН в скрытых слоях НС, то зависимость (12) становится близкой к линейной (рис. 3.8.) с тангенсом угла наклона, равным $\left(\frac{2m}{a} + 1\right)$.

3.2. Организация безопасного хранения информации

Нейросетевой командный пул строится на основе специализированных модулей памяти и ориентирован на управление потоком данных. Логика работы памяти в УПД-машинах обеспечивает защищенность хранимой информации: 1) операция записи данных производится не по конкретному адресу памяти, а по содержанию; 2) отсутствует операция считывания данных из ЗУ и, следовательно, непосредственный доступ к хранимой информации. Готовые к обработке данные, представленные в виде пакетов, извлекаются из памяти автоматически - без управления извне.

Нейросетевые СЗИ в командном пуле представляются конечным множеством КП - пакетной нейросетевой программой.

Командные пакеты содержат следующий набор полей:

OP	F	$Data_m \dots Data_1$	$DST_n \dots DST_1$	ACT

1) командное (OP) определяют одну из функций нейросетевого базиса;

2) функциональное (F) содержит значения весов и порогов срабатывания ФН или группы ФН;

3) операндные ($Data_m \dots Data_1$) предназначены для буферизации входной информации, поступающей в КП–приемник результата из КП-источников операндов, m – число операндных полей пакета;

4) коммуникационные ($DST_n \dots DST_1$) задают топологию связей между ФН, содержат адреса КП–приемников результата, n – число КП-приемников результата;

5) служебные (ACT) – вспомогательные поля, определяющие, как правило, контекст вычислений.

Готовые к обработке командные пакеты (с укомплектованными операндными полями) передаются через коммуникационную среду в процессорный узел, где свободный процессорный элемент (аналог ФН) выполняет преобразование содержимого КП и формирует пакеты данных по числу КП–приемников результата.

Пакет данных – контейнер, переносящий значения с выхода одного ФН на вход другого ФН, - как правило, состоит из следующих полей:

RES	DST_{ij}	ACT

1) результата (RES) содержит значение, сформированное в ФН-источнике, для передачи ФН–приемнику результата;

2) коммуникационного (DST_{ij}) задает связь между двумя ФН, по которой передается результат на j-й вход i-го ФН-приемника, $0 \le i \le r$; $0 \le j \le n$; здесь r - число ФН в слое НС, n - число входов отдельного ФН;

3) служебного (ACT) – вспомогательное поле.

Работа командного пула может быть описана следующим образом.

Исходное состояние. Многофункциональная память не производит операций, однако содержит конечное множество КП с заполненными командными, коммуникационными и функциональными полями, то есть загруженную в пул обученную НС. На входе командного пула может находиться входная очередь (или входной регистр), предназначенная для буферизации поступающих ПД и формирующая два флага: «Очередь пуста» и «Очередь заполнена». Задача входной очереди – накапливать асинхронно поступающие ПД и инициировать загрузку пакета данных, находящегося в начале очереди, в пул команд, если первый флаг сброшен.

В процессе загрузки ПД из входной очереди в пул команд поле результата *Res* ПД заносится в одно из операндных полей $Data_j$, $0 \le i \le m$, КП–приемника результата, определяемое коммуникационным полем (DST_{ij}) и служебным полем (*Act*) ПД. В блоке памяти готовности данных устанавливается бит готовности, ассоциированный с операндным полем.

Извлечение КП. Если заполнены данными все операндные поля $Data_m$ ... $Data_1$ некоторого КП (установлены все связанные с ним биты готовности), то КП выталкивается из пула команд и производится очистка ассоциированных с ним битов готовности данных в блоке памяти готовности данных.

Пул представляет собой память, не имеющую внешних шин записи/чтения, что исключает возможность записи по определенному адресу и считывания содержимого конкретной ячейки памяти. Доступной для загрузки является входная очередь, а для извлечения – выходная очередь пула. Т.е. командный пул является «непрозрачной» для пользователя памятью, в которую через входную очередь загружаются ПД, а из выходной очереди извлекаются готовые КП.

В качестве известного решения локального пула можно назвать командный пул мультипроцессорной системы DDDP с УПД [104]. Операндная память адресуется по содержанию коммуникационного (DST_i) и служебного (*ACT#*) полей ПД посредством механизма хэширования; командная память – только полем DST_i. Служебное поле *ACT#* необходимо для обеспечения корректной передачи результатов работы при одновременном вызове некоторой процедуры из различных частей программы или повторном прохождении циклических участков программы, при которых формируются КП с различными номерами активации и значениями операндов, но содержащие идентичные командные и коммуникационные поля.

Специфика организации НС требует внесения ряда изменений в командный пул и, прежде всего, введения модулей памяти для хранения функциональных параметров КП (FM) и механизма готовности данных (RCM), увеличения числа как операндных, так и коммуникационных полей (рис. 3.9).

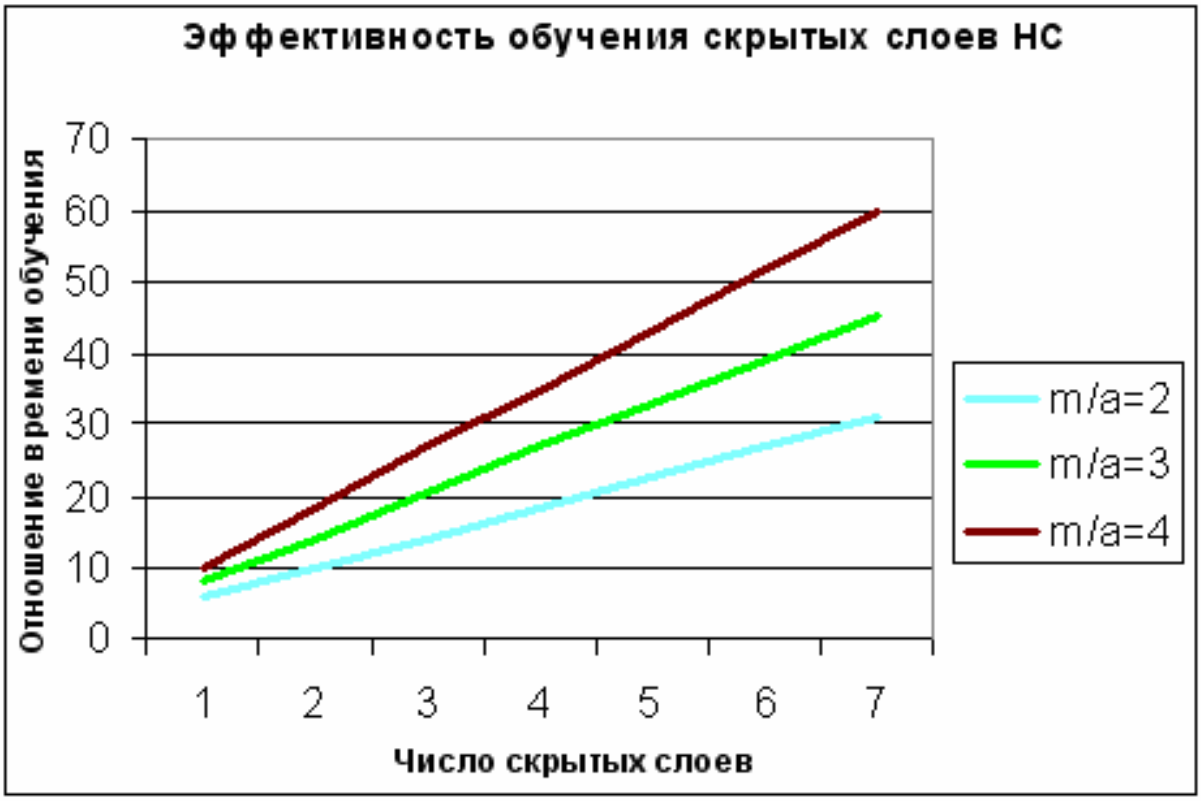

Рис. 3.8. Иллюстрация эффективности обучения логарифмических НС

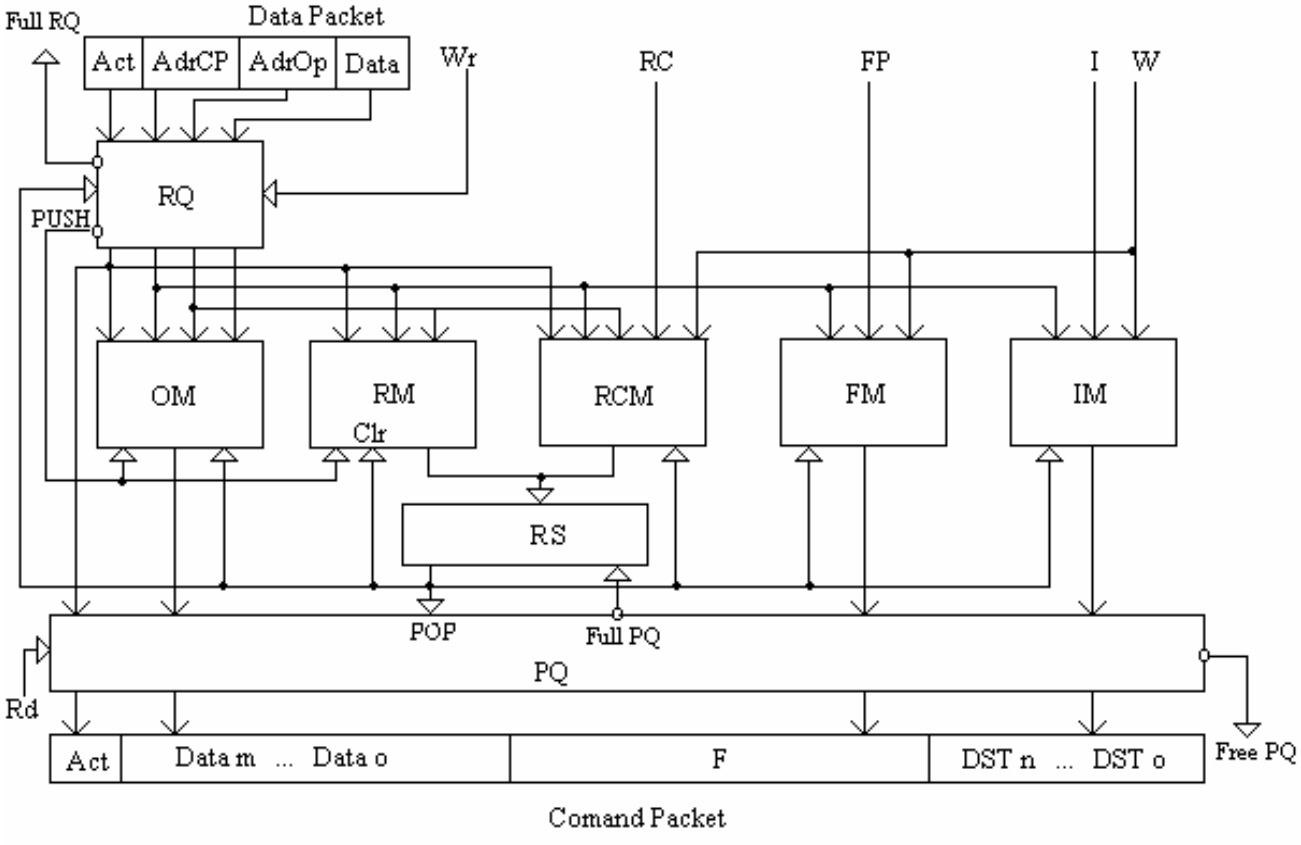

Рис. 3.9. Информационно защищенный пул команд

Командное поле в КП может отсутствовать, если все КП будут выполнять одну функцию – к примеру, функцию ФН или слоя ФН.

Информационно защищенный командный пул образован из следующих специализированных модулей памяти:

1) *OM* – память операндов предназначена для буферизации значений данных, передаваемых по межнейронным связям НС на входы ФН; в адресном сечении ОМ хранятся значения операндов, поступивших на входы конкретного ФН (или слоя ФН) к некоторому моменту времени;

2) *RM* – память готовности КП к обработке хранит булеву матрицу, отражающую динамику поступления операндов на входы ФН сети; заполнение единицами некоторого адресного сечения матрицы соответствует моменту поступления всех операндов на входы некоторого ФН сети; данный момент аппаратно отслеживается схемой готовности RS, формирующей сигнал POP извлечения КП из пула и сигнал Clr обнуления данного адресного сечения RM;

3) *RCM* – память управления готовностью позволяет явным образом указать, поступлением каких из операндов для данного ФН можно пренебречь при формировании сигнала POP схемой готовности RS; булева матрица, хранимая в RCM, маскирует булеву матрицу, формируемую в RM;

4) *FM* (Functional Memory – память функциональных параметров предназначена для долговременного (на срок функционирования НС) хранения значений весов и порогов срабатывания ФН;

5) *IM* – память команд хранит топологию НС; командные поля (в случае использования нескольких базовых функций) несут информацию о типе компонента сети, а коммуникационные поля определяют межкомпонентные связи; если командные пакеты реализуют одну базовую функцию (например, функцию ФН), то IM содержит только коммуникационную информацию;

6) RQ – магазинная память, размещаемая на входе пула с целью буферизации ПД;

7) PQ – магазинная память, размещаемая на выходе пула для буферизации готовых к обработке КП.

Если в командный пул загружены одна или ряд НС (ПНП), то пул будет находиться в состоянии покоя до тех пор, пока во входную очередь *RQ* не поступит хотя бы один ПД. Занесение ПД в *RQ* по внешнему сигналу *Wr* вызовет формирование внутреннего сигнала управления *PUSH*, который, вызовет запись значения из поля *Data* ПД в операндное поле КП, адрес размещения которого в модуле ОМ определяется полями: служебным *Act*, адреса КП *AdrCP* и адреса операнда в командном пакете *AdrOp*. По тому же адресу в модуль памяти *RM* будет записана единица в булеву матрицу готовности. Если в результате

последней операции в данном адресном сечении модуля памяти готовности образуется двоичное слово, которое при наложении слова маски, считанного по тому же адресу из модуля *RCM*, сформирует определенный двоичный код, к примеру, содержащий единицы во всех разрядах, то схема готовности *RS* аппаратно сформирует сигнал POP извлечения КП из пула. Сигнал *POP* вызовет считывание из модулей памяти *OM, FM, IM* и размещение в выходной очереди *PQ* готового к обработке КП, обнуление адресного сечения (по *AdrCP*) в модуле памяти готовности и извлечение из входной очереди *RQ* очередного ПД, если флаг «Очередь пуста» сброшен. Выходная магазинная память сбросит сигнал «Очередь *PQ* пуста», инициирующий извлечение готового КП для обработки процессорным блоком по сигналу *Rd*, формируемому извне.

Командный пул (рис. 3.9) в большей мере соответствует для размещения и функционирования уже обученной НС. В этом случае достаточно однократного занесения командной, функциональной информации и матрицы управления готовностью с функционально обособленных шин *I* (Instructions), *FP* (Functional Parameters) и *RC* (Readiness Control) по сигналу записи *W* в соответствующие модули памяти, после чего пул переводится в рабочий режим.

Если же предусматривается размещение в пуле команд НС, подлежащей обучению, то необходимо будет ввести коррективы во входные цепи модулей памяти *RCM, FM и IM*. Модули памяти функциональных параметров *FM* могут быть организованы аналогично модулям памяти операндов *OM* и заполняться пакетами данных через очередь результатов *RQ* в силу следующих соображений. Процесс настройки функциональных параметров НС будет производиться размещенной в пуле обучающей ПНП, результатом работы которой будет формирование ПД, содержащих в полях *Data* значения весов или порогов срабатывания ФН для обучаемой НС. То есть одна НС (обучающая) будет использовать поля функциональных параметров другой НС (обучаемой) в качестве своих операндных полей.

Командный пул представляет собой МРВС, размещенную в пределах функционально завершенного кристалла либо секционированного базового блока. МРВС и принципы монолитности исполнения позволяют обеспечить повышенную информационную защищенность пула за счет замыкания потоков данных в пределах устройства и минимизации обмена информацией с внешней средой, а подход УПД – за счет специфики работы пула команд, затрудняющей несанкционированный доступ к. размещенным в пуле данным.

Основным недостатком существующих подходов к организации нейросетевых вычислений является разнесение 1) в пространстве устройств хранения и обработки информации, 2) во времени процессов записи/считывания из памяти, передачи и обработки данных, что приводит к многочисленным непроизводи-

тельным затратам времени. МРВС позволяют выполнить пространственное и временное совмещение процесса обработки информации с операциями записи/чтения, проводимыми в многофункциональной памяти. Рассматриваемый подход к технической реализации командного пула базируется на особенностях МРВС, проекте интеллектуальной памяти *IRAM* и специфике выполнения операций в нейросетевом базисе.

МРВС – это структуры характеризующиеся многофункциональностью и регулярностью и поэтому максимально приспособленные к производству методами интегральной технологии. Многофункциональность определяется возможностью реализации структурой неединичного набора функций. Регулярность – повторяемостью элементов и связей структуры [105]. Интеллектуальная память *IRAM* дополняет базовые положения МРВС принципом монолитности исполнения вычислителя, что приводит к пространственно-временному замыканию основных потоков преобразуемых данных внутри функционально завершенного кристалла, минимизации обмена информацией с внешней средой [106], и, следовательно, снижает вероятность несанкционированного доступа к информации. Управление вычислительным процессом с помощью потоков данных обеспечивает инициализацию параллельной обработки данных в пределах МРВС в зависимости от порядка поступления значений данных, передаваемых посредством пакетов. НС являются частным случаем МРВС, так как в качестве базового многократно повторяющегося в структуре элемента используется ФН, реализующий набор операций нейросетевого базиса, и имеет место повторяемость связей между ФН в сети. Кроме того, специфика НС позволяет строить надежные сложные системы даже из малонадежных элементов, а функциональная избыточность НС – при разрушении части не вызывать потери системой своей функциональности [107].

Задачу разработки НС можно представить как отражение процесса нейросетевых вычислений в структуре многофункциональной памяти в соответствии с идеологией МРВС, интеллектуальной памяти и УПД. Многообразие реализуемых НС функций, основные достоинства, прежде всего, информационная защищенность зависят от системы связей между ФН. Другим достоинством НС является внутренний параллелизм, который позволяет при относительно скромном быстродействии базового элемента решать достаточно сложные, трудно формализуемые задачи в реальном масштабе времени [107]. Следовательно, при проектировании нейросетевой систем ИТ необходимо ориентироваться на принципы УПД, которые позволяют реализоваться присущему НС самоуправлению вычислениями. Кроме того, логика работы памяти в УПД-машинах обеспечивает защищенность хранимой информации: операция записи может производиться не по конкретному адресу памяти, а «по содержанию», то

есть с использованием ассоциативного доступа к информации; отсутствует операция считывания данных из ЗУ и, следовательно, исключен непосредственный доступ к хранимой информации.

Объектом дальнейшего рассмотрения является многофункциональная память с аппаратной реализацией базовых функций ФН для выполнения распределенных вычислений под управлением потоком данных, НС в которой представляется пакетной нейросетевой программой, размещенной в командных ячейках пула команд.

В соответствии с подходом УПД не важен порядок поступления в соответствующие операндные поля КП входных значений X_i, приводящих к установке битов готовности R_i. Однако такие архитектурные особенности как структура пула команд, используемый интерфейс могут влиять на производительность вычислений. Так наличие входной очереди для фиксации асинхронно поступающих в пул команд ПД задает последовательный характер заполнения полей X_i и позволяет совместить занесение входных данных с обработкой информации непосредственно в командном пуле. В частности, умножение аргумента X_i на вес W_i и последующее накопление результата X_iW_i в поле аккумулятора А позволяет заменить в формате КП операндные поля X_i одним накопительным полем А (рис. 3.10).

Над полями готовых к обработке КП выполняются преобразования, аналогичные функции ФН, и формируются ПД по числу КП–приемников результата. Специфика пула команд состоит в построении «непрозрачной» для пользователя памяти, в которую через входную очередь загружаются пакеты данных, а из выходной очереди извлекаются готовые к обработке КП или ПД. Пул команд представляет собой информационно защищенную память, не имеющую внешних шин записи/чтения, что исключает возможность записи информации по определенному адресу и считывания содержимого конкретной ячейки памяти.

Согласно идеологии УПД-машин готовый КП через селекторную сеть должен передаваться к процессорным узлам, а результаты обработки в виде ПД через распределительную сеть – в командные ячейки пула команд. При достаточно большом числе PU, что характерно для НС, возрастает сложность и временные задержки в сетях передачи пакетов. В соответствии с идеологией МРВС следует произвести обработку готовых КП непосредственно в пуле команд.

Для обеспечения распараллеливания вычислений необходимо перейти к множеству локальных пулов команд, что позволяет сочетать последовательный характер обработки отдельных КП в пулах с распределением обработки информации по значительному числу одновременно работающих локальных пулов.

Нейронная сеть формируется путем размещения КП в командных ячейках локальных пулов и заполнения командных, коммуникационных и функциональных полей либо на этапе обучения сети, либо (в случае уже обученной НС) на этапе программирования. В связи с тем, что НС в виде ПНП размещается в командных ячейках локальных пулов, целесообразно при распределении КП отобразить двумерную совокупность ФН слоистой сети на линейную последовательности локальных пулов таким образом, чтобы КП, соответствующие ФН отдельного слоя, размещались в командных ячейках различных локальных пулов.

Топология НС определяется коммуникационными полями Di, которые определяют связи между ФН слоев НС, которые задаются в процессе программирования. Результаты распределенной обработки в виде ПД направляются в ряд локальных пулов, что делает обязательным наличие распределительной сети.

Функциональные поля Wi, отмеченные в командном пакете признаком C (const), не должны изменяться в режиме функционирования НС, но в процессе настройки функциональных параметров (обучения сети) они выполняют роль операндных полей командных ячеек и подлежат модификации.

Нейронная сеть самоуправляется механизмом полной или частичной готовности данных и передачей результатов обработки КП-источников в операндные поля КП-приемников посредством ПД. Нейронные сети начинают функционировать по мере загрузки ПД во входную очередь и далее – в поля командных ячеек.

3.2.1. Оценка эффективности многофункционального пула

Эффективность многофункционального командного пула обусловлена совмещением в пространстве и времени процессов передачи, хранения и обработки информации. Рост функциональной устойчивости нейросетевой системы обеспечивается замыканием большей части информационного потока в пределах многофункционального пула, а повышение производительности связано с минимизацией пересылок информации через интерфейсы.

Для иллюстрации оценим временные затраты, связанные с циклом работы НС, размещенной в 1) командных пулах с разнесенными в пространстве зонами хранения и обработки информации и 2) на базе многофункционального командного пула.

Для первого случая свойственна передача по интерфейсам двух разновидностей пакетов: пакетов данных и командных пакетов и послойная реализация функции НС [4, 108]. Для последовательного занесения в пул команд ПД, относящихся к отдельному ФН слоя НС необходимы затраты времени $n_{i-1}\, t_c$ (t_c - время передачи через интерфейс одного ПД, n_{i-1} - число ФН предыдущего слоя

НС), а для слоя в целом - $n_{i-1} n_i t_c$ (n_i - число ФН текущего слоя НС). По числу ФН текущего слоя формируются КП для передачи по интерфейсу в зону обработки (затраты времени $n_i t_c$). В операционной зоне в для каждого ФН вычисляются взвешенные сигналы $n_{i-1} t_m$ (t_m - время выполнения операции умножения) с последующим накоплением результата $n_{i-1} t_a$ (t_a - время выполнения операции сложения), а для слоя НС - $n_i n_{i-1} (t_m + t_a)$.

Затраты времени для слоя НС - $n_i (n_{i-1} t_c + t_c + n_{i-1} (t_m + t_a)$, а НС из k слоев

$$t_{NN} = n_k t_c + \sum_{i=1}^{k-1} n_i (n_{i-1} t_c + t_c + n_{i-1} (t_m + t_a) , \quad (13)$$

где первое слагаемой учитывает передачу по интерфейсу ПД с результатами вычислений из выходного слоя НС, содержащего n_k ФН.

Для случая многофункционального пула операции передачи ПД по интерфейсу $n_{i-1} n_i t_c$ совмещены с процессом обработки (по мере поступления операндов) - $t_m + t_a$, отсутствует необходимость формирования и передачи КП через интерфейс в зону обработки, поэтому общие затраты времени снижаются

$$t_{MNN} = n_k t_c + \sum_{i=1}^{k-1} n_i (n_{i-1} t_c + t_m + t_a) . \quad (14)$$

Эффективность использования многофункционального пула оценим отношением выражений (13) и (14). Для простоты иллюстрации (рис. 3.11) полагаем, что число ФН в слоях НС одинаково, исключаем первое слагаемое, связанное с выдачей результатов работы НС, в качестве параметра выбрано tc - время передачи пакета по интерфейсу.

3.3. Уровни описания нейросетевых СЗИ

При описании нейронных сетей нейросетевыми пакетными программами возможна различная степень детализации: командный пакет может соответствовать одной из функций нейросетевого логического базиса, функции формального нейрона, слоя из формальных нейронов или нейронной сети в целом [116]. Соответственно изменяются требования к проектированию базовых блоков и сложность технической реализации нейронной сети. Представление командными пакетами операций, соответствующих отдельным функциям формального нейрона, не целесообразно ввиду разнородности и малой функциональной сложности операций и возрастания потока пакетов данных с промежуточными результатами вычислений. Поэтому следует рассматривать градации сложности командных пакетов, начиная с уровня формальных нейронов, а именно, учитывая следующие соответствия: *КП–ФН*, *КП–слой ФН*, *КП–НС*.

D_1	D_2	...			D_m
C	w_0	w_1	...		w_n
N	A	R_1	R_2	...	R_n

Рис. 3.10. Командный пакет для слоистой НС

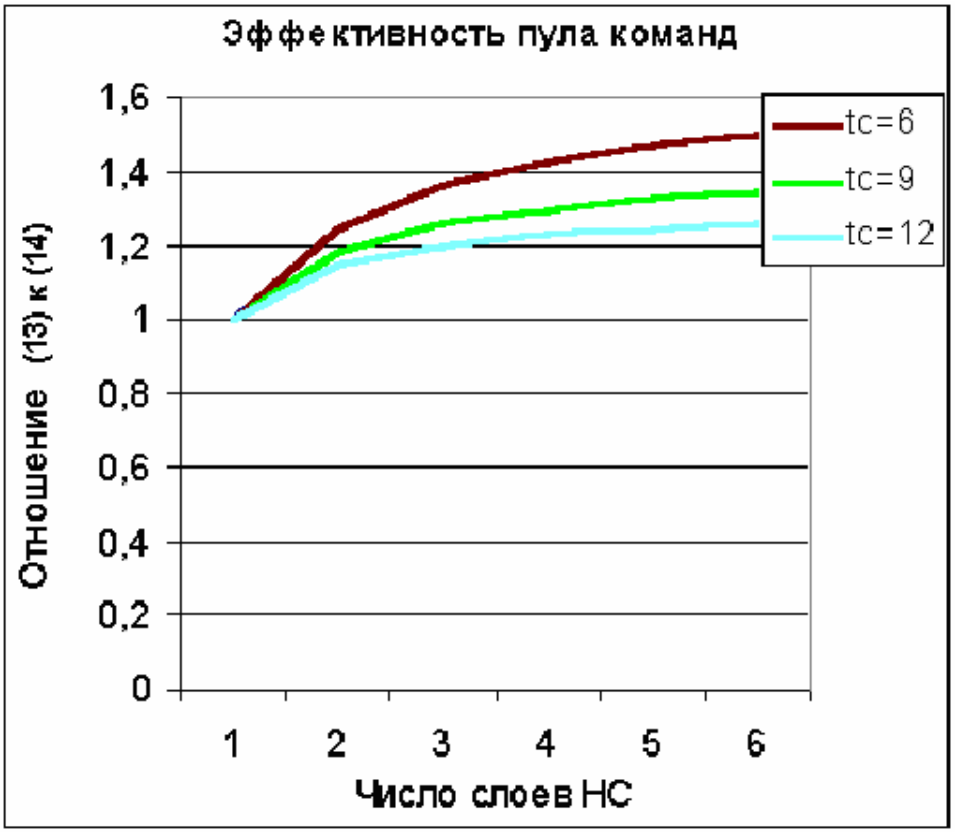

Рис. 3.11. Эффективность многофункционального пула по сравнению с пулом команд

Таблица 3.1

	По входам ФН	По ФН	По слоям НС	Архитектурные особенности среды
1.	Послед.	Послед.	Послед.	Последовательная в пуле распределенная обработка, один PU в пуле
2.	Послед.	Парал.	Послед.	Параллельная в пуле распределенная обработка, PU по числу ФН в слое
3.	Парал.	Послед.	Послед.	Последовательная в пуле распределенная обработка, один PU в пуле
4.	Парал.	Парал.	Послед.	Параллельная в пуле распределенная обработка, PU по числу ФН в слое
5.	Парал.	Парал.	Парал.	Параллельная распределенная обработка, PU по числу ФН в слое, конвейеризация обработки по слоям НС

Возможные архитектурные решения нейросетевой вычислительной среды для соответствия КП–ФН в зависимости от характера выполнения операций сведены в табл. 3.1.

Реализация функции формального нейрона на основе последовательной распределенной арифметики (SDA - Serial Distributed Arithmetic) дает наибольший выигрыш по аппаратным ресурсам, однако требует больших временных затрат [5, 103]. При использовании параллельной распределенной арифметики (PDA - Parallel Distributed Arithmetic - варианты 1 и 2 из табл. 3.1) получаем компромиссное решение с точки зрения аппаратных затрат и времени реализации функции формального нейрона. В техническом решении (рис. 3.12) результат работы формального нейрона формируется в локальном пуле путем последовательного суммирования взвешенных значений входов формального нейрона, последовательно или параллельно по ФН отдельного слоя сети и последовательно по слоям НС. Последовательный характер обработки по входам формального нейрона обусловлен использованием в качестве интерфейса для доставки пакетов данных кольцевой шины с последовательной передачей пакетов данных с выходов на входы локальных пулов. Поступление пакетов данных в локальный пул вызывает запуск цепочки операций чтение – модификация - запись с проверкой готовности командных пакетов, что эквивалентно выполнению операций нейросетевого логического базиса по взвешиванию отдельного входа и накопления поступивших взвешенных входов в командном пакете.

Последовательный или параллельный характер обработки данных по отдельным формальным нейронам слоя сети зависит от распределения командных пакетов нейросетевой пакетной программы по локальным пулам. Если все командные пакеты программы фиксированы в одном локальном пуле («вертикальное» размещение командных пакетов), то возможно только последовательное выполнение функций отдельных формальных нейронов сети. Если же командные пакеты, соответствующие формальным нейронам одного слоя, распределены по различным локальным пулам («горизонтальное» размещение КП), то возможны варианты параллельной или параллельно-последовательной обработки. Во всех рассмотренных случаях сохраняется последовательный характер выполнения вычислений по слоям нейронной сети.

При параллельной обработке значений координат входного вектора в пределах локального пула (варианты 3 и 4 табл. 3.1) возрастают аппаратные затраты из-за одновременного выполнения операций взвешивания входного вектора и применения свертывающего дерева сумматоров. При переходе от бинарного представления формального нейрона к формальному нейрону с вещественными значениями обрабатываемых данных потребность в аппаратных ресурсах воз-

растает пропорционально числу входов в основном из-за увеличения числа блоков умножения.

По оценкам [103] использование восьми конвейерных умножителей 8x8 бит в дополнительном коде, выполненных по алгоритму Бута, свертывающего дерева сумматоров и компаратора с загружаемым 8-разрядным порогом требует значительного объема логических ресурсов: 44% от ПЛИС XC4036XLA и до 18% от ПЛИС XC4085XLA. Применение восьми параллельно-последовательных 8-разрядных умножителей, 16-разрядного аккумулятора частичных произведений с временным мультиплексированием приводит к снижению объема аппаратных затрат до 12% от логических ресурсов ПЛИС XC4036XLA и 5% от XC4085XLA на формальный нейрон. То есть на кристалле XC4085 размещается до двадцати 8-входовых формальных нейронов. Реализация формальных нейронов на основе последовательной распределенной арифметики SDA дает наибольший выигрыш по занимаемым ресурсам - около 3% от общих возможностей ПЛИС XC4085XLA, что эквивалентно размещению на одном кристалле около 30 формальных нейронов (85 Кбайт/нейрон против 570 Кбайт/нейрон для случая параллельной арифметики).

Слоистая структура нейронной сети определяет последовательный характер обработки информации по слоям НС. В этой связи параллелизм вычислений по слоям нейронной сети (5 вариант из табл. 3.1) может быть обеспечен только конвейеризацией вычислений, производимой параллельно в локальных пулах команд над последовательно во времени поступающими входными векторами с фиксацией промежуточных векторов результатов.

Основным достоинством нейросетевой вычислительной среды, описываемой командными пакетами на уровне соответствия *КП–ФН*, является независимость от топологии реализуемых нейронных сетей, так как коммуникационными полями командных пакетов задаются все связи между отдельными формальными нейронами сети. Обратная сторона подобной детализации – повышенная загрузка цепей коммуникации, так как каждый пакет данных соответствует отдельной связи формального нейрона. Другими словами, PU формируют большое количество малоинформативных пакетов данных, для передачи которых необходимы высокоскоростные интерфейсы. Частично данная проблема может быть решена за счет указания в адресном поле пакетов данных *всех связей* конкретного формального нейрона-источника с формальными нейронами-приемниками вместо единственной связи с конкретным формальным нейроном-приемником.

Адр.входа ФН–приемника 1	...	Адр.входа ФН–приемника *n*	Данные от ФН-источника

Такое представление ПД - суперпакетом данных позволяет в *n* раз, где *n* – число формальных нейронов в слое, уменьшить количество передаваемых пакетов данных, но потребует усложнения цепей адресной селекции формальных нейронов.

Степень детализации КП–слой ФН. Следующим шагом, дающим возможность сократить количество пакетов данных в цепях коммуникации, является переход к степени детализации *КП–слой ФН*. То есть командный пакет в качестве объекта описывает слой формальных нейронов, который производит обработку входного вектора $\boldsymbol{X}$ путем умножения на матрицу весовых коэффициентов $\boldsymbol{W}$ с целью формирования выходного вектора $\boldsymbol{OUT = \varphi(XW)}$. При этом можно вновь вернуться к простой форме пакетов данных, число которых в пакетной программе определяется количеством слоев представляемой НС.

Адрес входа слоя–приемника	Вектор данных от слоя-источника

Необходимо обратить внимание на повышенное потребление аппаратных ресурсов нейросетевой вычислительной средой в рассматриваемом случае, так как при реализации нейронной сети с различным числом формальных нейронов в отдельных слоях НС в качестве ориентира для выделения аппаратных средств будет выбран слой с максимальным количеством формальных нейронов.

Для перехода к схеме: параллельно по входам формального нейрона – параллельно по ФН слоя – последовательно по слоям нейронной сети следует ожидать дальнейшего увеличения аппаратных затрат, так как процессы взвешивания элементов входного вектора $\boldsymbol{X}$ потребуют увеличению числа функциональных блоков пропорционально числу входов формального нейрона. В этом случае базовый блок нейросетевой вычислительной среды будет представлять собой двумерную систолическую матрицу нейропроцессорных блоков - *PN*, «горизонтальное» измерение которой будет соответствовать числу формальных нейронов слоя нейронной сети, а «вертикальное» – количеству входов формального нейрона.

При выборе архитектуры базового блока нейросетевой вычислительной среды, соответствующей схеме: параллельно по входам формального нейрона – параллельно по ФН слоя – параллельно по слоям нейронной сети следует ориентироваться на послойную передачу с фиксацией промежуточных результатов вычислений в процессе конвейеризации работы нейронной сети.

Степень детализации КП–нейронная сеть. Максимально возможная степень сложности описания нейронной сети – это задание всех функциональных параметров и связей одним командным пакетом. Загрузка подобного командного пакета в базовый блок нейросетевой вычислительной среды может рассматриваться как ее настройка на выполнение функции конкретной нейронной сети.

Поступление входного вектора в виде пакета данных запускает в базовом блоке процесс вычислений, который завершается формированием пакетов данных с вектором результата:

Адрес приемника	Вектор данных от нейронной сети

Подобная организация работы нейронной сети не сопровождается передачей промежуточных результатов вычислений – все потоки данных замкнуты внутри базового блока - и практически исключена возможность оказания несанкционированного воздействия на ход процесса формирования результата.

3.3.1. Командные пулы уровня формального нейрона

При описании нейронных сетей, детализированных до уровня формальных нейронов, необходима соответствующая аппаратно-программная реализация нейросетевой вычислительной среды. Степень детализации *КП–ФН* является минимально возможной для представления объектов-данных, передаваемых пакетами данных по межнейронным связям, а именно, между конкретным выходом формального нейрона-источника и определенным входом формального нейрона–приемника результата.

На рис. 3.12 представлена структура локального пула, организованного в соответствии с вышеприведенными положениями и использующего механизм частичной готовности для отслеживания поступления заданной совокупности операндов, представленных бинарными значениями.

Локальный пул образован из следующих специализированных модулей памяти:

- *RM* (Readiness Memory) – память готовности командных пакетов к обработке хранит булеву матрицу, отражающую динамику поступления операндов на информационные входы формальных нейронов сети и фиксирующая факт поступления операнда в локальный пул установкой единицы в соответствующем бите матрицы; заполнение единицами некоторого адресного сечения матрицы соответствует моменту поступления всех операндов на входы некоторого формального нейрона; данный момент аппаратно отслеживается схемой готовности *RS* (Readiness Scheme), формирующей сигнал *POP* извлечения пакета из локального пула и сигнал обнуления данного адресного сечения *RM*;

- *RCM* (Readiness Control Memory) – память управления готовностью позволяет явным образом указать, поступлением каких из операндов для данного формального нейрона можно пренебречь при формировании сигнала *POP* схемой готовности *RS*; булева матрица, хранимая в *RCM*, маскирует булеву матрицу, формируемую в *RM* в процессе загрузки пакетов данных в локальный пул;

- *IM* (Instructions Memory) – память команд хранит топологию нейронной сети; так как все командные пакеты нейронной сети реализуют одну базовую функцию формального нейрона, то *IM* содержит только коммуникационную информацию;

- *FM* (Functional Memory) – память функциональных параметров предназначена для долговременного (на срок функционирования нейронной сети) хранения значений весовых коэффициентов и порогов срабатывания формальных нейронов, которые формируются в процессе обучения нейронной сети;

- *AM* (Accumulator Memory) – аккумуляторная память предназначена для накопления значений произведений W_iX_i весовых коэффициентов W_i, ассоциированных со всеми входными значениями X_i, поступившими в локальный пул к некоторому моменту времени;

- *DQ* (Data Queue) – магазинная память, размещаемая на входе локального пула для буферизации входных пакетов данных;

- *RQ* (Results Queue) – магазинная память, размещаемая на выходе локального пула команд для буферизации пакетов данных с результатами обработки.

Если в локальном пуле размещены командные пакеты, входящие в состав одной или нескольких нейронных сетей (нейросетевых пакетных программ), то локальный пул будет находиться в состоянии покоя до тех пор, пока во входную очередь *DQ* не поступит хотя бы один из пакетов данных. Занесение пакета данных в *DQ* по внешнему сигналу *Wr* приведет к формированию внутреннего сигнала управления *PUSH*, который вызовет добавление (если $X_i = 1$) очередного значения весового коэффициента W_i в аккумуляторную память *AM* по адресу, определяемому коммутационным полем входного пакета данных, а именно: адресом командного пакета *AdrCP* и адреса операнда в командном пакете *AdrOp*.

Причем в локальном пуле фиксируется не бинарное значение X_i, а выбранное из того же адресного сечения функциональной памяти *FM* значение весового коэффициента W_i, которое при передаче через блок вентилей *GATES* преобразуется операцией поразрядной конъюнкции X_iW_i и складывается на сумматоре *SUM* с ранее накопленным значением, выбранным из того же адресного сечения аккумуляторной памяти *AM*. Сформированное на выходе сумматора новое значение суммы фиксируется по прежнему адресу в аккумуляторной памяти. Одновременно выполняется установка бита готовности R_i, соответствующего бинарному значению X_i, в выбранном из памяти *RM* текущем значении слова готовности. Новое значение слова готовности маскируется выбранным из того же адресного сечения памяти управления готовностью *CRM* словом маски. Ре-

зультат фиксируется по прежнему адресу в памяти готовности данных *RM* в качестве текущего слова готовности данных. Установка всех битов готовности в некотором адресном сечении памяти готовности данных отслеживается схемой готовности *RS*, которая инициирует выдачу в очередь *RQ* пакета данных, в коммуникационные поля которого заносится из памяти команд *IM* коммуникационная информация о командном пакете–приемниках результата $D_m, \ldots, D_1$, а в поле *OUT* - значение с выхода дискриминатора *COMP*, который выполняет сравнение накопленной в аккумуляторной памяти суммы всех поступивших к данному моменту произведений X_iW_i с порогом срабатывания формального нейрона W_0, выбранного из функциональной памяти *FM*, и формирование бинарного результата *OUT*. После фиксации пакета данных в очереди *RQ* по сигналу *POP* командная ячейка переводится в исходное состояние путем обнуления соответствующего адресного сечения аккумуляторной памяти и того же адресного сечения локальной памяти готовности данных. Выходная магазинная память формирует сигнал «Очередь RQ не пуста» для извлечение пакета данных из локального пула.

Вышерассмотренный процесс обработки информации производится непосредственно в командных ячейках локальных пулов и совмещен во времени с фиксацией во входном стеке *DQ* вновь поступающих пакетов данных. Причем сам процесс обработки информации заключается в циклическом выполнении операций чтения, модификации и записи содержимого памяти готовности данных *RM*, аккумуляторной памяти *AM* и памяти функциональных параметров *FM*, завершение которого контролируется схемой готовности *RS*, переводящей командную ячейку в исходное состояние и разрешающей формирование пакетов данных с результатом преобразования.

Следует обратить внимание, что организация пула команд в виде многофункциональной памяти, управляемой потоками данных, приводит к децентрализации управления, то есть имеют место не только распределенные вычисления, но и локальное распределенное управление.

Детализация описания нейронной сети уровня соответствия *КП–ФН*, когда коммуникационными полями командного пакета задаются все связи между отдельными формальными нейронами сети, приводит к повышенной нагрузке на интерфейс, в который практически одновременно поступает количество пакетов данных, равное числу связей между формальными нейронами соседних слоев нейронной сети. Уменьшить число пакетов данных в интерфейсе возможно за счет описания в коммуникационном поле пакета данных всех связей конкретного формального нейрона-источника.

Адр.входа ФН–приемника 1	...	Адр.входа ФН–приемника *n*	Данные от ФН-источника

Представления объекта-данного в виде суперпакета позволяет в *n* раз, где *n* – число формальных нейронов в принимающем слое, уменьшить количество передаваемых между двумя слоями пакетов данных.

Для иллюстрации последнего тезиса рассмотрим архитектурное решение нейросетевой вычислительной среды с локальными пулами команд (рис. 3.13), в качестве распределительной сети суперпакетов данных в которой использован механизм кольцевой шины.

При вводе исходные данные в виде пакетов данных поступают из устройства ввода *IN* на кольцевую шину в формате

D1	...	*Dn*	*X*

и далее во входные цепи очередей данных *DQ*. Адресные селекторы каждой из очередей *DQ* производят сравнение полей «Адрес пула» - *AdrPool* в адресах командных пакетов-приемников - D_i с номером соответствующего локального пула. В случае совпадения формируются внутренние пакеты данных формата

D_i	*X*

$0 \le i \le n$, и заносятся в соответствующие очереди *DQ*.

Внутренние пакеты данных, находящиеся в началах соответствующих очередей *DQ*, загружаются в командные ячейки локальных пулов команд, вызывая выполнение преобразований поступающих в командные пакеты данных *X* под управлением механизма готовности данных. В процессе распределенных по локальным пулам вычислений асинхронно формируются пакеты данных с результатами преобразований в формате, аналогичном формату исходных данных. Пакеты данных загружаются в очереди результатов *RQ*, откуда поступают на кольцевую шину и далее во входные цепи очередей данных *DQ*.

Процесс вывода результатов преобразований подобен адресной рассылке исходных данных и отличается тем, что совпадение полей *AdrPool* в адресах командных пакетов-приемников D_i происходит не в адресных селекторах очередей *DQ*, а в адресных селекторах блоков вывода данных *OUT*.

«Узким» местом в рассмотренной структуре является адресное распределение пакетов результатов по очередям данных *DQ* локальных пулов команд с использованием механизма кольцевой шины.

Выходом может являться применение в качестве распределительной сети координатного коммутатора, связывающего выходы очередей результатов с входами локальных пулов. Применение подобного коммутатора в интеллектуальной памяти *IRAM* [118] решает проблему развязки множества внутренних потоков многоразрядных данных при выполнении суперскалярных и векторных вычислений в монолитном функционально завершенном устройстве.

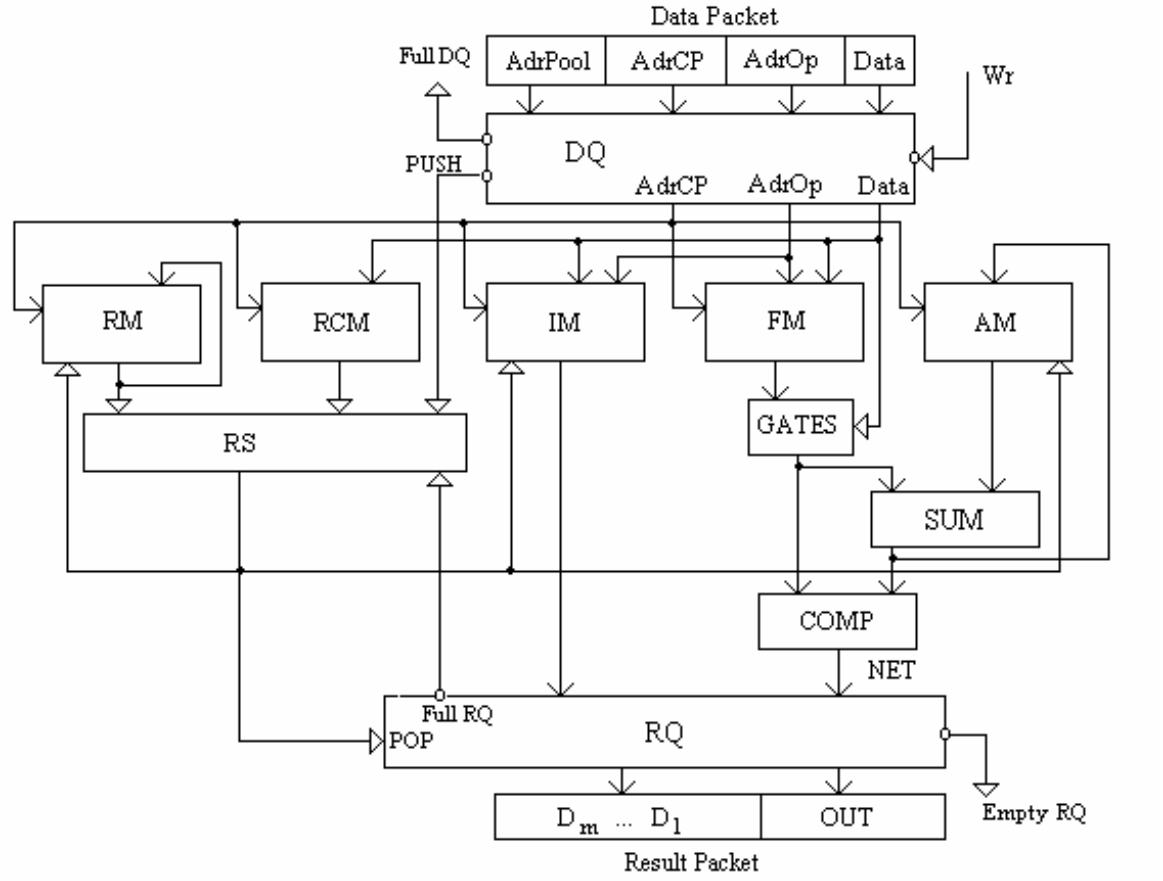

Рис. 3.12. Локальный пул команд

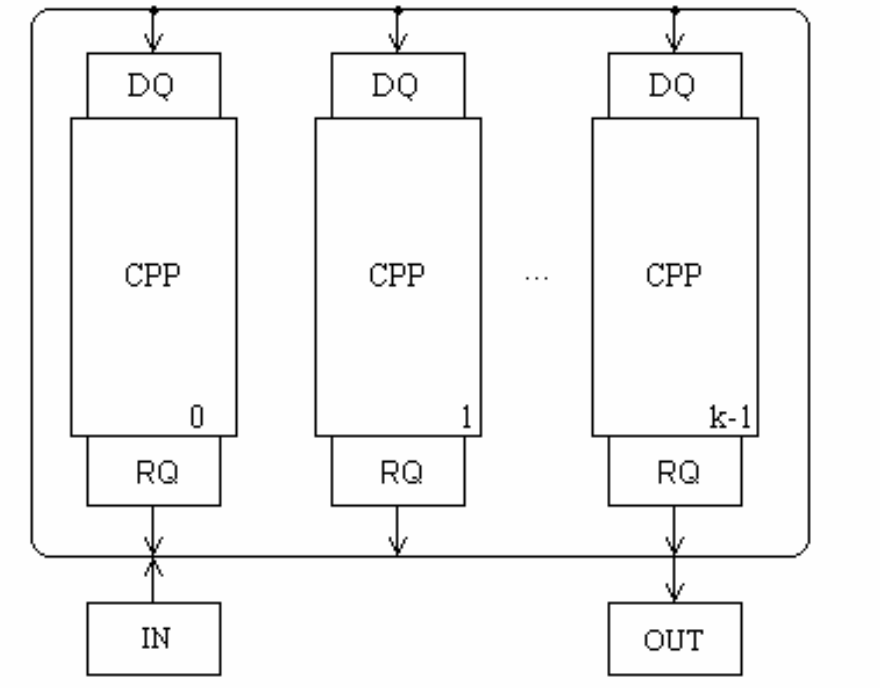

Рис. 3.13. Нейросетевая среда с уровнем детализации *КП-ФН*

3.3.2. Командные пулы уровня слоя формальных нейронов

Для представления нейронных сетей, детализированных до уровня слоя формальных нейронов, следует иметь в виду, что КП описывает операции умножения входного вектора $\boldsymbol{X}$ на матрицу весовых коэффициентов $\boldsymbol{W}$ и нелинейного преобразования φ над координатами выходного вектора $\boldsymbol{NET=XW}$. При этом объекты-данные передаются в виде пакетов данных, число которых в интерфейсе ограничено и равно количеству слоев в представляемой нейронной сети, в формате:

Адрес входа слоя–приемника	Вектор данных от слоя-источника

Рис. 3.14 иллюстрирует вариант построения наращиваемой секции базового блока нейросетевой вычислительной среды, соответствующей случаю обработки командных пакетов последовательно по входам формального нейрона – параллельно по ФН слоя – последовательно по слоям нейронной сети. Базовый блок образован рангом нейропроцессорных узлов *PN*, взаимосвязанных общей магистралью и цепями адресной селекции *DC*. Наращивание функциональной мощности нейросетевого вычислителя возможно за счет увеличения числа формальных нейронов путем соединения секций базовых блоков «по горизонтали» при помощи системы интерфейсных шин адреса *Adr*, данных *Data* и управления *Ctrl*.

Базовый блок может выполнять функцию локального пула команд в следующих случаях: при параллельном выполнении операции слоя формальных нейронов, если число нейронов в слое нейронной сети не превышает количества нейропроцессорных узлов PN; при параллельно-последовательном вычислении выходного вектора слоя формальных нейронов.

Базовый блок образован из следующих модулей памяти:

- *LM* (Links Memory) – память связей - хранит топологию нейронной сети; так как командные пакеты реализуют одну базовую функцию – слоя формальных нейронов, то *LM* содержит только коммуникационную информацию;

- *WM* (Weights Memory) – память весов предназначена для долговременного (на срок работы нейронной сети) хранения значений весовых коэффициентов и порогов срабатывания формальных нейронов, которые формируются в процессе обучения нейронной сети;

- *DQ* (Data Queue) – магазинная память, размещаемая на входе пула команд для буферизации входных пакетов данных;

- *RQ* (Results Queue) – магазинная память, размещаемая на выходе пула для буферизации пакетов данных с результатами обработки.

Если в базовом блоке размещены командные пакеты, входящие в состав одной или нескольких нейросетевых пакетных программ, то пул команд не будет выполнять преобразований информации до тех пор, пока во входную магазинную память не поступит хотя бы один пакет данных с входной шины данных *INBUS* в формате

Adr._Neural_Network	*Adr._Layer*	*N*	*X1*	...	*XN*

Занесение пакета данных в *DQ* приводит к активации цепей адресной селекции *DC*. Если поля адреса *Adr._Neural_Network, Adr._Layer* не соответствуют размещенным в базовом блоке командным пакетам, то формируется внутренний сигнал извлечения ПД из входной очереди. В противном случае запускается цикл обработки пакетов данных во всех *PN* данного базового блока. Сигнал адресной селекции вызывает обнуление аккумуляторов *Ac*, извлечение из модулей *WM* значений порогов срабатывания (смещений) W_{i0} , где i – номер *PN*, и их фиксацию в *Ac*. Затем поле *N* пакета данных, которое задает размерность входного вектора ***X,*** загружается в счетчик *Cnt*, задающий номер входа нейронов слоя нейронной сети. Выходная шина счетчика *Cnt* управляет мультиплексором *MS*, который осуществляет последовательную коммутацию полей X_1 ... X_N магазинной памяти на шину *Data*. Процесс реализации функции формального нейрона происходит путем повторения цикла накопления результата. В соответствии со значением кода адреса, задаваемого на шине *Adr* полями *Adr._Neural_Network, Adr._Layer* и *N*, производится выборка значения очередного весового коэффициента W_{ij}, где j – номер входа формального нейрона, умножение W_{ij} на значение поля X_j в умножителях *Mul* и добавление значений произведения X_jW_{ij} в накапливающие сумматоры, образованные из комбинационных сумматоров *Sum* и аккумуляторов *Ac*. После добавления значений произведения X_jW_{ij} в аккумуляторы *Ac* производится операция декремента счетчика *Cnt* и повторение цикла накопления результата до тех пор, пока счетчик *Cnt* не обнулится. Обнуление *Cnt* разрешает работу модуля памяти связей *LM* и табличных преобразователей *Tab*, реализующих функцию активации φ формального нейрона. В результате модуль выходной магазинной памяти *RQ* фиксирует пакет данных в вышеприведенном формате, который поступает на выходную шину данных *OUTBUS*.

Процесс обработки информации в базовом блоке нейросетевой вычислительной среды производится параллельно во всех *PN* и совмещен во времени с фиксацией во входной очереди *DQ* вновь поступающих пакетов данных. Причем сам процесс обработки информации заключается в циклическом выполнении вышеописанной последовательности операций и завершается по сигналу счетчика *Cnt*. Следует обратить внимание на отсутствие в базовом блоке, как модуля памяти, так и логической схемы готовности данных, которые были не-

обходимы для отслеживания поступления необходимого количества операндов на входы формальных нейронов. Эта функция аппаратно реализуется счетчиком *Cnt*.

Рассмотренное техническое решение оптимально с точки зрения аппаратных затрат в пересчете на формальный нейрон нейронной сети, каждый из которых содержит по одному умножителю, сумматору и аккумулятору; обладает функциональной гибкостью за счет возможности наращивания по «горизонтали» отдельными базовыми блоками, и по «вертикали» - изменяя размерность входного вектора ***X***; характеризуется последовательной обработкой элементов входного вектора ***X***.

Дальнейшее снижение аппаратных затрат в нейросетевой вычислительной среде с уровнем детализации *КП-слой ФН* может быть достигнуто за счет упрощения узла синаптического взвешивания входных значений формальных нейронов, представленных логарифмической моделью [119] (рис. 3.15).

Для взвешивания входных сигналов используется операция сложения логарифмов значений вместо операции умножения самих значений, что эквивалентно замене блока умножения менее ресурсоемкими сумматором и табличным функциональным преобразователем. Согласно логарифмической модели *ФН* при выполнении операции синаптического взвешивания задействованы два нелинейных преобразователя. Первый из них $\varphi(x) = a \ln bx$, $a<1$, $b>1$, $x>0$ (на рис. 3.15 соответствующий блок обозначен *ln x*) размещен на выходе нейропроцессорного блока *PN* (выход аккумулятора *Ac*) и выполняет в нейронной сети функцию масштабирования выходных значений формальных нейронов.

Второй преобразователь реализует функцию $\psi^{-1}(x) = p\, e^{\,mx}$, $p<1$, $m>1$, (на рис. 3.15 соответствующий блок обозначен e^x), с помощью которой в модели *B* решаются две задачи: потенцирование значений логарифма взвешенных значений перед их суммированием в теле формального нейрона и дополнительное нелинейное преобразование $\eta(x) = k\, x^{ma}$, $k = p\, b^{\,ma}$, $ma<1$, которое в рассматриваемой модели формальных нейронов играет роль функции активации. То есть. функция $\eta(x)$, неявно реализуемая в синапсах при суперпозиции функций $\psi^{-1}(\varphi(x))$ за счет соответствующего подбора коэффициентов, переносит основное нелинейное преобразование с выхода на входы формального нейрона.

При переходе к схеме: параллельно по входам формальных нейронов – параллельно по ФН слоя – последовательно по слоям нейронной сети следует ожидать увеличения аппаратных затрат, так как процессы взвешивания элементов входного вектора ***X*** потребуют увеличению числа вышеперечисленных функциональных блоков пропорционально числу входов формального нейрона.

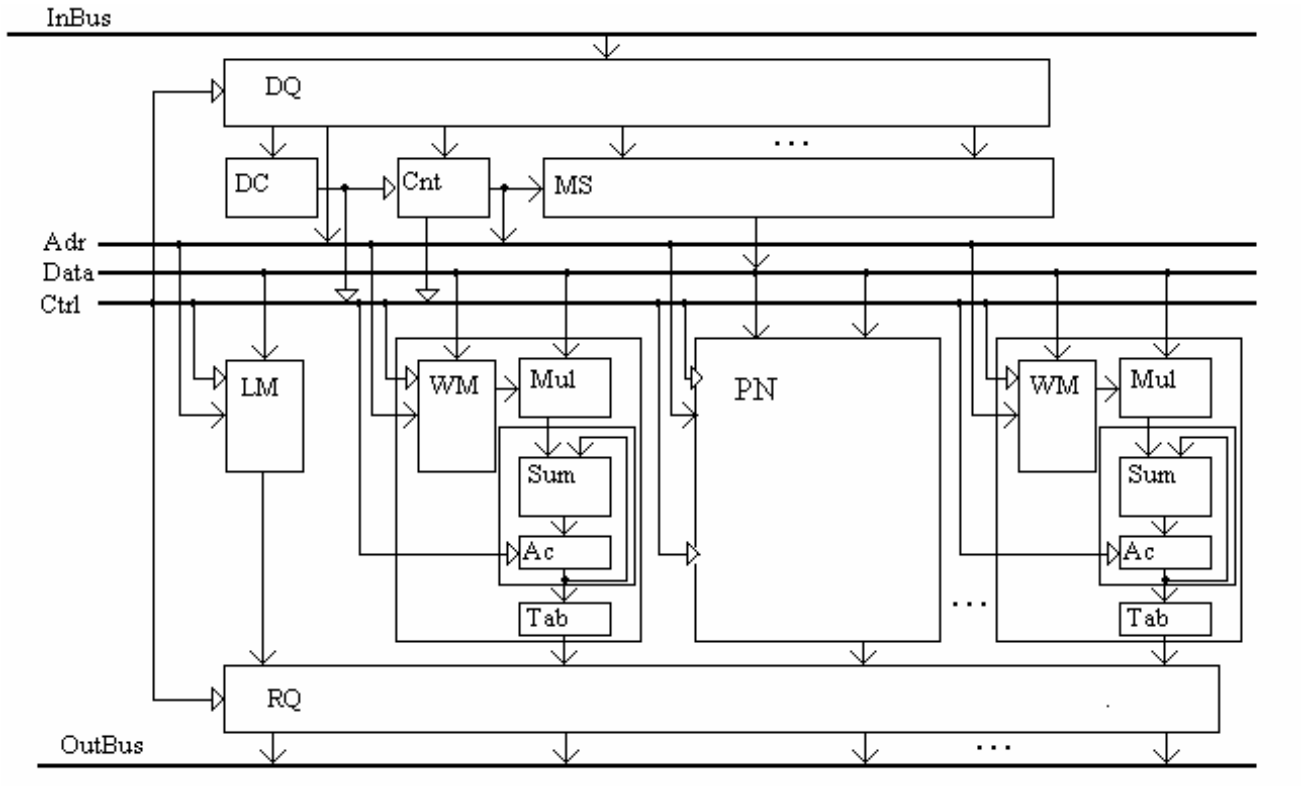

Рис. 3.14. Нейросетевая среда с уровнем детализации *КП-слой ФН*

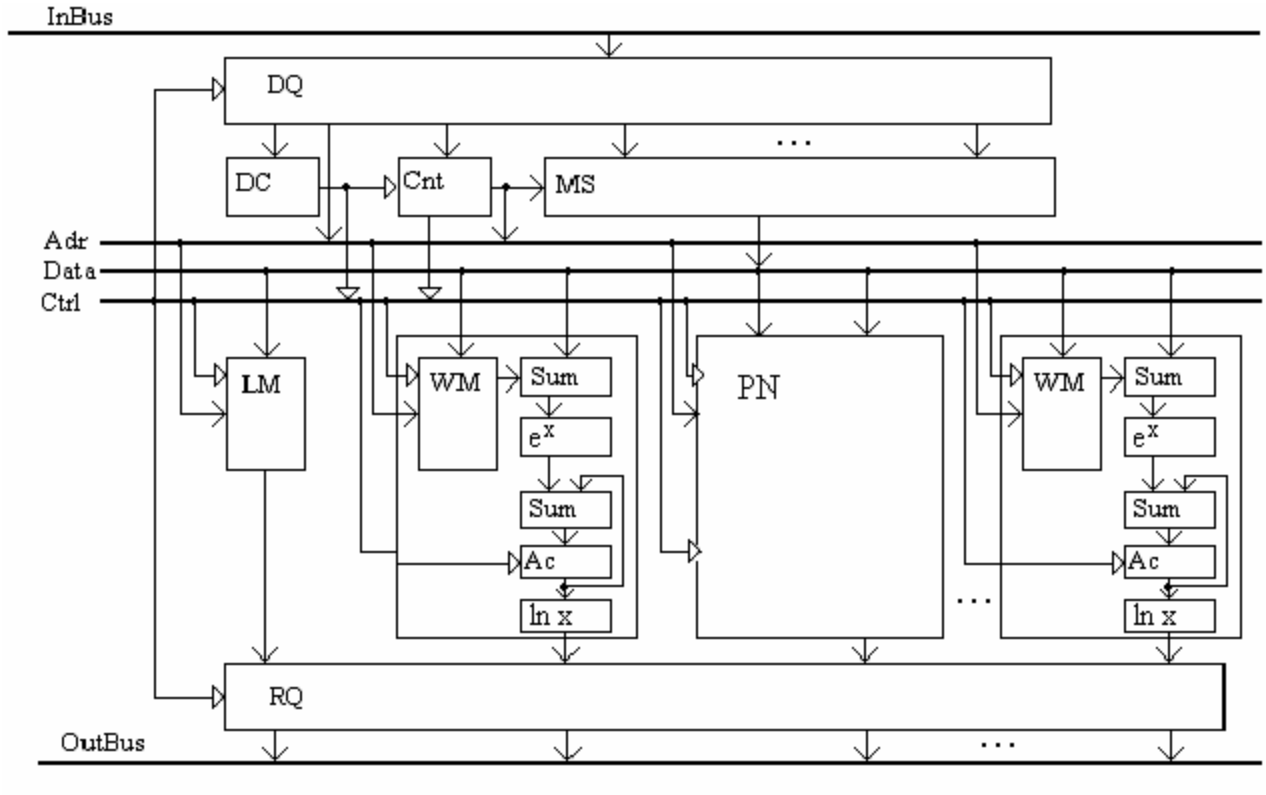

Рис. 3.15. Нейросетевая среда с логарифмическим взвешиванием входов

В этом случае базовый блок нейросетевой вычислительной среды будет представлять собой двумерную систолическую матрицу *PN*, «горизонтальное» измерение которой будет соответствовать числу формальных нейронов слоя нейронной сети, а «вертикальное» – количеству входов формального нейрона.

При выборе архитектуры базового блока нейросетевой вычислительной среды, соответствующей схеме: параллельно по входам формальных нейронов – параллельно по ФН слоя – параллельно по слоям нейронной сети следует ориентироваться на послойную передачу с фиксацией промежуточных результатов вычислений в процессе конвейеризации работы нейронной сети.

Таким образом, объединение функций хранения и обработки информации в многофункциональных пулах упрощает структуру нейросетевой вычислительной среды за счет исключения части коммуникационных цепей, предназначенной для передачи готовых к обработке командных пакетов от локальных пулов команд к процессорным узлам, и соответственно снижает загрузку интерфейса между базовыми блоками. Реализация нейросетевой вычислительной среды из базовых блоков, поддерживающих распределенный характер вычислений, и размещение нейросетевой вычислительной среды в пределах функционально завершенного блока дает возможность снять проблему большой разрядности внешних параллельных шин для передачи, коммутации многоразрядных данных, возникающую при увеличения числа локальных пулов команд, организуемых в регулярных структурах многоблочной памяти с произвольной выборкой. Минимизация потоков данных между базовыми блоками нейросетевой вычислительной среды позволяет использовать простейшие виды интерфейсов для передачи пакетов данных. Последовательный характер реализации функции адаптивного сумматора позволяет совместить операции загрузки пакетов данных в пулы команд с реализацией функций формального нейрона - накопления суммы взвешенных входных сигналов ФН в командных ячейках пула, упрощает формат командного пакета и заменяет большое число операндных полей командного пакета (по числу входов ФН) одним полем накопления суммы взвешенных входных сигналов формального нейрона.

По мере повышения функциональной мощности командных пакетов наблюдается снижение объема передачи пакетов данных и функциональная специализация базовых блоков нейросетевой вычислительной среды. И наоборот, снижение функциональной мощности командных пакетов приводит к универсальности используемых базовых блоков, интенсификации трафика передачи сообщений, что предъявляет повышенные требования к скоростным возможностям коммуникационных цепей нейросетевой вычислительной среды.

Наличие современной технологической базы – необходимое условие для создания функционально мощных базовых блоков – делает целесообразным использование командных пакетов, соответствующих уровню детализации *КП-слой ФН*, *КП-НС*. Для реализации нейросетевой вычислительной среды на базе СБИС с программируемой структурой следует ограничиться уровнем сложности *КП-ФН* или *КП–слой ФН*, а минимизацию информационного потока обеспечивать путем размещения нейронной сети (пакетной программы) в пределах базового блока (ограниченного числа соседних базовых блоков) с целью замыкания передачи промежуточных результатов между слоями или формальными нейронами НС в рамках отдельных СБИС.

3.4. Организация адаптивной СЗИ

Задачи защиты информации в системах ИТ должны решаться комплексно на всех уровнях иерархии системы, как аппаратно, так и программными средствами. Программные или аппаратно-программные средства реализуют методы защиты информации, аналогичные механизму иммунной защиты биосистем, путем прослушивания сообщений, передаваемых по интерфейсу нейросетевой вычислительной среды. Выявление «чужих» сообщений вызывает их изъятие из командных пулов и перевод системы защиты информации в режим адаптации к угрозам.

Биосистемная аналогия в разработке систем ИТ основывается на специфике внутриклеточных механизмов и, прежде всего, информационных свойствах ДНК. Высокая защищенность органической жизни обеспечивается информационной избыточностью и комплементарностью представления данных, равномерностью распределения масс и уравновешенностью системы водородных связей вдоль молекулы ДНК.

Информационная избыточность и комплементарность представления данных наиболее просто реализуются за счет парафазного представления информации в защищаемых полях передаваемых сообщений (в первую очередь коммуникационных полей пакетов данных).

Определенной моделью равномерного распределения масс по коду ДНК можно считать равное число 0 и 1 в коде, приходящихся на единицу длины сообщения, например в байте, слове, и т. п. Парафазное кодирование полей сообщения удовлетворяет указанной модели распределения мольных масс, так как каждому символу x_i, где i – порядковый номер символа в сообщении будет соответствовать пара символов x_i, $\overline{x}_i$, в которой суммарное количество 0 и 1 одинаково.

Некоторым приближением к системе уравновешенных водородных связей в молекуле ДНК можно считать равное число четных и нечетных групп двоич-

ных символов, используемых для представления значений в полях сообщений. В последнем случае также можно использовать парафазное кодирование данных, которое обеспечивает равное число четных и нечетных групп в полях пакетов данных.

Для продолжения аналогии с молекулой ДНК, которая одновременно является и формой представления, и самой информацией, можно рассматривать сообщения, передаваемые по интерфейсу нейросетевой вычислительной среды в виде пакетов команд и данных, в том же двуединстве формы и содержания. Каждый двоичный символ $x_i \in \{0,1\}$ подобного сообщения будет представляться в парафазном виде x_i, $\overline{x}_i$, где i – порядковый номер символа в сообщении, и кодироваться симметричными группами двоичных символов, например, x_i = 01 и $\overline{x}_i$ = 10, или x_i = 00 и $\overline{x}_i$ = 11. Подобное кодирование соответствует всем отмеченным особенностям защищенного представления информации в молекуле ДНК, а именно: избыточностью и комплементарностью представления данных, равномерностью распределения масс и уравновешенностью связей.

Двуединство формы и содержания сообщений выражается в том, что, с одной стороны, информация заключена в форму пакета данных, который однозначно определяет его принадлежность данной системе ИТ («свой» ПД) без введения дополнительных идентифицирующих полей, так как достаточно исследовать любой фрагмент кода пакета на комплементарность, равномерность распределения масс и уравновешенность связей; с другой стороны, используемая система кодирования поместила в форму полей ПД конкретное информационное наполнение, защищенное теми же избыточностью, комплементарностью, равномерностью распределения масс и уравновешенностью связей. В частности, изменение конкретного разряда в каждой отдельной группе x_i, $\overline{x}_i$ (или во всех группах одновременно) достаточно просто аппаратно выявляется и самокорректируется вследствие того, что нарушается, как равенство 0 и 1 в соответствующей группе, так и равенство четных и нечетных последовательностей в пределах группы.

Согласно рассматриваемому подходу система ИТ реализуется в виде единой иерархической адаптивной системы с внутренне присущими функциями защиты; проектирование конкретной системы ИТ осуществляется программной настройкой командных пулов, в процессе которой формируется заданный спецификацией на проектирование и взаимосвязанный интерфейсом набор функциональных устройств, включающий средства защиты информации; при эксплуатации системы ИТ функции отдельных устройств могут изменяться путем адаптации; функции защиты информации распределены по командным пулам и

реализуются на всех уровнях иерархии системы; обмен информацией между функциональными устройствами организуется через интерфейс в виде закодированных сообщений, а информационная защита осуществляется путем проверки передаваемых по интерфейсу сообщений по критерию «свой-чужой» с помощью адаптивных нейросетевых СЗИ.

Адаптивные свойства СЗИ базируются на механизмах нейронных сетей, а обучающим фактором являются присутствующие в данных скрытые закономерности и информационная избыточность. Начальная настройка НС производится на наборе известных угроз, составляющих обучающую выборку входных векторов. Нейронная сеть производит классификацию известных угроз безопасности проектируемой системы, формируя кластеры, соответствующие реальной кластеризации векторов в обучающей выборке через адаптивный подбор числа нейронов-прототипов. Процесс адаптации заключается в сравнении очередного вектора угроз с функциональными параметрами нейронов-прототипов, в результате чего входной вектор либо будет отнесен к одному из известных классов угроз (по критерию близости к функциональным параметрам одного из нейронов-прототипов), либо будет произведено расширение классификации за счет добавления нового нейрона-прототипа с параметрами предъявленного вектора.

Адаптивная нейросетевая защита может быть распределенной по базовым блокам нейросетевой вычислительной среды, либо локализованной в одном из базовых блоков нейросетевой вычислительной среды. В последнем случае адаптивную СЗИ можно обучить, предъявляя в качестве векторов обучающей выборки выходные коды аппаратных схем контроля, проверяющих нарушение комплементарности представления данных, равномерности распределения масс и уравновешенности связей в различных сочетаниях и с различными объемами искажений. В рабочем режиме адаптивных средств защиты информации сформированная при обучении система кластеров либо отнесет поступивший со схем контроля вектор ошибок к уже известным нарушениям в передаваемых по интерфейсу сообщениях (классифицирует вид и степень искажения сообщения и проведет коррекцию своих функциональных параметров), либо создаст новый кластер (нейрон-прототип с параметрами новой угрозы).

Если адаптивная СЗИ распределена по базовым блокам, то помимо уже названных угроз встроенные в блоки средства защиты информации можно обучить классифицировать ситуации попытки адресации к не развернутым в данной нейросетевой вычислительной среде (несуществующим) нейронным сетям, слоям НС, отдельным формальным нейронам, несуществующим входам конкретных формальных нейронов и т. п.

С точки зрения технической реализации (рис. 3.16) структура командных пулов претерпевает минимальные изменения, связанные с необходимостью формирования входного вектора для адаптивных средств защиты информации и выполнения операции параллельного сравнения поступившего входного вектора с функциональными параметрами нейронов–прототипов.

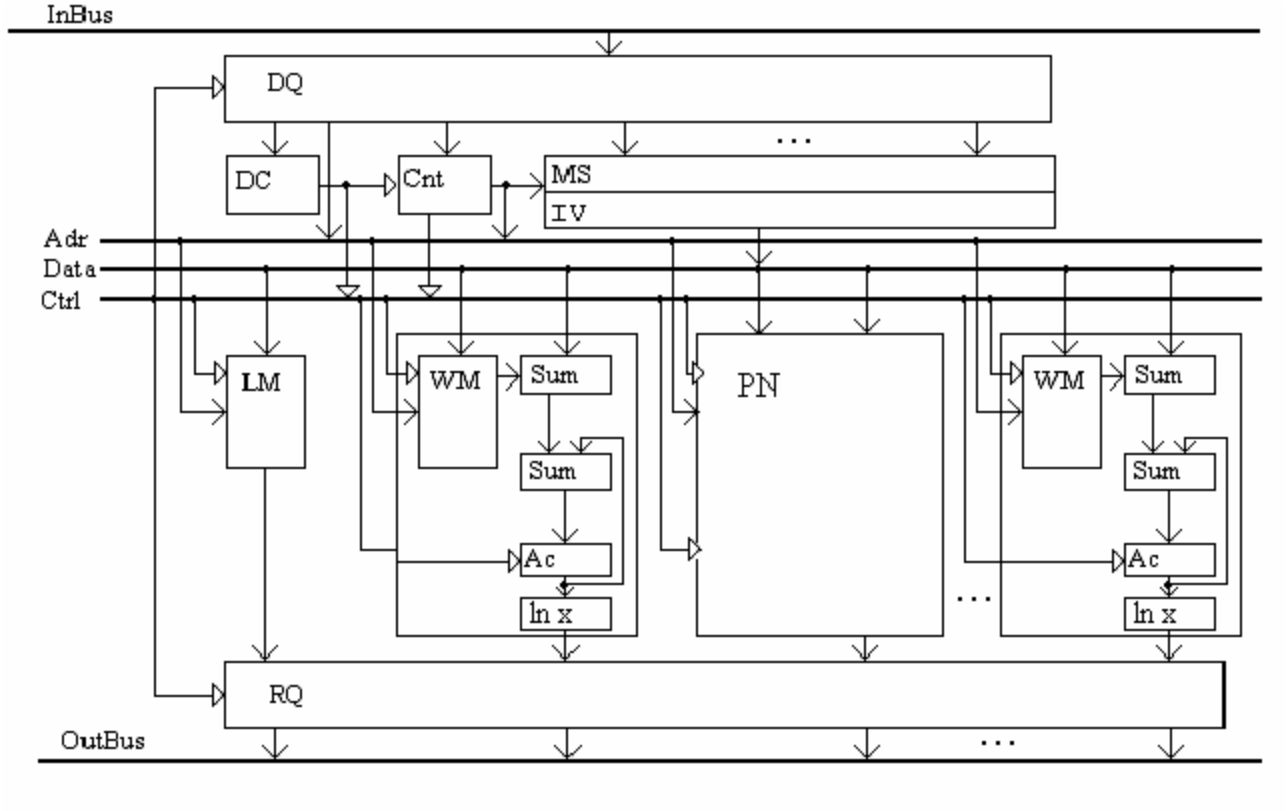

Рис. 3.16. Адаптивная СЗИ, размещенная в командном пуле

По сравнению с базовым блоком, изображенным на рис. 3.15, модернизируются выходные цепи мультиплексора MS за счет размещения аппаратных схем контроля *IV* (Input Vector), призванной выделить распределенную по полям пакетов данных системную информацию о комплементарности, равномерности распределения масс и уравновешенности системы связей.

Кроме того, в структуре нейропроцессорного узла PN следует отключить функциональный преобразователь e^x, так как при выполнении операции сравнения поступившего входного вектора с функциональными параметрами нейронов–прототипов (первый сумматор Sum) отпадает необходимость в умножении входного вектора на вектор весовых коэффициентов. В рассматриваемой структуре PN второй сумматор Sum совместно с аккумулятором Ac используется для накопления значений несовпадений входного вектора с функциональными параметрами каждого из нейронов-прототипов так, что после просмотра всех полей входного вектора на выходах PN сформируется вектор несовпадений, определяющий степень близости входного вектора к каждому из нейронов-прототипов.

Выводы по главе 3

Для реализации адаптивной системы защиты информации разработаны архитектурные решения командных пулов, адаптивная модель СЗИ, инструментальные средства и методика их применения для оптимизации СЗИ по критерию «стоимость/защищенность». Реализация нейросетевых СЗИ базируется на принципах подобия архитектуры и механизмов защиты системы ИТ архитектуре и механизмам защиты биологических систем.

Адаптивность СЗИ обеспечивается использованием элементной базы, способной к обучению, и, прежде всего, нейронных сетей. Нейросетевые СЗИ согласно принципу биосистемной аналогии следует представлять в виде описания структурированных информационных полей иммунного и рецептороного уровней защиты. Показано, что в качестве языковых средств для описания нейросетевых систем защиты информации целесообразно использовать язык пакетных нейросетевых программ. В этом случае НС представляется в виде совокупности взаимосвязанных командных пакетов – ПНП, которая помещается в командных пулах. При описании НС пакетными нейросетевыми программами возможна различная степень детализации: командный пакет может соответствовать одной из функций нейросетевого логического базиса, функции формального нейрона, слоя из формальных нейронов или нейронной сети в целом.

Для адаптивных СЗИ, построенных на базе логарифмической модели формального нейрона, предложен алгоритм обучения по методу обратного распространения ошибки. Процедура обратного распространения ошибки при вычислении поправок к весовым коэффициентам многократно использует операцию умножения. В логарифмической модели формального нейрона (ФН) умножение в процессе взвешивания заменено суммированием. Показано, что при обучении внутренних слоев нейросетевых СЗИ соотношение эффективность обучения логарифмической НС возрастает с увеличением числа слоев и стремится к зависимости, близкой к линейной, с тангенсом угла наклона, равным $\left(\frac{2m}{a}+1\right)$.

Командные пулы организуется в виде многофункциональной регулярной вычислительной структуры - МРВС, в которой размещены пакетные нейросетевые программы. В качестве средства формализации выбран язык графического описания объектов, а в качестве механизма управления вычислениями - способ управления потоком данных. Логика работы памяти в машинах с УПД обеспечивает безопасность хранимой информации: 1) операция записи данных производится не по конкретному адресу памяти, а по содержанию; 2) отсутствует операция считывания данных из ЗУ и, следовательно, непосредственный доступ к хранимой информации. Готовые к обработке данные, представленные в виде пакетов, извлекаются из памяти автоматически - без управления извне.

Отмечено, что объединение функций хранения и обработки информации в многофункциональных пулах упрощает их структуру за счет исключения части коммуникационных цепей, предназначенной для передачи готовых к обработке командных пакетов от локальных пулов команд к процессорным узлам, и снижает загрузку интерфейса. Минимизация потоков данных между командными пулами позволяет использовать простейшие виды интерфейсов для передачи пакетов данных. По мере повышения функциональной мощности командных пакетов наблюдается снижение объема передачи пакетов и функциональная специализация командных пулов. И наоборот, снижение функциональной мощности командных пакетов приводит к универсальности командных пулов, интенсификации трафика передачи сообщений, что предъявляет повышенные требования к скоростным возможностям интерфейса.

Показано, что наличие современной технологической базы делает целесообразным использование командных пакетов, соответствующих уровню детализации КП-слой ФН, КП-НС. Для реализации командных пулов на базе СБИС с программируемой структурой следует ограничиться уровнем сложности КП-ФН или КП–слой ФН, а минимизацию информационного обмена обеспечивать путем размещения пакетной нейросетевой программы в пределах базового блока (ряда базовых блоков) для замыкания информационных потоков между слоями или формальными нейронами НС в рамках отдельных СБИС.

Для реализации в командных пулах адаптивных свойств используются механизмы нейронных и нейро-нечетких сетей, причем средства адаптивной защиты могут быть распределенными по базовым блокам, либо локализованными в отдельном базовом блоке. Предложены варианты реализации адаптивной нейросетевой вычислительной среды и алгоритм обучения нейросетевых СЗИ, построенных на базе логарифмической структурной модели формального нейрона, позволяющий ускорить процессы адаптации в СЗИ за счет исключения «длинных» арифметических операций из итеративной части алгоритма обучения НС.

Заключение

Учебное пособие посвящено решению научно-технической задачи, имеющей существенное значение для обеспечения безопасности систем информационных технологий, используемых в критических приложениях, - задача разработки модели адаптивной системы защиты информации, для которой характерны: использование интеллектуальных механизмов нейронных сетей, нечеткой логики, генетических алгоритмов, разработка комплекса показателей защищенности системы ИТ.

Основные научные и практические результаты, содержащиеся в материалах пособия, заключаются в следующем.

- Предложена модель адаптивной СЗИ, отличающаяся использованием иерархии адаптивных нейронных средств защиты информации, комплекса показателей информационной защищенности системы ИТ, основанного на экспертных оценках, интерактивных инструментальных средств и методик оптимизации распределения механизмов защиты в многоуровневой СЗИ.
- Разработана методика проведения анализа и осуществления развития адаптивной системы защиты информации, отличающийся использованием адаптируемых экспертных оценок, интеллектуальных механизмов нейронных сетей для минимизации соотношения «затраты/защищенность» в СЗИ.
- Предложен комплекс показателей для оценки защищенности СЗИ, отличающийся учетом достоверности активации механизмов защиты, частоты активации угроз, потенциального ущерба от реализации угроз в системе ИТ.

СПИСОК ИСТОЧНИКОВ

1. Кузнецова В. Л., Раков М. А. Самоорганизация в технических системах. – Киев: Наук. думка,1987.

2. Лобашев М. Е. Генетика. – Л.: Изд-во ленинградского университета, 1969.

3. Осовецкий Л. Г., Нестерук Г. Ф., Бормотов В. М. К вопросу иммунологии сложных информационных систем // Изв. вузов. Приборостроение. 2003. Т.46, № 7. С. 34-40.

4. Нестерук Г. Ф., Нестерук Ф. Г. Организация параллельной обработки данных в многофункциональной памяти // Омский научный вестник. 2000. Вып. 10. С.100 - 104.

5. Галушкин А. И. Нейрокомпьютеры и их применение. – М.: ИПРЖР, 2000. - Кн. 3.

6. Осовецкий Л. Г. Научно-технические предпосылки роста роли защиты информации в современных информационных технологиях // Изв. вузов. Приборостроение. 2003. Т.46, № 7. С. 5-18

7. Слива К. Защита будет активной // Computerworld Россия. 2004, № 11. с. 49.

8. Робертс П. Защита на клиенте // Computerworld Россия. 2004, № 16. с. 44.

9. Коржов В. Автоматизация безопасности // Computerworld Россия. 2004, № 17- 18. с. 53.

10. Кеммерер Р., Виджна Дж. Обнаружение вторжений: краткая история и обзор // Открытые системы. 2002, № 7 - 8.

11. Лукацкий А. В. Обнаружение атак. – 2-е изд. – СПб.: БХВ-Петербург, 2003. – 608 с.

12. Милославская Н. Г., Толстой А. И. Интрасети: Доступ в Internet, защита. - М.: ЮНИТИ, 2000.

13. Amoroso E. Intrusion Detection. An Introduction to Internet Surveillance, Correlation, Trace Back, Traps, and Response. Intrusion. Net Books, 1999.

14. Лукацкий А. В. Системы обнаружения атак. "Банковские технологии", 2, 1999. с. 54 -58.

15. Коэн Ф. 50 способов обойти систему обнаружения атак / Пер. с англ. А. В. Лукацкого (http://infosec.ru/pub/pub/13_09.htm).

16. Медведовский И. Д., Платонов В. В., Семьянинов П. В.Атака через Интернет. – СПб.: НПО Мир и семья, 1997.

17. Милославская Н. Г., Тимофеев Ю. А., Толстой А. И. Уязвимость и методы защиты в глобальной сети Internet. – М.: МИФИ, 1997.

18. Вакка Дж. Секреты безопасности в Internet. – Киев: Диалектика, 1997.

19. Зегжда Д. П., Мешков А. В., Семьянов П. В., Шведов Д. В. Как противостоять вирусной атаке. – СПб.: BHV, 1995.

20. Tan K. The Application of Neural Networks to UNIX Computer Security //Proc. of the IEEE International Conf. on Neural Networks, 1995. V.1. P. 476-481.

21. Корнеев В. В., Маслович А. И. и др. Распознавание программных модулей и обнаружение несанкционированных действий с применением аппарата нейросетей // Информационные технологии, 1997. №10.

22. Porras P. A., Ilgun K., and Kemmerer R. A. State transition analysis: A rule-based intrusion detection approach. // IEEE Trans. on Software Engineering, 1995. SE-21. P. 181 – 199.

23. Ивахненко А.Г., Ивахненко Г.А., Савченко Е.А., Гергей Т. Самоорганизация дважды многорядных нейронных сетей для фильтрации помех и оценки неизвестных аргументов // Нейрокомпьютеры: разработка и применение. 2001, № 12.

24. Ивахненко А.Г., Савченко Е.А., Ивахненко Г.А., Гергей Т., Надирадзе А.Б., Тоценко В.Г. Нейрокомпьютеры в информационных и экспертных системах // Нейрокомпьютеры: разработка и применение. 2003, № 2.

25. Корнеев В. В., Васютин С. В. Самоорганизующийся иерархический коллектив экспертов // Нейрокомпьютеры: разработка и применение. 2003, № 2.

26. Helman P., Liepins G., Richards W. Foundations of Intrusion Detection // Proc. of the 15th Computer Security Foundations Workshop. 1992. P. 114-120.

27. Ryan J., Lin M., Miikkulainen R. Intrusion Detection with Neural Networks. AI Approaches to Fraud Detection and Risk Management: Papers from the 1997 AAAIWorkshop (Providence, Rhode Island), pp. 72-79. Menlo Park, CA: AAAI. 1997.

28. Bace R. An Introduction to Intrusion Detection Assessment for System and Network Security Management. 1999.

29. Kumar S., Spafford E. A Pattern Matching Model for Misuse Intrusion Detection // Proc. of the 17th National Computer Security Conference. 1994. P. 11-21.

30. Allen J., Christie A., Fithen W., McHugh J., Pickel J., Stoner E. State of the Practice of Intrusion Detection Technologies. Carnegie Mellon University. Networked Systems Survivability Program. Technical Report CMU/SEI-99-TR-028 ESC-99-028. 2000, January.

31. Denning D. E. An intrusion detection model // IEEE Trans. on Software Engineering, 1987, SE-13. P. 222–232.

32. Garvey T. D. Lunt T. F. Model-based intrusion detection // Proc. of the 14th National Computer Security Conference. 1991.

33. Teng H. S., Chen K., Lu S. C. Adaptive real-time anomaly detection using

inductively generated sequential patterns // Proc. of the IEEE Symposium on Research in Computer Security and Privacy.1990. P. 278–284.

34. Червяков Н. И., Малофей О. П., Шапошников А. В., Бондарь В. В. Нейронные сети в системах криптографической защиты информации // Нейрокомпьютеры: разработка и применение. 2001, № 10.

35. Fu L. A Neural Network Model for Learning Rule-Based Systems // Proc. of the International Joint Conference on Neural Networks. 1992. I. P. 343-348.

36. Negneyitsky M. Artificial intelligence: a guide to intelligent systems. Addison-Wesley, 2002.

37. Фатеев В. А., Бочков М. В. Методика обнаружения несанкционированных процессов при выполнении прикладных программ, основанная на аппарате скрытых марковских цепей // // Сб. докл. VI Междунар. конф. SCM'2003. – СПб.: СПГЭТУ, 2003. т. 1. С. 218-220.

38. Бочков М. В., Копчак Я. М. Метод идентификации вычислительных сетей при ведении компьютерной разведки // Сб. докл. VI Междунар. конф. SCM'2003 – СПб.: СПГЭТУ, 2003. т. 1. С.288-290.

39. Зима В. М., Молдовян А. А., Молдовян Н. А. Безопасность глобальных сетевых технологий. – СПб.: Изд-во БХВ-Петербург, 2003.

40. Штрик А. А., Осовецкий Л. Г., Месих И. Г. Структурное проектирование надежных программ встроенных ЭВМ. - М.: Москва, 1986.

41. Игнатьев М. Б., Фильчаков В. В., Осовецкий Л. Г. Активные методы обеспечения надежности алгоритмов и программ. - СПб.: Политехника, 1992.

42. Липаев В. В., Филинов Е. Н. Мобильность программ и данных в открытых информационных системах, М.,1997.

43. Зегжда П. Д., Зегжда Д. П., Семьянов П. В., Корт С. С., Кузьмич В. М., Медведовский И. Д., Ивашко А. М., Баранов А. П. Теория и практика обеспечения информационной безопасности. – М.: Яхтсмен, 1996.

44. Городецкий В. И., Карсаев О. В., Котенко И. В. Программный прототип многоагентной системы обнаружения вторжений в компьютерные сети // ICAI'2001. Международный конгресс "Искусственный интеллект в XXI веке". Труды конгресса. Том 1. М.: Физматлит, 2001.

45. Котенко И. В., Карсаев А. В., Самойлов В. В. Онтология предметной области обучения обнаружению вторжений в компьютерные сети // Сб. докл. V Междунар. конф. SCM'2002. – СПб.: СПГЭТУ, 2002. т. 1. С. 255-258.

46. Алексеев А. С., Котенко И. В. Командная работа агентов по защите от распределенных атак "отказ в обслуживании" // Сб. докл. VI Международной конф. по мягким вычислениям и измерениям SCM'2003. – СПб.: СПГЭТУ, 2003. т. 1. С. 294 -297.

47. Котенко И. В. Модели противоборства команд агентов по реализации и защите от распределенных атак «Отказ в обслуживании» // Тр. междунар. научно-технич. конф. IEEE AIS'03 и CAD-2003. – М.: Физматлит, 2003. т. 1. С. 422 - 428.

48. Городецкий В. И., Котенко И. В. Командная работа агентов-хакеров: применение многоагентной технологии для моделирования распределенных атак на компьютерные сети // КИИ-2002. VIII Национальная конференция по искус-

ственному интеллекту. Труды конференции. М.: Физматлит, 2002.

49. Городецкий В. И., Котенко И. В. Командная работа агентов в антагонистической среде // Сб. докл. V Междунар. конф. SCM'2002. – СПб.: СПГЭТУ, 2002. т. 1. С. 259-262.

50. Котенко И. В, Степашкин М. В. Интеллектуальная система моделирования атак на web-сервер для анализа уязвимостей компьютерных систем // Сб. докл. VI Международной конф. по мягким вычислениям и измерениям SCM'2003. – СПб.: СПГЭТУ, 2003. т. 1. С. 298-301.

51. Gorodetski V., Kotenko I. Attacks against Computer Network: Formal Grammar-based Framework and Simulation Tool // Recent Advances in Intrusion Detection. Switzerland. Proceedings. Lecture Notes in Computer Science, V.2516. 2002.

52. Степашкин М. В., Котенко И. В. Классификация атак на Web-сервер // VIII Санкт-Петербургская Международная Конференция "Региональная информатика-2002" Материалы конференции. Ч. 1. СПб., 2002.

53. Пантелеев С. В. Решение задач идентификации динамических объектов с использование нейронных сетей // Сб. докл. VI Международной конф. SCM'2003. – СПб.: СПГЭТУ, 2003. т. 1. С. 334-336.

54. Веселов В.В., Елманов О.А., Карелов И.Н. Комплекс мониторинга информационных систем на основе нейросетевых технологий // Нейрокомпьютеры: разработка и применение. 2001, № 12.

55. Гриняев С. Н. Интеллектуальное противодействие информационному оружию. М.: СИНТЕГ, 1999.

56. Осипов В. Ю. Концептуальные положения программного подавления вычислительных систем // Защита информации. Конфидент. 2002. № 4-5. С. 89–93.

57. Бочков М. В., Крупский С. А., Саенко И. Б. Применение генетических алгоритмов оптимизации в задачах информационного противодействия сетевым атакам. // Управление и информационные технологии. Всерос. науч. конф.. Сб. док. Том 2. - СПб.: ЛЭТИ, 2003. С.13-16.

58. Бочков М. В. Реализация методов обнаружения программных атак и противодействия программному подавлению в компьютерных сетях на основе нейронных сетей и генетических алгоритмов оптимизации // Сб. докл. VI Межд. конф. по мягким вычислениям и измерениям SCM'2003. – СПб.: СПГЭТУ, 2003. т. 1. С. 376 - 378.

59. Бочков М. В., Логинов В. А., Саенко И. Б. Активный аудит действий пользователей в защищенной сети // Защита информации. Конфидент. 2002, № 4-5. С.94-98.

60. Логинов В. А. Методика активного аудита действий субъектов доступа в корпоративных вычислительных сетях на основе аппарата нечетких множеств // Сб. докл. VI Междунар. конф. SCM'2003. – СПб.: СПГЭТУ, 2003. т. 1. С. 240-243.

61. Головань А. В., Шевцова Н. А., Подладчикова Л. Н., Маркин С. Н., Шапошников Д. Г. Детектирование информативных областей лиц с помощью локальных признаков // Нейрокомпьютеры: разработка и применение. 2001, № 1.

62. Макаревич О. Б., Федоров В. М., Тумоян Е. П. Применение сетей функций радиального базиса для текстонезависимой идентификации диктора // Нейрокомпьютеры: разработка и применение. 2001, № 7-8.

63. Юрков П. Ю., Федоров В. М., Бабенко Л. К. Распознавание фонем русского

языка с помощью нейронных сетей на основе вейвлет-преобразования // Нейрокомпьютеры: разработка и применение. 2001, № 7-8.

64. Гузик В. Ф., Галуев Г. А., Десятерик М. Н. Биометрическая нейросетевая система идентификации пользователя по особенностям клавиатурного почерка // Нейрокомпьютеры: разработка и применение. 2001, № 7-8.

65. Бабенко Л. К., Макаревич О. Б., Федоров В. М., Юрков П. Ю. Голосовая текстонезависимая система аутентификации идентификации пользователя // Нейрокомпьютеры: разработка и применение. 2003, № 10-11.

66. Бабенко Л. К., Макаревич О. Б., Федоров В. М., Юрков П. Ю. Голосовая текстонезависимая система аутентификации/идентификации пользователя // Нейрокомпьютеры: разработка и применение. 2003, № 12.

67. Кулик С. Д. Биометрические системы идентификации личности для автоматизированных фактографических информационно-поисковых систем // Нейрокомпьютеры: разработка и применение. 2003, № 12.

68. Норткатт С. Анализ типовых нарушений безопасности в сетях. М.: Издательский дом «Вильямс», 2001.

69. Норткат С., Новак Дж. Обнаружение вторжений в сеть.: Пер. с англ. – М.: Издательство «ЛОРИ», 2001. – 384с.

70. Скотт Хокдал Дж. Анализ и диагностика компьютерных сетей.: Пер. с англ. – М.: Издательство «ЛОРИ», 2001. – 354с.

71. Стенг Д., Мун С. Секреты безопасности сетей. - Киев: Диалектика, 1996.

72. Understanding DDOS Attack, Tools and Free Anti-tools with Recommendation. SANS Institute. April 7, 2001.

73. Noureldien A. N. Protecting Web Servers from DoS/DDoS Flooding Attacks. A Technical Overview. International Conference on Web-Management for International Organisations. Proceedings. Geneva, October, 2002.

74. Tambe M., Pynadath D. V. Towards Heterogeneous Agent Teams // Lecture Notes in Artificial Intelligence. V.2086, Springer Verlag, 2001.

75. Осовецкий Л. Г., Нестерук Г.Ф., Куприянов М.С., Нестерук Ф. Г. Иммунология сложных вычислительных систем // Труды 8-го междунар. НПС "Защита и безопасность вычислительных технологий ". - СПб, 2002. С. 18 - 25.

76. Fuller R. Neural Fuzzy Systems. - Abo: Abo Akademi University, 1995.

77. Круглов В.В. Нечеткая игровая модель с единичным экспериментом // Нейрокомпьютеры: разработка и применение. 2003, № 8-9.

78. Усков А.А. Адаптивная нечеткая нейронная сеть для решения задач оптимизации функционалов // Нейрокомпьютеры: разработка и применение. 2003, № 12.

79. Нестерук Г. Ф., Куприянов М. С., Нестерук Л. Г. О реализации интеллектуальных систем в нечетком и нейросетевом базисах // Сб. докл. VI Междунар. конф. SCM’2003. – СПб.: СПГЭТУ, 2003. т. 1. С. 330-333.

80. Nesteruk G. Ph., Kupriyanov M. C. Neural-fuzzy systems with fuzzy links // Proc. of the VI-th Int. Conference SCM’2003. – СПб.: СПГЭТУ, 2003. т. 1. С. 341-344.

81. Мелик-Гайназян И. В. Информационные процессы и реальность. М.: Наука, 1998. - 192 с.

82. Нестерук Ф. Г., Нестерук Г. Ф., Харченко А. Ф. Моделирование адаптивных процессов защиты информационных ресурсов экономических объектов // Сб. докл. Междунар. НПК «Глобальные тенденции в статистике и математических методах в экономике». - СПб, 2004. С. 218-220.

83. Нестерук Г. Ф., Осовецкий Л. Г., Нестерук Ф. Г. Адаптивная модель нейросетевых систем информационной безопасности // Перспективные информационные технологии и интеллектуальные системы. 2003, № 3.

84. Nesteruk Ph., Kharchenko A., Nesteruk G. Information safety in electronic business: adaptive model of systems safety of information technologies // Proc. of Int. Conf. “Information technology in business” (St. Petersburg, October 8-10, 2003) - St. Petersburg, 2003. P. 124-128.

85. Карпычев В.Ю., Минаев В.А. Цена информационной безопасности // Системы безопасности. 2003, № 5. С.128 - 130.

86. Касперски К. Атака на Windows NT. Вкладка «Обзор антивирусных средств от AIDSTEST до информационной иммунной системы» // LAN / Журнал сетевых решений. 2000, декабрь, С. 88 - 95.

87. Яковлев Н. Н. Жизнь и среда: Молекулярные и функциональные основы приспособления организма к условиям среды. – Л.: Наука, 1986.

88. Девянин П.Н. и др. Теоретические основы компьютерной безопасности. - М.: «Радио и Связь» - 2000.

89. Осовецкий Л., Шевченко В. Оценка защищенности сетей и систем // Экспресс электроника. 2002. № 2-3. С.20-24.

90. Жижелев А. В., Панфилов А. П., Язов Ю. К., Батищев Р. В. К оценке эффективности защиты информации в телекоммуникационных системах посредством нечетких множеств // Изв. вузов. Приборостроение. 2003. т. 46, № 7. С. 22-29.

91. Нестерук Л. Г., Нестерук Ф. Г. Нечеткое представление экономической информации в нейронных сетях // Труды 8-го международного научно-практического семинара "Защита и безопасность информационных технологий". - СПб, 2002. С. 68-74.

92. Нестерук Г. Ф., Осовецкий Л. Г., Нестерук Ф. Г. К оценке защищенности систем информационных технологий // Перспективные информационные технологии и интеллектуальные системы. 2004, № 1 (17). С. 31-41.

93. Корнеев В. В., Гареев А. Ф., Васютин С. В., Райх В. В. Базы данных. Интеллектуальная обработка информации. – М.: Нолидж, 2001.

94. Асаи К., Ватада Д., Иваи С. и др. Прикладные нечёткие системы. / Под ред. Т. Тэрано, К. Асаи, М. Сугэно.– М.: Мир, 1993.

95. Дрожжинов В., Штрик А. ИКТ в обществе // PC WEEK/RE. 2005, № 3.

96. Rodriguez F., Wilson E. J. Are Poor Countries Losing the Information Revolution? // InfoDev Working Paper. May 2000. - University of Maryland at College Park.

97. Science and Engineering Indicators – 2000. National Science Foundation (NSF). - (http://www.nsf.gov/sbe/srs/seind00/frame.htm)

98. Information Society Index // WorldPaper. January 2001. - (http://www.worldpaper.com)

99. Девянин П.Н. и др. Теоретические основы компьютерной безопасности. - М.: «Радио и Связь» - 2000.

100. ГОСТ / ИСО МЭК 15408 – 2002 «Общие критерии оценки безопасности информационных технологий».

101. Нестерук Г. Ф., Куприянов М. С., Елизаров С. И. К решению задачи нейро-нечеткой классификации // Сб. докл. VI меж. конф. SCM'2003. – СПб.: СПГЭТУ, 2003. т. 1. С. С. 244 - 246.

102. Круглов В. В., Борисов В. В. Искусственные нейронные сети. Теория и практика.- 2-е изд., стереотип. – М.: Горячая линия - Телеком, 2002.

103. Применение ПЛИС XILINX для построения нейронных сетей. – Scan Eng.Telecom, 1999.

104. Компьютеры на СБИС: В 2-х кн. Кн. 1 / Мотоока Т., Томита С.. и др. – М.: Мир, 1988.

105. Балашов Е. П., Смолов Б. В., Петров Г. А., Пузанков Д. В. Многофункциональные регулярные вычислительные структуры. - М.: Сов. Радио, 1978.

106. Нестерук Ф. Г. Безопасное хранение данных в нейросетевых информационных системах // Изв. вузов. Приборостроение. 2003. Т.46, № 7. С. 52-57.

107. Нейроинформатика. / А. Н. Горбань, В. Л. Дунин-Барковский, А. Н. Кирдин и др. - Новосибирск: Наука. Сиб. отд-ние, 1998.

108. Патент 2179739 РФ, МПК G 06 F 15/00. Устройство для обработки информации. / Г. Ф. Нестерук, Ф. Г. Нестерук. - № 2000108883/09; Заявлено 10.04.2000; Опубл. 20.02.2002. Бюл. № 5. Приоритет от 10.04.2000. 4 с.

109. Backus J. Can programming be liberated from the von Neumann style? A functional style and its algebra of programs // Communications of the ACM. 1978. № 21(8). P. 613- 641.

110. Arvind A. Critique of multiprocessing von Neumann style // Proc. of 10th Annual Int. Symp. on Computer Architecture. 1983. P. 426-436.

111. Mayers G. J. Advances in computer architecture. 2nd edition. - JONH WILLEY & SONS. 1982.

112. Dennis J. B., Misunas D. P. A preliminary architecture for basic data flow processor // Proc. of 2nd annual Int. Symp. on Computer Architecture. – N.Y.1975. P. 126 -132.

113. Misunas D. P. A computer architecture for data-flow computation // Laboratory for Computer Science. MIT. – Cambridge. MA. 1978.

114. Ackerman W. B. Data flow languages // Proc. of the NCC, Montvale. - NJ, AFIPS. 1979. P. 1087—1095.

115. McGraw J. R. Data flow computing, software development // Proc. of the Int. Conf. on Distributed Computing Systems. – N.Y., IEEE. 1979. P. 242—251.

116. Нестерук Г. Ф., Куприянов М. С., Нестерук Ф. Г. О разработке языковых средств для программирования нейросетевых структур // Сб. докл. V междунар. конф. SCM'2002. - СПб, 2002, Т.2.С. 52-55.

117. Куприянов М. С., Нестерук Г. Ф., Пузанков Д. В. Реализация мягких вычислений в распределенных системах // Изв. СПбГЭТУ «ЛЭТИ»: Серия «Информатика, управление и компьютерные технологии». – СПб.: СПбГЭТУ «ЛЭТИ», 2002. Вып.1. С. 34 - 39.

118. Нестерук Ф. Г., Осовецкий Л. Г., Нестерук Г. Ф., Воскресенский С.И. К моделированию адаптивной системы информационной безопасности // Перспективные информационные технологии и интеллектуальные системы. 2004, № 4, С.25 - 31.