

Введение

Применение нейронных сетей (НС) для решения задач защиты информации связано, в первую очередь, с интеллектуальным анализом и предсказанием *временных рядов* (например, динамики трафика защищаемой локальной сети), а также *поиском скрытых закономерностей* в массивах первичных данных посредством средств data mining and knowledge engineering [1].

Лабораторные работы преследуют цель ознакомления с методом поиска скрытых закономерностей в массивах первичных данных, рассматриваемых в качестве обучающей выборки, образованной из множества пар векторов $\{X, Y\}$ – входного X и выходного Y , которые используют для обучения НС, состоящей из слоев формальных нейронов (ФН).

1. Теоретическая часть

Рассмотрим нейронную сеть, описываемую конфигурацией 2 – 1 – 2 – 1 (рис. 1) [2], то есть имеющую два входа во входном слое (input layer), один скрытый слой (hidden layer), содержащий два формальных нейрона (ФН), и один ФН в выходном слое (output layer).

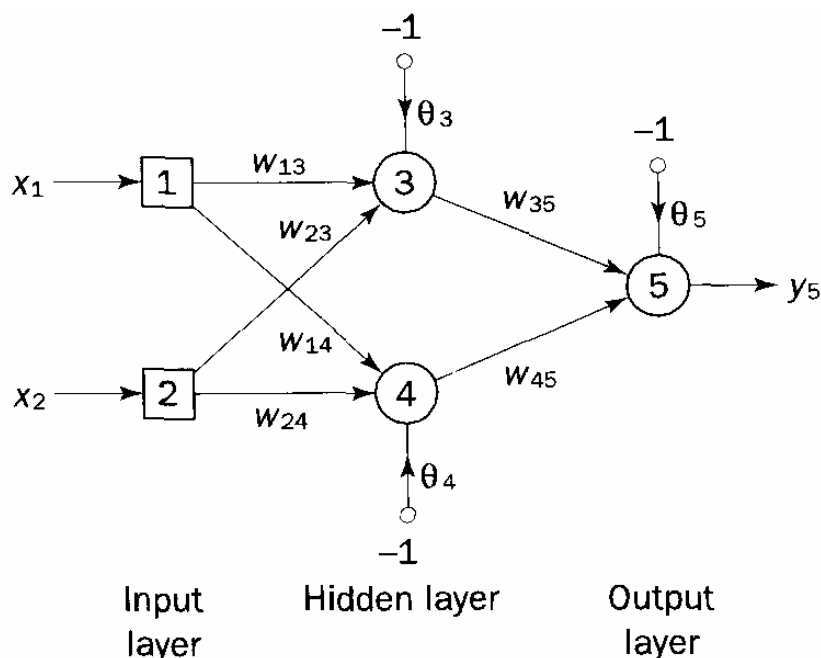


Рис. 1. Структура нейронной сети 2-1-2-1

Входной вектор X составлен из координат (x_2, x_1) , а выходной вектор Y – из координаты y_5 . Входы НС, отмеченные символами «-1», соответствуют смещениям (порогам) ФН. Входной слой нейронной сети (узлы 1 и 2 на рис. 1) вы-

полняет функции репликации (дублирования) координат входного вектора по числу ФН в скрытом слое. Узлы 3 – 5 являются ФН.

Формальный нейрон представляет собой упрощённую модель биологического нейрона.

1.1. Биологический нейрон

Нейрон (от греч. *νευρον* — нерв) представляет собой нервную клетку, которая способна обрабатывать информацию (рис. 2). Она состоит из тела клетки - сомы (*soma*), и двух типов отростков: аксона (*axon*) и дендритов (*dendrites*). Сомма включает ядро (*nucleus*), содержит информацию о наследственных свойствах, и плазму, обладающую молекулярными средствами для жизнедеятельности нейрона.

Нейрон получает сигналы через дендриты (приемники), суммирует их, сравнивает с пороговым значением (смещением) и в случае превышения порога формирует сигналы, сгенерированные телом клетки, на аксон (передатчик), который в конце разветвляется на волокна (*strands*).

На окончаниях волокон находятся синапсы (*synapses*) - связующие звенья между аксоном одного нейрона и дендритами других. Веса синапсов и смещения настраиваются в зависимости от активности происходящих в НС процессов.

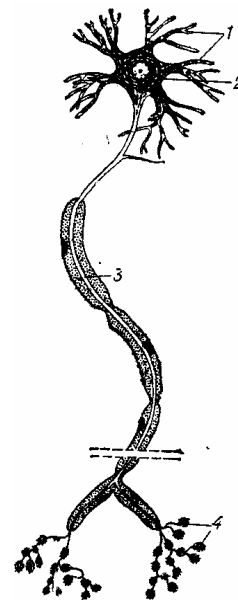


Рис. 2. Схема нейрона:
1 — дендриты; 2 — тело;
3 — аксон; 4 — синапсы

1.2. Формальный нейрон – математическая модель нейрона

В формальном нейроне веса связей соответствуют синапсам, а функция *NET* отражает активность сомы. На *i*-е входы ФН поступают сигналы в виде вектора X , $i = \overline{1, n}$, n – размерность вектора X , который скалярно умножается на вектор весов W , соответствующий синаптической силе входов: $NET = XW$. Подключение к выходу ФН нелинейного преобразователя $\varphi(x)$ сужает диапазон выходных значений и расширяет функциональные возможности ФН (рис. 3) [3].

К базовым элементам НС относятся *система межнейронных связей* (на рис. 1 обозначены символами w_{ij}), выполняющая функцию взвешивания $x_i w_i$

сигналов, а также *формальные нейроны* (на рис. 1 обозначены окружностями), реализующие функцию

$$OUT = \varphi \left(\sum_{i=1}^n x_i w_{ij} + \theta_j \right),$$

как суперпозицию функции адаптивного сумматора $NET = \sum_{i=1}^n x_i w_{ij} + \theta_j$ и функции «сжимающего» нелинейного преобразователя $OUT = \varphi(NET)$.

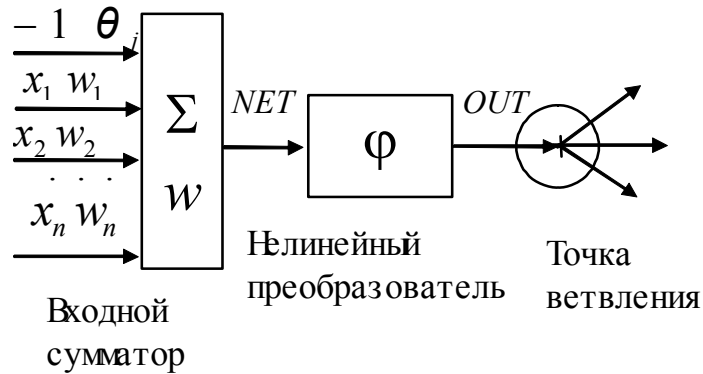


Рис. 3. Схема формального нейрона

В качестве «сжимающей» функции часто используется сигмоидальная (S-образная) функция $\varphi(x) = 1/(1 + e^{-x})$ [4]. На рис. 4 изображена «сжимающая»

функция $OUT = \frac{1}{1 + e^{-NET}}$.

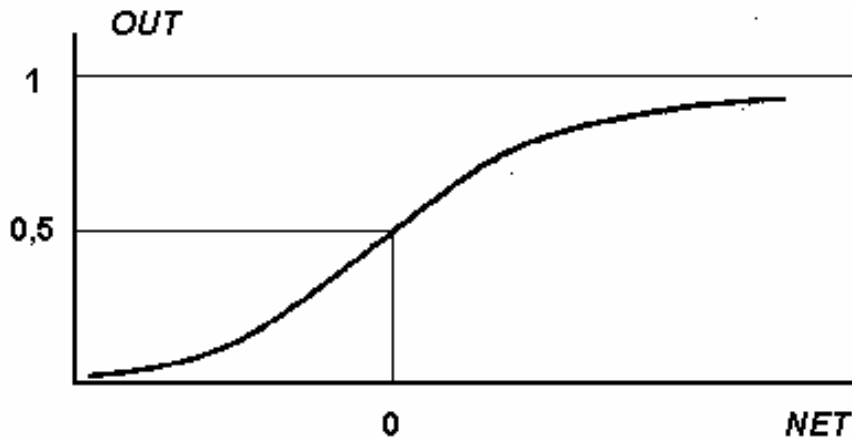


Рис. 4. График сигмоидальной функции

Функцию $\varphi(x)$ можно считать нелинейной усилительной характеристикой формального нейрона. Коэффициент усиления вычисляется как отношение приращения величины OUT к вызвавшему его небольшому приращению величины NET.

Для иллюстрации процесса интеллектуального анализа данных с помощью нейронной сети будем использовать ряд логических функций двух переменных,

а именно: И-НЕ (NAND), ИЛИ (OR), ИЛИ-НЕ (NOR), Равнозначность (XOR), Неравнозначность (NXOR), представленные в табл. 1.

Таблица 1

Входы		Логические функции					
x_2	x_1	AND	NAND	OR	NOR	XOR	NXOR
0	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	0	1	0	0	1

Необходимо, чтобы НС (рис. 1), обученная с помощью массива первичных данных, содержащего информацию об одной из логических функций, выявляла скрытую закономерность – вид соответствующей логической функции.

Для выбора одной из угроз ($Y_1 - Y_4$), исходя из значений совокупности признаков атаки (x_1, x_2), можно использовать НС с конфигурацией, например, 2 – 1 – 5 – 4. То есть в НС имеется два входа во входном слое, один скрытый слой, содержащий пять ФН, и четыре ФН в выходном слое (рис. 5).

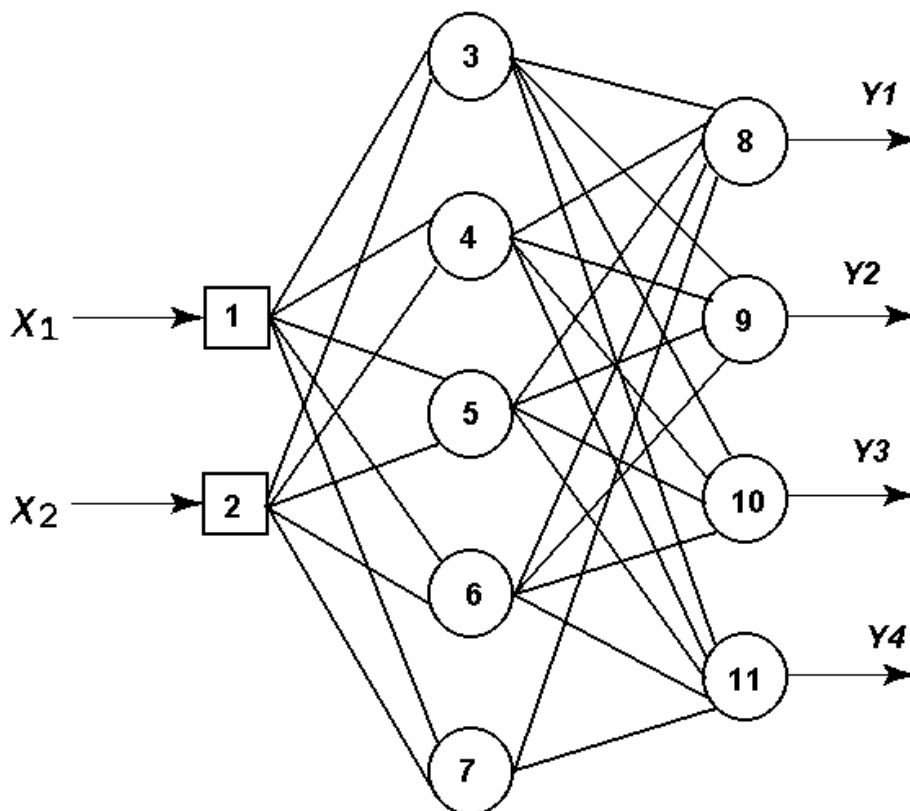


Рис. 5. Структура нейронной сети 2-1-5-4

После обучения НС сможет выбирать одну из четырех угроз в зависимости от значений признаков атаки, поступающих на входы x_1 и x_2 нейронной сети.

Причем значения признаков атаки могут находиться в диапазоне $[-1, 1]$, что говорит о том, что НС будет формировать классификационные заключения даже в случае не вполне достоверных данных (с истинностью, отличной от 1 или 0).

2. Возможности программы Fann

Программа **Fann** состоит из двух модулей: мультипоточкового ядра **fannKernel** и интерфейсного модуля **fannExplorer** (рис. 6).

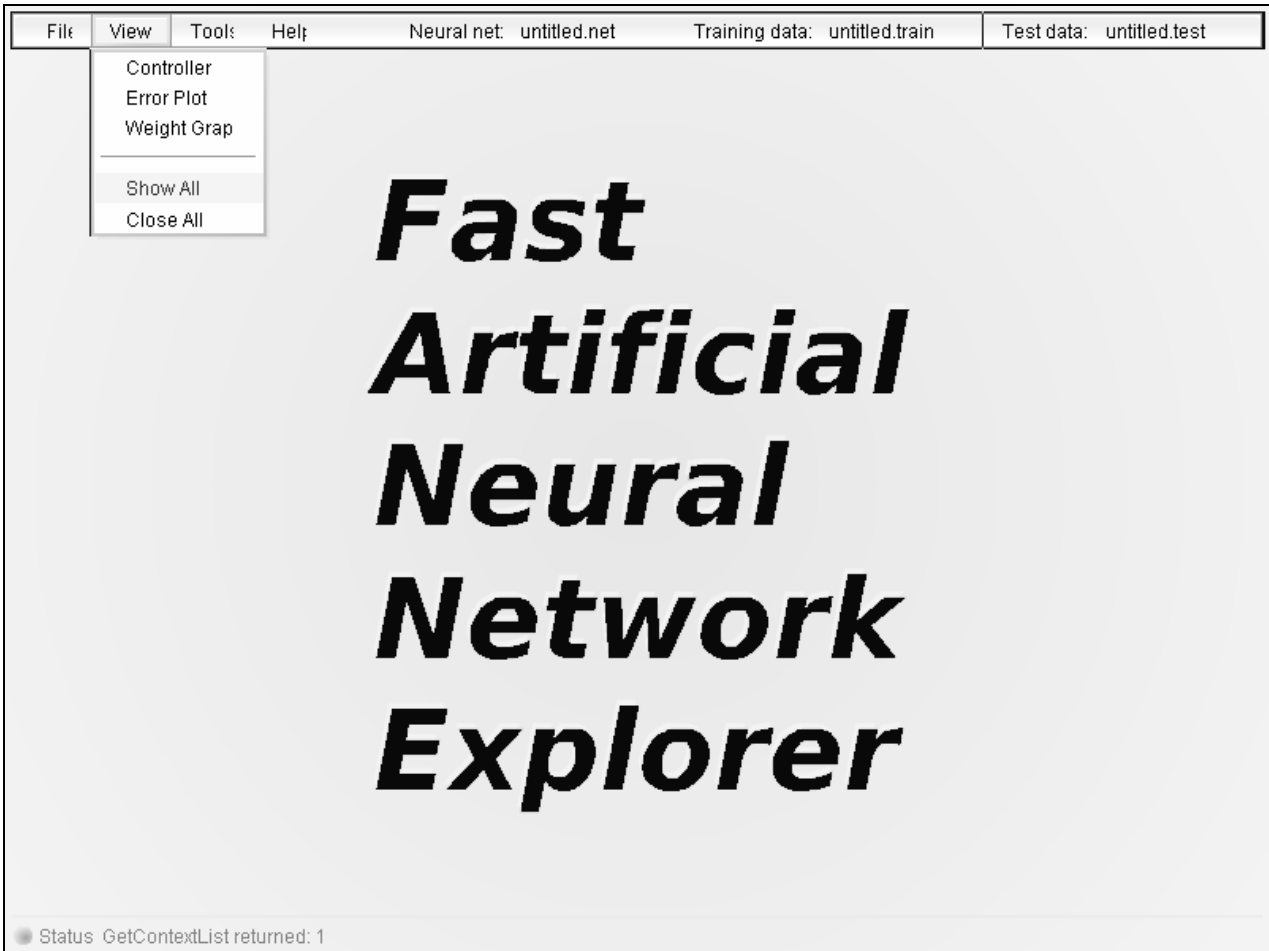


Рис. 6. Главное окно программы

2.1. Активация среды моделирования Fann

1. В каталоге **fannKernel** активируйте мультипоточковое ядро **fannKernel.exe**.
2. В каталоге **fannExplorer** активируйте интерфейсный модуль **fannExplorer.exe**.

Так как программный модуль ядра **fannKernel.exe** является мультипоточковым, одновременно можно запустить несколько интерфейсных модулей **fannExplorer.exe**. Этим пользуются, если необходимо исследовать несколько нейронных сетей одновременно.

2.2. Пользовательский интерфейс программы FannExplorer

Описание главного меню

Ниже приведены описания пунктов меню, которые вам могут понадобиться при работе с интерфейсным модулем **fannExplorer**. Поскольку программа не русифицирована, то основное назначение данного раздела описания - перевод на русский язык наименований используемых в лабораторном практикуме элементов интерфейсного модуля.

Опция File

- **New Neural Network** – позволяет создавать топологию **Нейронной сети**;
- **Save Neural Network** – осуществляет **Сохранение нейронной сети**;
- **Load Neural Network** – позволяет **Загрузить нейронную сеть** для дальнейших исследований.

В последнем случае открывается окно (рис. 7), которое содержит список файлов с описанием нейронных сетей, размещенных в каталоге **net**.

Остальные строки опции **File** позволяют выполнять операции загрузки (**Load**) и сохранения (**Save**), но только уже с данными для обучения (**Training Data**) и данными для тестирования (**Test Data**).

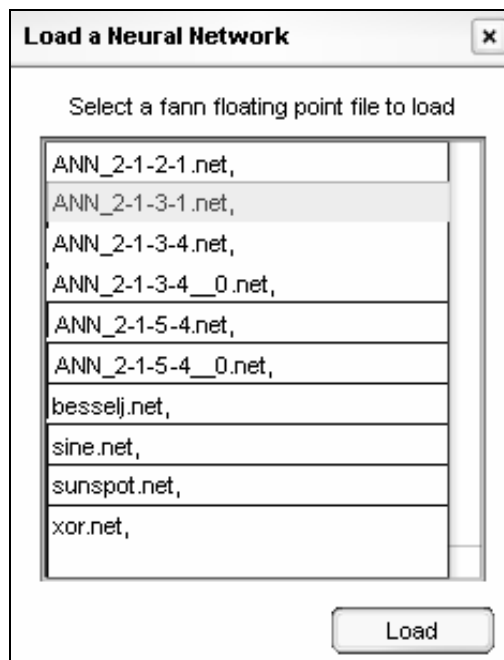


Рис. 7. Окно загрузки НС

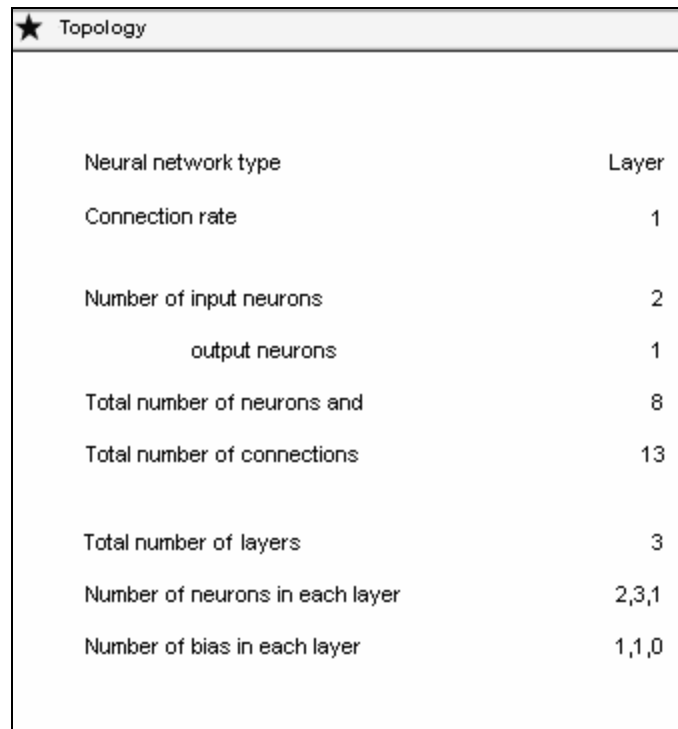
Следует заметить, что данные для обучения и тестирования НС следует создавать заблаговременно с помощью текстового редактора. В интерфейсном модуле **FannExplorer**, возможно только редактирование этих данных.

Опция View

Данная опция позволяет открывать (закрывать) три основных окна, размещенных в главном окне интерфейсного модуля **FannExplorer**, а именно: **Controller**, **Weight Graph** и **Error plot**.

Controller

Neural Network Controller - окно, предназначенное для настройки и управления процессом моделирования нейронной сети. Структура НС формируется заранее (см. **Создание нейронной сети**), а на вкладке **Topology** (рис. 8) можно ознакомиться с основными характеристиками ее топологии.



Neural network type	Layer
Connection rate	1
Number of input neurons	2
output neurons	1
Total number of neurons and	8
Total number of connections	13
Total number of layers	3
Number of neurons in each layer	2,3,1
Number of bias in each layer	1,1,0

Рис. 8. Вкладка топологии НС

На вкладке **Algorithm** (рис. 9) отображается информация по параметрам обучения НС.

На вкладке **Algorithm** указывается:

- ◆ **алгоритм обучения (Training algorithm):**
 - метод последовательного обучения (**Incremental**),
 - эластичный метод (**Resilient**),
 - быстрый (**Quick**),
 - периодический метод (**Batch**);
- ◆ **функции ошибки (Error function):**
 - линейная (**Linear**),

- тангенциальная (**Tanh**);
- ◆ **функции активации скрытых слоев (Hidden layer activation) и выходных нейронов (Output layer activation):**
 - симметричная/несимметричная (**Symmetric**),
 - линейная (**Linear**),
 - сигмоидальная (**Sigmoid**),
 - ступенчатая (**Stepwise**),
 - пороговая (**Threshold**).
- ◆ параметр **скорости (норма) обучения (Learning rate)** и пр.

Algorithm	
Training algorithm	<input checked="" type="radio"/> Resilient <input type="radio"/> Quick <input type="radio"/> Incremental <input type="radio"/> Batch
Error function	<input type="radio"/> Linear <input checked="" type="radio"/> Tanh
Hidden layer activation	<input type="radio"/> Linear <input type="radio"/> Threshold <input checked="" type="checkbox"/> Symmetric <input checked="" type="radio"/> Sigmoid <input type="radio"/> Stepwise
Output layer activation	<input type="radio"/> Linear <input type="radio"/> Threshold <input checked="" type="checkbox"/> Symmetric <input checked="" type="radio"/> Sigmoid <input type="radio"/> Stepwise
Learning rate	<input type="text" value="0.15000001"/>
Hidden layer steepness	<input type="text" value="0.5"/>
Output layer steepness	<input type="text" value="0.5"/>
Resilient propagation advanced parameters	
Increase factor	<input type="text" value="1.2"/>
Decrease factor	<input type="text" value="0.5"/>
Minimum delta	<input type="text" value="0"/>
Maximum delta	<input type="text" value="50"/>

Рис. 9. Вкладка настройки параметров алгоритма обучения НС

Данные, заданные по умолчанию, подходят для проведения лабораторных работ. Наиболее важным (для моделирования процесса обучения НС) является **норма обучения (Learning rate)**.

На вкладке **Training** управления процессом обучения НС (рис. 10) задаются параметры адаптации весов связей нейронной сети, т.е. количество эпох обучения (**Maximum number of training epochs**), значение среднего отклонения ошибки (**Square mean error**).

На этой же вкладке устанавливаются начальные значения весов связей нейронной сети (**Initialize**). При инициализации величины весов связей используется алгоритм Widrow и Nguyen, зависящий от входных данных.

Назначать значения весов связей НС можно и случайным образом (**Randomize**), предварительно задав диапазон значений весов (**Minimum random weight value** и **Maximum random weight value**).

При нажатии на кнопку **Train** НС начнет обучаться. Обучение продолжается до тех пор, пока не будет пройдено количество эпох обучения, заданное в окне **Maximum number of training epochs** или среднеквадратичное отклонение ошибки не достигнет величины, указанной в окне **Square mean error**.

The screenshot shows a window titled "Training" with the following controls:

- Minimum random weight value:** -0.1
- Maximum random weight value:** 0.1
- Buttons:** Shuffle training data, Randomize
- Initialize weights from training data using Widrow & Nguyen's algorithm:** Initialize
- Maximum number of training epochs:** 200
- Stop when the mean square error falls below:** 0.001
- Epochs between mean square error reports:** 10
- Buttons:** Edit training data, Train
- Epochs between display updates:** 20
- Update output plot while animating:** Animate

Рис. 10. Вкладка управления процессом обучения НС

На вкладке **Testing** (рис. 11) можно проверить, насколько хорошо обучилась исследуемая нейронная сеть.

Имеется возможность редактирования входных данных для теста (эталонных значений). Результаты теста выражаются в значении среднеквадратичного отклонения полученных результатов на выходе с заданными эталонными значениями.

Для каждого выхода можно графически отобразить эталонное значение и значение, получаемое на выходе, что значительно улучшает наглядность представления результатов. Для этого на этой вкладке укажите, для какого выходного нейрона следует построить график (**Select output neuron for plot**), и нажмите кнопку **Execute**. На графике красным цветом отображаются полученные значения, формируемые выходным нейроном, а зеленым – эталонные значения.

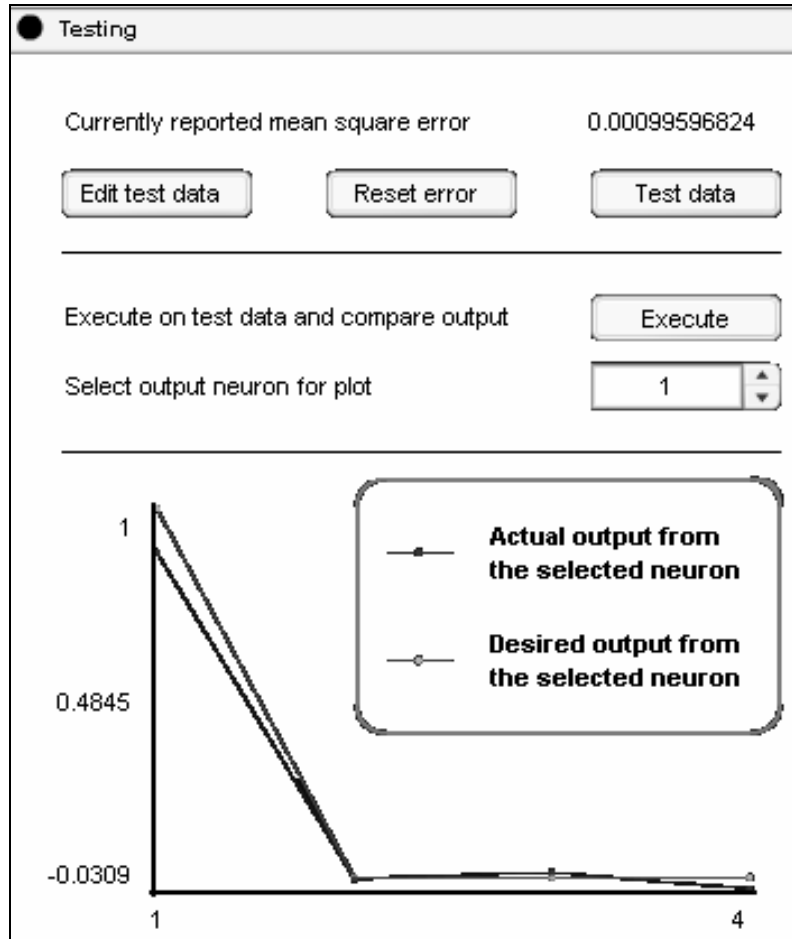


Рис. 11. Вкладка тестирования обученной НС

Внимание! Знаком вопроса отмечены всплывающие подсказки. Обычно они содержат пояснения к графикам или схемам.

Weight Graph

При выборе данного пункта опции **View** открывается окно, в котором отображается топология нейронной сети (рис. 12). Следует отметить, что нейроны на графе изображаются желтым цветом, смещения – зеленым; положительные связи - синим цветом, а отрицательные - красным. Значения весов связей можно редактировать вручную, если они представлены в табличном виде. Переключение режимов осуществляется с помощью кнопки, расположенной в левом нижнем углу окна.

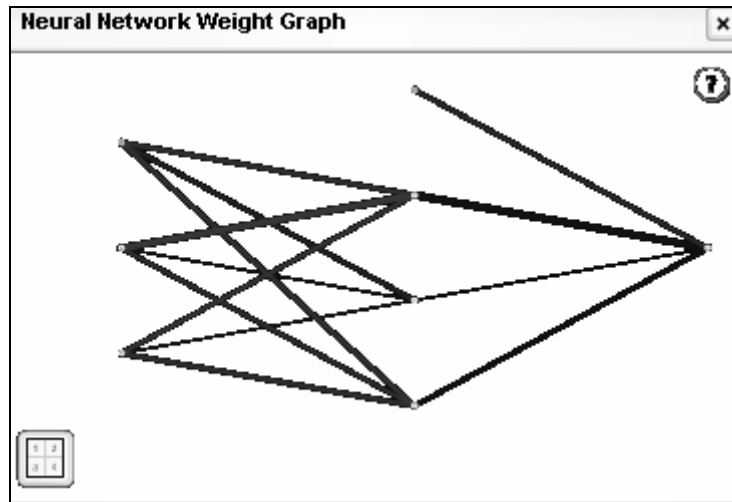


Рис. 12. Окно топологии исследуемой НС

Error plot

При выборе данного пункта опции **View** открывается окно (рис. 13), в котором отображается процесс построения графика среднеквадратичной ошибки.

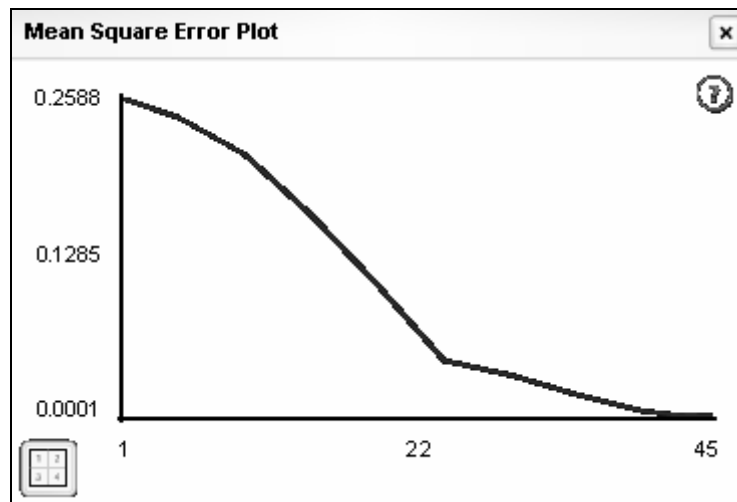


Рис. 13. Окно динамики ошибки при обучении исследуемой НС

Причем также как и в предыдущем случае значения ошибки могут быть представлены как в виде графика, так в виде таблицы.

Show all и Close All

Очевидно, что выбор пункта **Show all** опции **View** открывает все вышеописанные окна, а пункта **Close all** – закрывает все открытые окна главного окна интерфейсной среды.

Опция Tools

Предназначена для установления связи с удаленным ядром **fannKernel.exe**. Однако при проведении лабораторного практикума данная опция главного меню не используется.

Дополнительные сведения можно почерпнуть, обратившись к опции **Help** главного меню или к серверу **Macromedia Inc.**

3. Описание среды моделирования

Рассмотрим более детально процесс моделирования нейронных сетей с использованием программы *FANN* (*Fast Artificial Neural Networks*) от *Macromedia Inc.*

В *FANN* нейронная сеть, изображённая на рис. 1, представляется в следующем виде (рис. 14.), т. е. узлы, формирующие сигналы смещения для нейронов (на рис. 14 заштрихованы), отнесены к предыдущему слою НС. На рис. 14 сохранена нумерация весов и порогов (смещений), используемая на рис. 1, в то же время узлы НС пронумерованы по слоям, чтобы можно было анализировать результаты обучения.

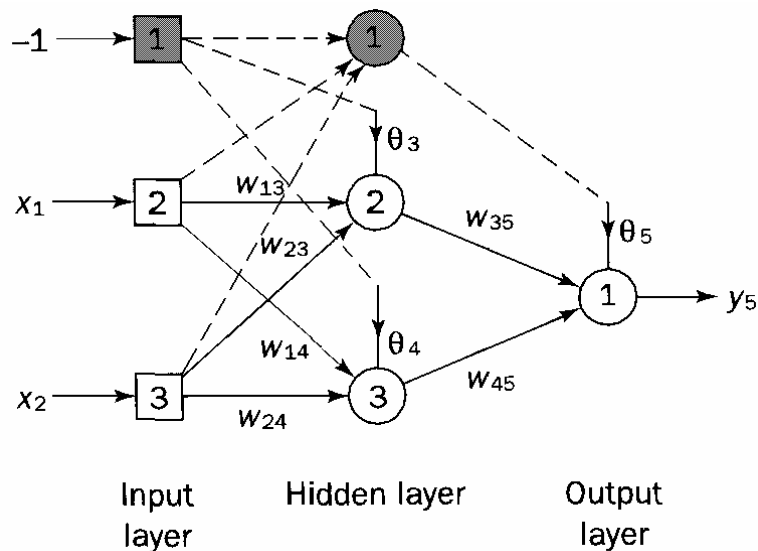


Рис. 14. Представление НС в среде FANN

3.1. Работа с программой FANN

Программа *FANN* (далее по тексту - программа) активируется файлом **fannKernel.exe**, размещенным в каталоге **fannKernel**, а среда моделирования – запуском файла **fannExplorer.exe**, размещенным в каталоге **fannExplorer**, в результате чего открывается главное окно, представленное на рис. 6.

Внимание! Следует соблюдать указанный порядок активации файлов: вначале запускается ядро программы - **fannKernel.exe**, а затем при активном ядре запускается интерактивная среда моделирования **fannExplorer.exe**.

Рекомендуем знакомство с программой начать с опции **View** (рис. 15), выбрав в качестве пункта меню **Show all**.

В результате открываются три окна: **Neural Network Controller**, **Neural Network Weight Graph**, **Mean Square Error Plot**.

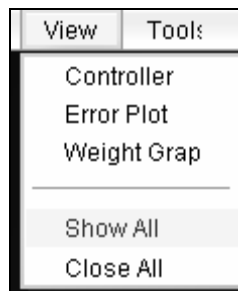


Рис. 15. Меню опции View

В первом окне осуществляется диалог с программой в процессе настройки и проведения моделирования НС, а два оставшихся служат для вывода результатов моделирования в графической и цифровой формах (рис. 16).

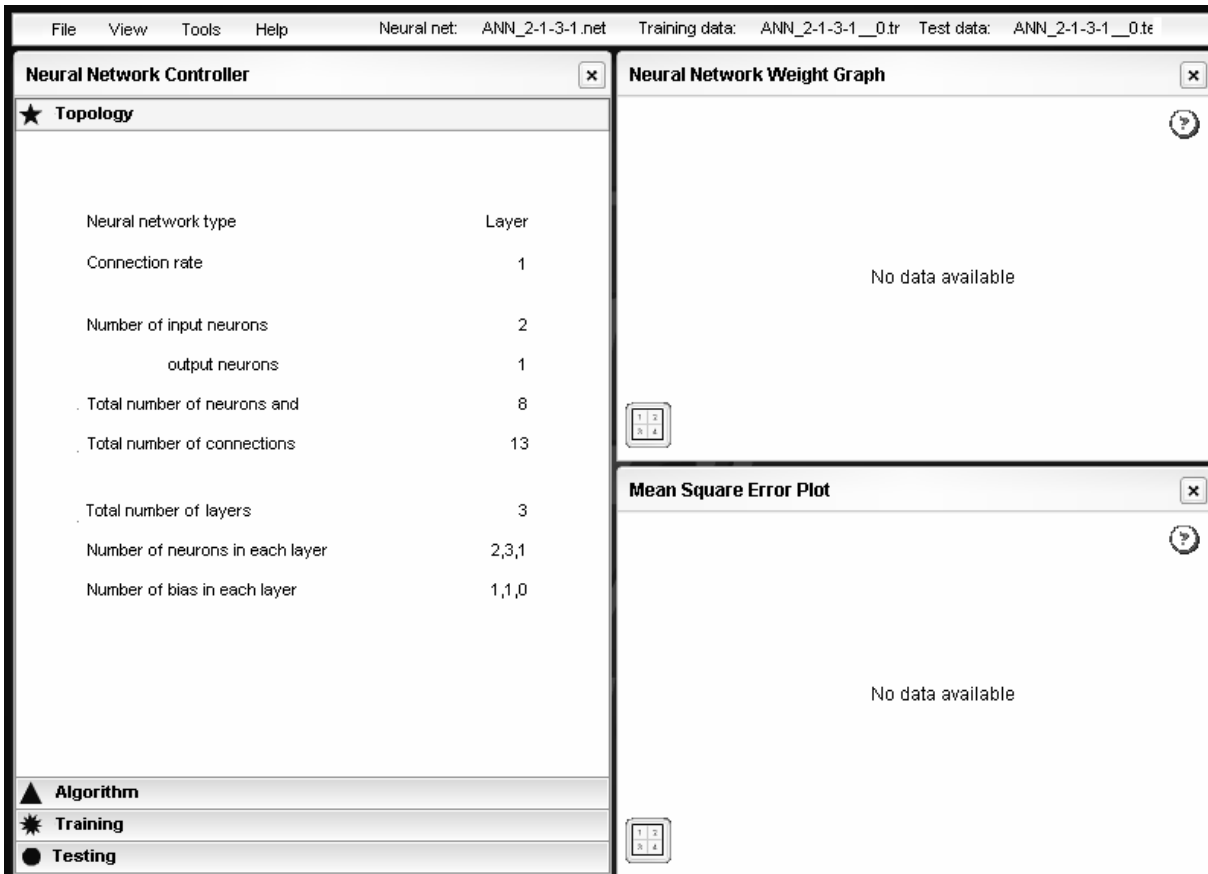


Рис. 16. Окна для моделирования НС

Для описания модели создаваемой нейронной сети необходимо в главном меню выбрать опцию **File** и затем в открывшемся меню активировать строку **New Neural Network** (рис. 17).

В открывшемся окне **Create New Neural Network** (рис. 18) следует задать конфигурацию исследуемой нейронной сети.

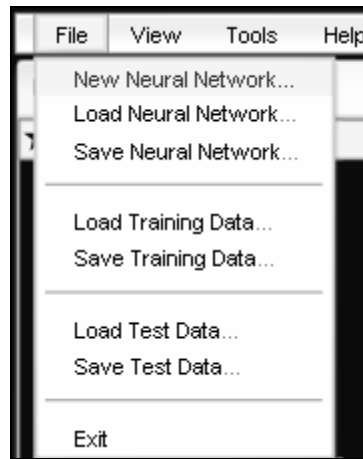


Рис. 17. Меню опции File

Для задания конфигурация в окнах списков задаются целочисленные значения, а именно: число входов (**Input Neuron**), число выходных нейронов (**Output Neuron**), количество скрытых слоев (**Hidden layers**), число нейронов в 1-м скрытом слое (**Neurons in 1st hidden layer**). Например, на рис. 18 задана конфигурация НС 2-1-3-1.

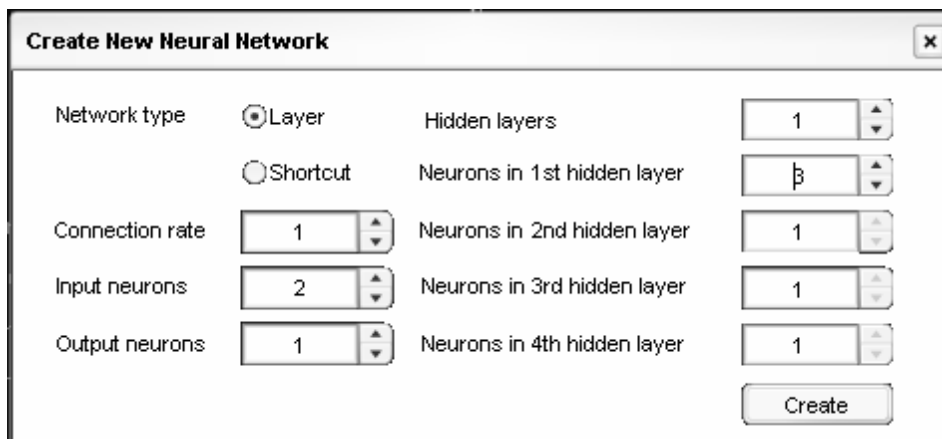


Рис. 18. Окно «Создание новой нейронной сети»

Активация кнопки **Create** приводит к выводу в окне **Neural Network Weight Graph** структуры исследуемой нейронной сети (рис. 12).

Перемещение указателя мыши на знак вопроса, расположенный в верхнем правом углу окна, открывает окно комментариев (рис. 19).

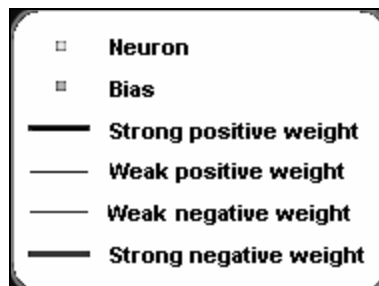


Рис. 19. Окно комментариев

Для сохранения конфигурации созданной НС необходимо в меню опции **File** (рис. 17) активировать строку **Save Neural Network** и в открывшемся окне

набрать имя файла, который будет сохранен с расширением *.net, например, ANN_2-1-3-1.net (рис. 20).

Внимание! Расширение *.net формируется автоматически (набирать его не следует).

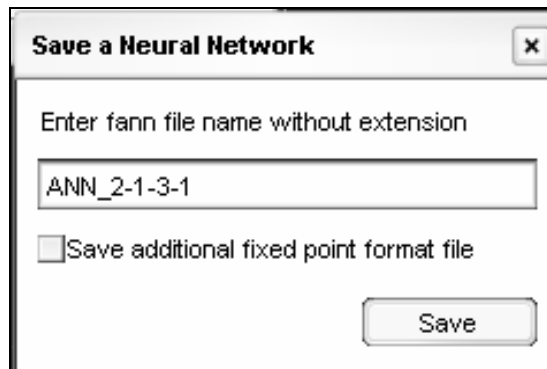


Рис. 20. Окно сохранения конфигурации

Моделирование процесса обучения НС может быть произведено с ранее созданной и сохраненной в каталоге **net** нейронной сетью. Загрузка производится через опцию **File** главного меню (рис. 17) и подменю **Load Neural Network** (рис. 21).

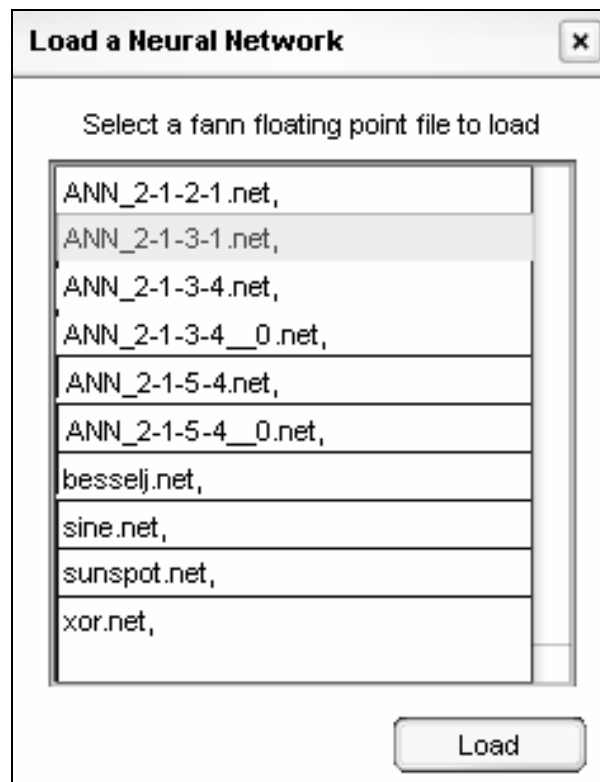


Рис. 21. Окно загрузки НС

Для дальнейшей работы программы необходимо загрузить файлы обучающей выборки и тестовых последовательностей, которые необходимо создать заранее с помощью программы Блокнот (рис. 22). Приведенная в примере обучающая выборка соответствует переключательной функции «ИЛИ-НЕ».

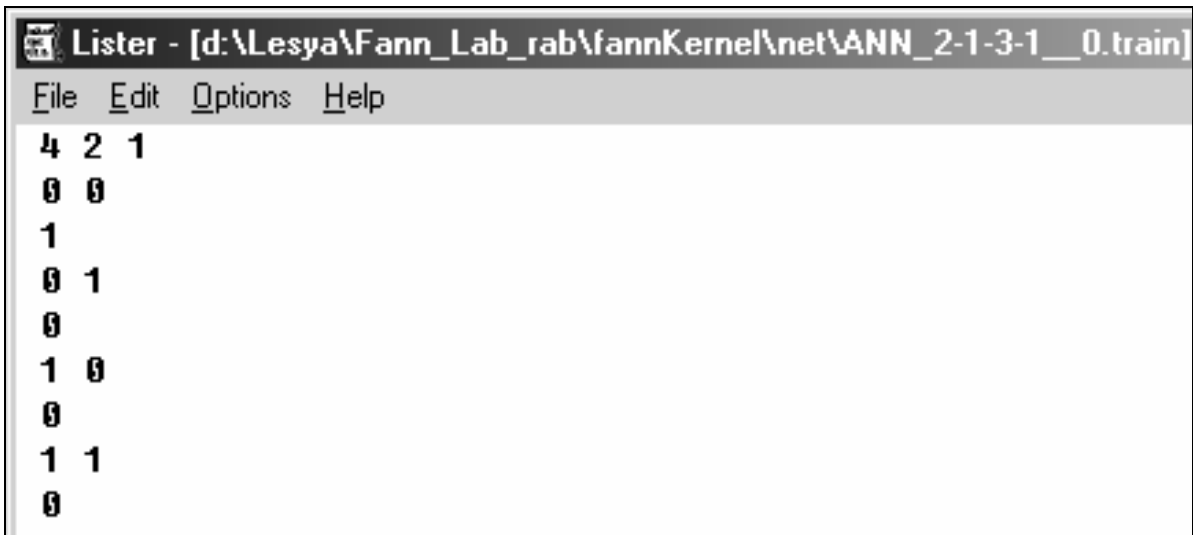


Рис. 22. Окно Блокнота

Например, файл **ANN_2-1-3-1.train** описывает структуру, аналогичную представленной на рис. 12, построчно:

1-я строка:	4 2 1	4 примера обучающей выборки, состоящей из 2 входных и 1 выходного значения
2-я строка:	0 0	2 входных значения
3-я строка:	1	1 выходное значения
4-я строка:	0 1	2 входных значения
5-я строка:	0	1 выходное значения
6-я строка:	1 0	2 входных значения
7-я строка:	0	1 выходное значения
8-я строка:	1 1	2 входных значения
9-я строка:	0	1 выходное значения

В фрагменте все цифровые значения разделены пробелами.

Внимание! Для правильной работы программы необходимо размещать файлы с расширениями ***.train** и ***.test** в каталоге **net**, который в свою очередь должен находиться в каталоге **fannKernel**.

Для обучения нейронной сети следует загрузить файлы с расширениями ***.train** и ***.test** с помощью опции **File** (рис. 17) из главного меню, для чего необходимо в открывшемся подменю выбрать строку **Load Training Data...** и соответствующий файл (на рис. 23 выбран файл **ANN_2-1-3-1.train**).

Аналогично загружаются тестовые последовательности значений (например, файл **ANN_2-1-3-1.test**), используя строку **Load Test Data....**

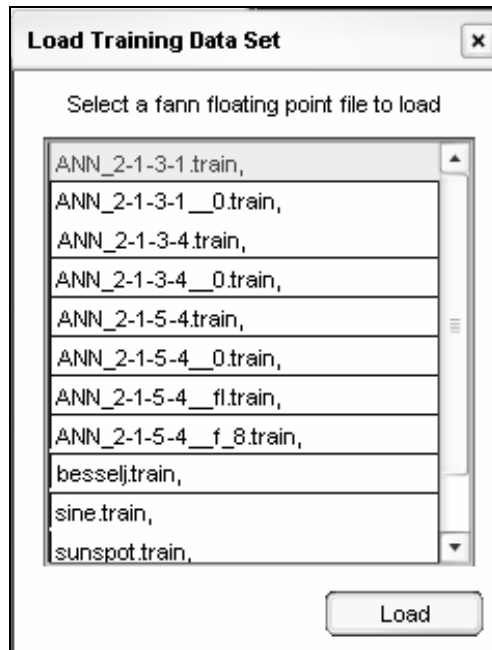


Рис. 23. Окно загрузки обучающей выборки

Дальнейшие действия по обучению НС связаны с окном **Neural Network Controller** (рис. 24), которое содержит четыре вкладки: **Topology**, **Algorithm**, **Training**, **Testing**.

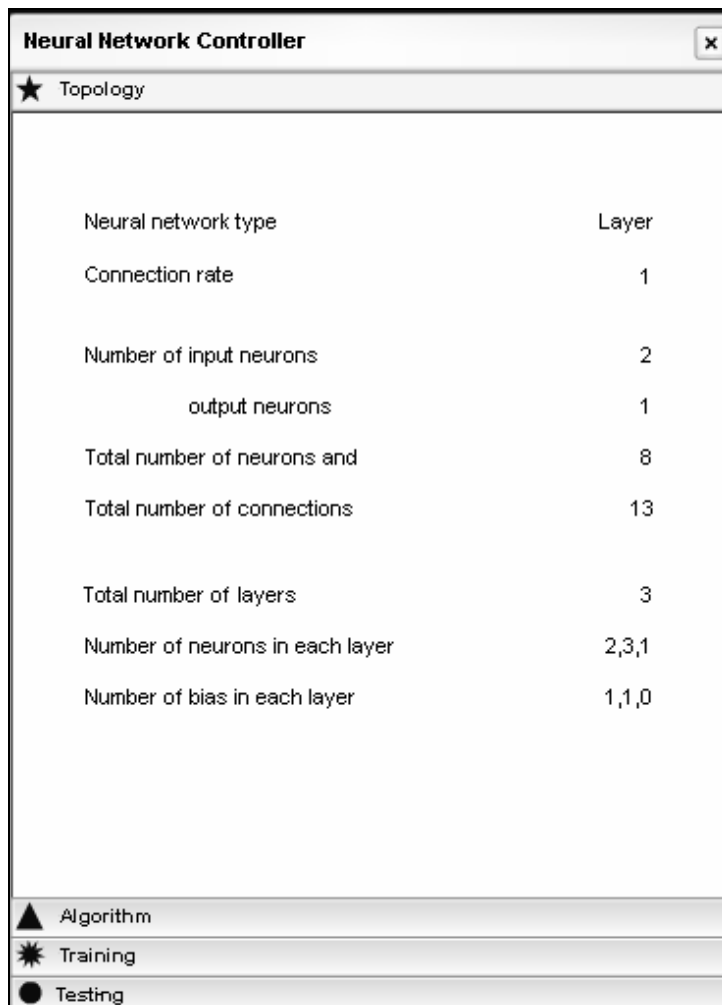


Рис. 24. Вкладка топологии НС

Вкладка **Topology** содержит справочную информацию о вновь созданной или загруженной конфигурации нейронной сети. В основном (рис. 24) на вкладке повторяются численные значения, характеризующие топологию нейронной сети, которые были введены в процессе задания конфигурации НС.

Вкладка **Algorithm** позволяет выбрать один из алгоритмов и параметры обучения нейронной сети. В частности на рис. 25 выбран алгоритм обучения (**Training Algorithm**) **Resilient**, функция ошибки (**Error function**) - **Tanh**, в качестве сжимающих (активационных) функций в скрытом и выходных слоях НС – сигмоидальная (**Sigmoid**) симметричная (**Symmetric**) функция. Также задаются значения скорости (**Learning Rate**), шага (**Steepness**) обучения и дополнительные параметры используемого алгоритма обучения НС (**Resilient propagation advanced parameters**).

После настройки параметров алгоритма обучения можно перейти к моделированию самого процесса обучения нейронной сети с отображением динамики изменения как весов связей (окно **Neural Network Weight Graph**), так и ошибки обучения НС (окно **Mean Square Error Plot**).

Neural Network Controller

★ Topology

▲ Algorithm

Training algorithm: Resilient Quick
 Incremental Batch

Error function: Linear Tanh

Hidden layer activation: Linear Threshold
 Symmetric Sigmoid Stepwise

Output layer activation: Linear Threshold
 Symmetric Sigmoid Stepwise

Learning rate:

Hidden layer steepness:

Output layer steepness:

Resilient propagation advanced parameters

Increase factor: Decrease factor:

Minimum delta: Maximum delta:

☀ Training

● Testing

Рис. 25. Вкладка выбора и настройки алгоритма обучения НС

Процесс моделирования нейронной сети запускается с вкладки обучения (рис. 26) **Training**. Вначале следует инициализировать нейронную сеть. Инициализация нейронной сети производится:

- путем генерации последовательности случайных чисел (**Randomize**) в конкретном диапазоне значений (**Minimum random weight value** и **Maximum random weight value**),
- с использованием конкретного алгоритма задания последовательности значений в качестве весов связей нейронной сети (**Initialize weights from training data**).

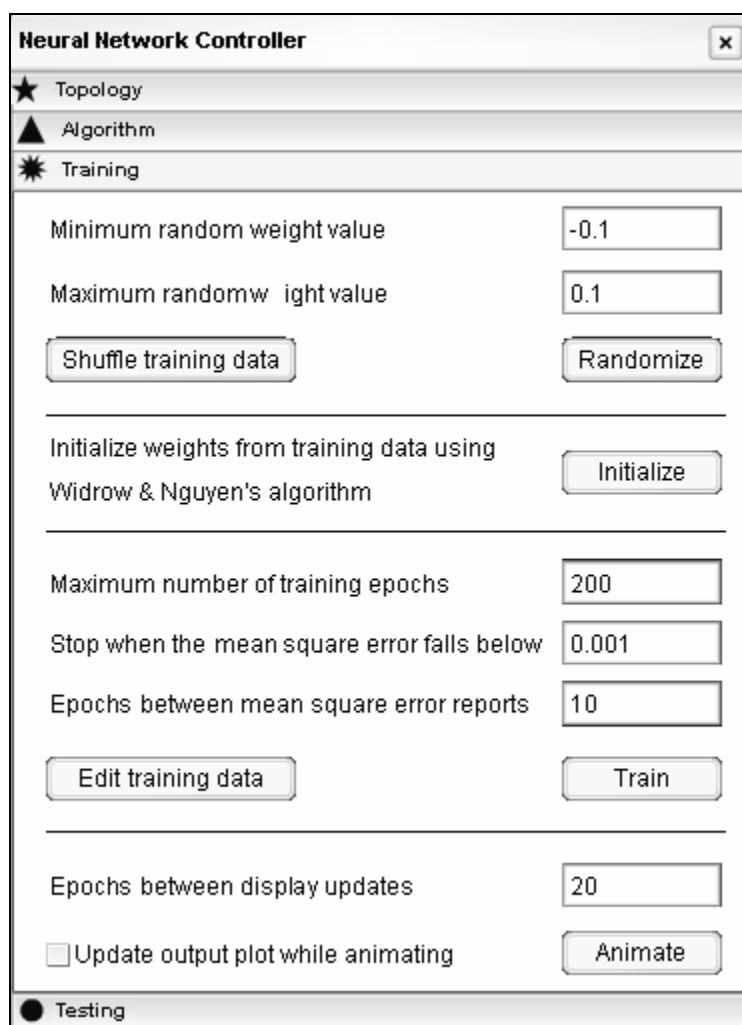


Рис. 26. Вкладка обучения НС

На рис. 26 присвоение весам связей *случайных значений* выполняется после активации кнопки **Randomize**, причем диапазон значений весов связей ограничен интервалом $[-1, 1]$. При нажатии кнопки **Initialize** веса связей формируются по алгоритму **Windrow & Nguyen's Algorithm**.

Также следует определить максимальное число эпох обучения (**Maximum number of training epoch**). Под эпохой понимается цикл обучающего алгоритма, в течение которого нейронной сети предъявляют все примеры из обучаю-

щей выборки. В рассматриваемом примере нейронной сети предъявляются 4 варианта входных и выходных значений, соответствующие логической функции NOR из табл. 1), а именно – строки со 2-й по 9-ю (рис. 22).

Следует также задать точность обучения (**Stop when the mean square falls below**) и шаг формирования отчета о динамике точности обучения НС (**Epochs between mean square reports**).

Активировать процесс обучения НС можно нажав кнопку **Train** или **Animate**. В последнем случае можно наблюдать динамику изменения ошибки обучения в зависимости от числа эпох обучения. Шаг отражения динамики изменения ошибки задается в окне **Epoch between means square error reports**. Следует обратить внимание, что возможна коррекция таблицы обучающей выборки путем активации кнопки **Edit training data**, вызывающей появление окна редактирования обучающих данных (рис. 27). В окне возможно как удаление (**Delete**), так и добавление (**Add**) отдельных строк таблицы. При завершении редактирования нажатием кнопки **Apply** происходит возврат на вкладку **Training**.

Input 1	Input 2	Output 1
-1	1	1
-1	1	-1
1	-1	-1
1	1	-1

Buttons: Add, Delete, Apply

Рис. 27. Окно редактирования обучающих данных

Кроме того, нажатием кнопки **Shuffle training data** (Перетасовка) можно изменять порядок размещения примеров обучающей выборки в таблице, представленной на рис. 27.

Результат моделирования процесса обучения нейронной сети представлен на рис. 28.

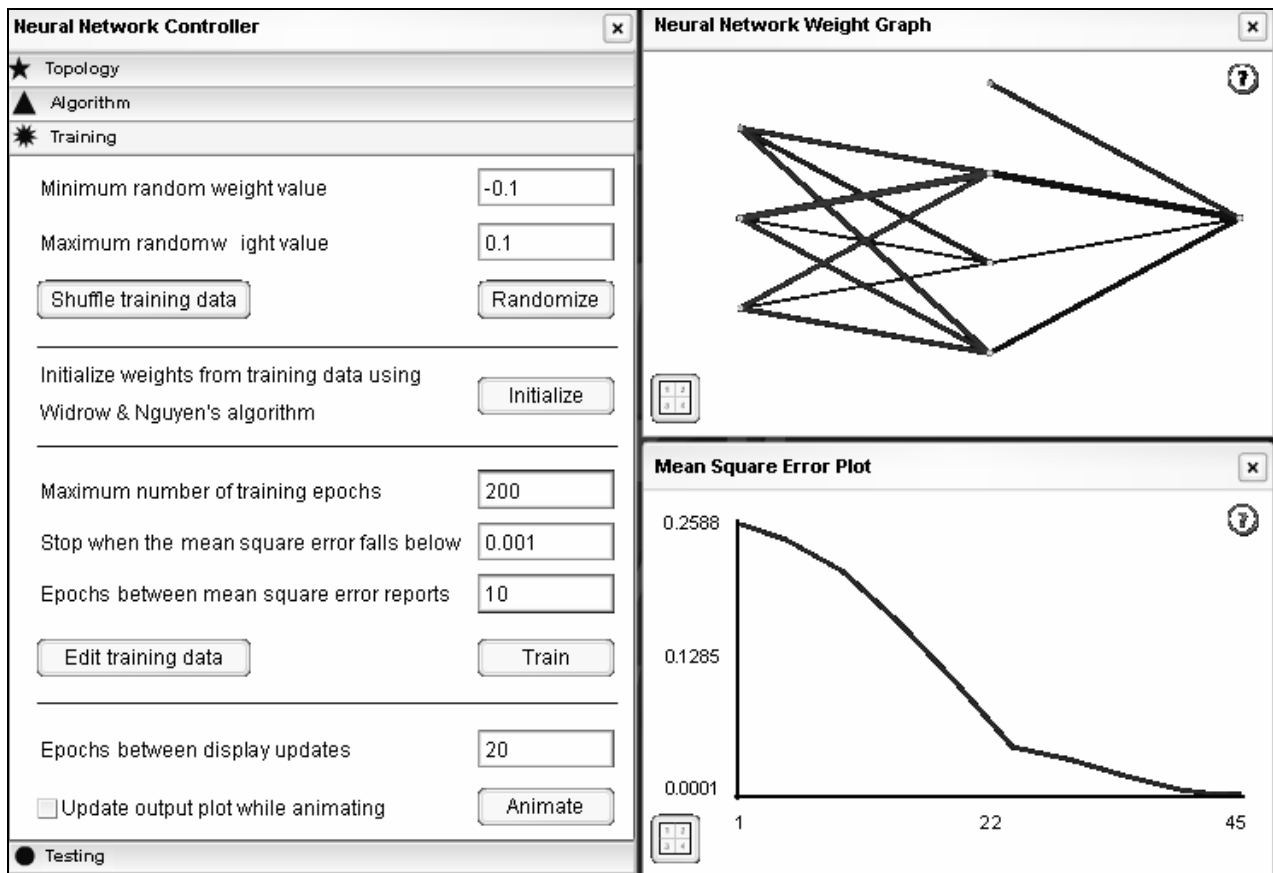


Рис. 28. Результирующее окно обучения НС

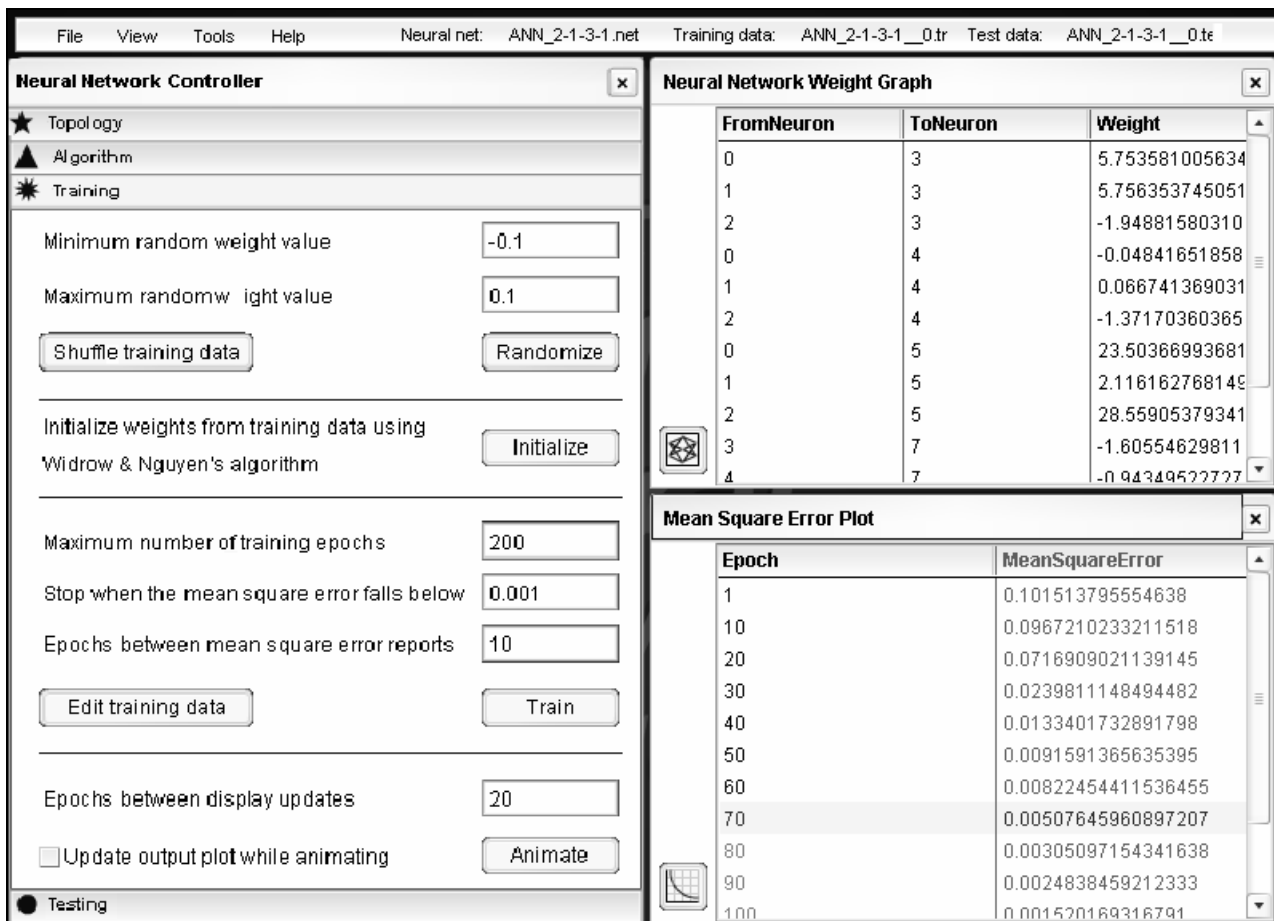


Рис. 29. Результирующее окно с таблицами весов и динамики ошибки обучения НС

После завершения процесса моделирования нажатию кнопки в левом нижнем углу окон **Neural Network Weight Graph**, **Mean Square Error Plot** можно просмотреть таблицы значений весов и динамику ошибки в зависимости от числа эпох обучения (рис. 29).

Повторное нажатие тех же кнопок возвращает исходный вид результирующему окну (рис. 28).

Последний этап моделирования процесса обучения нейронной сети заключается в верификации результатов обучения НС, когда сравниваются реальные и желаемые значения на выходах обученной нейронной сети. Для перехода в режим тестирования активируется вкладка **Testing**, представленная на рис. 30.

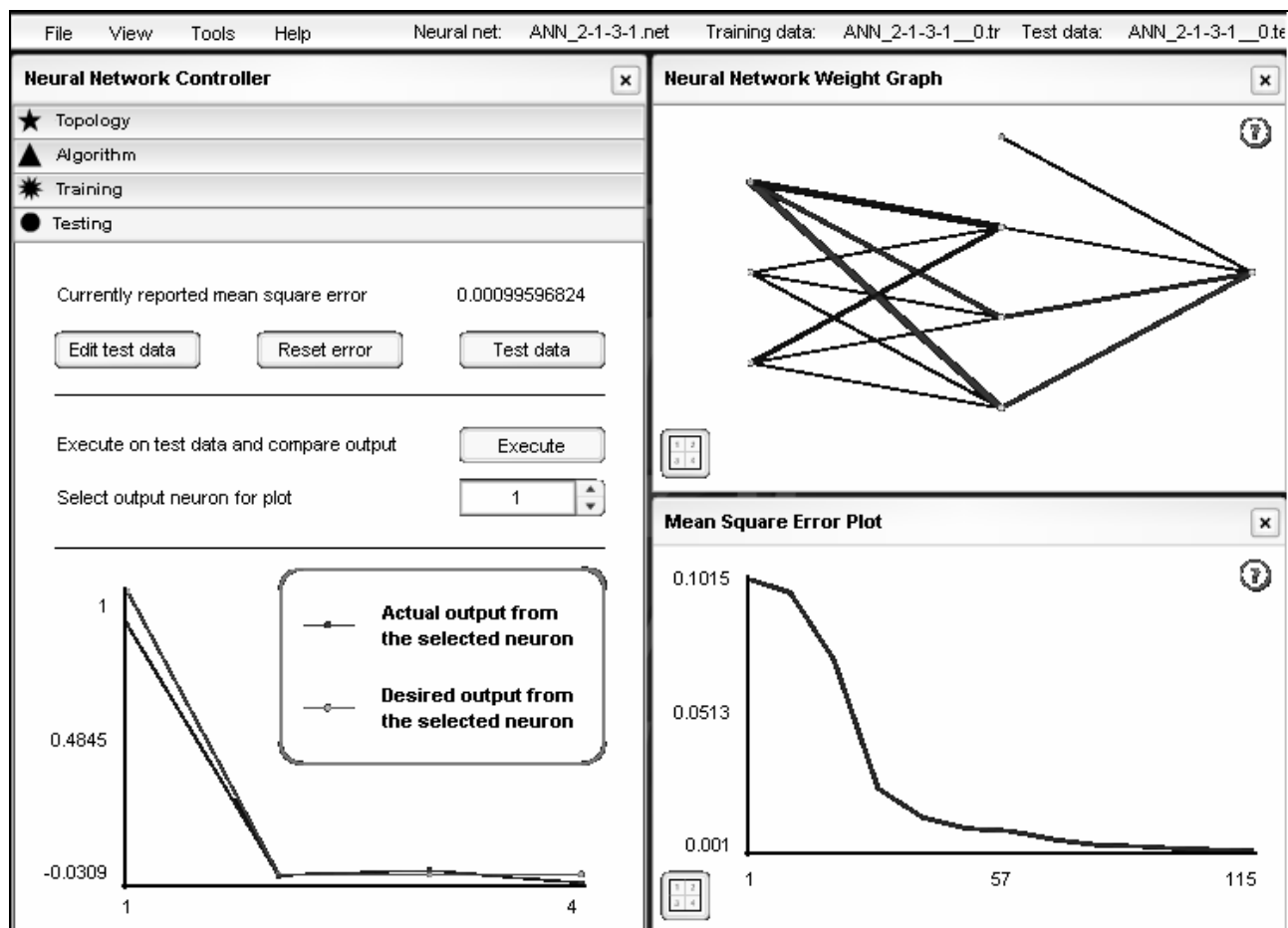


Рис. 30. Окно верификации результатов ошибки обучения НС

Перед началом тестирования можно (как и в случае коррекции данных обучающей выборки) можно провести редактирования тестовых наборов. Для этого достаточно активировать кнопку **Edit test data** и перейти к окну, аналогичному изображенному на рис. 27.

Нажатием кнопки **Reset error** производится сброс значения ошибки обучения, а активация кнопки **Test data** приводит к выводу итогового значения ошибки обучения НС.

Кнопка **Execute** вызывает обновление содержимого окна, в графической форме отражающего степень близости желаемого и реального значений выходов НС, а список **Select output neuron for plot** позволяет просмотреть результаты обучения выбранного выхода нейронной сети.

Сохранить результаты моделирования можно в файле, для чего достаточно в главном меню выбрать опцию **File**, а в появившемся подменю строку **Save Neural Network** (рис. 17).

Затем набрать имя файла с результатами обучения, например, (рис. 20).

Содержимое файла **ANN_2-1-3-1.net** с результатами обучения можно просмотреть с помощью программы Блокнот (рис. 31), где приведены значения весов связей обученной нейронной сети.

```

FANN_FLO_1.1
3 0.150000 1.000000 0 5 5 5.000000000000000000000000e-001 5.000000000000000000
3 4 2
0 0 0
3 3 3 0
4 0
(0 -1.09143726450931380000e+000) (1 -1.06422355291217530000e+000) (2 -5.8
-1.01403603986118200000e+001) (1 -1.12242801958735950000e+000) (2 -2.8941
1.24363315924766170000e+000) (1 1.29308161348635790000e+000) (2 5.8559814
5.16417373895344150000e+000) (4 9.05605212731239510000e-001) (5 -1.977811
-7.87367638416803680000e-001)

```

Рис. 31. Содержимое файла ANN_2-1-3-1.net

4. Практическая часть

В практической части методических указаний содержатся рекомендации и задания, необходимые для формирования, обучения и верификации результатов обучения нейронных сетей, решающих задачи классификации угроз.

В первой части лабораторного практикума проводится исследование фрагмента НС, представленного на рис. 1, а именно: нейронной сети с конфигурацией 2-1-2-1, осуществляющей выбор одной из возможных угроз. Необходимо исследовать влияние на время обучения нейронной сети (количество эпох обучения, необходимых для достижения заданной точности обучения НС) следующих параметров:

- параметра скорости обучения (Задание на лаб. раб № 1),
- числа формальных нейронов в скрытом слое (Задание на лаб. раб № 2),
- числа скрытых слоев (Задание на лаб. раб № 3).

Следует обратить внимание, что при одной и той же структуре нейронной сети (рис. 1) в процессе обучения весов межнейронных связей формируется такая совокупность весов (см. файл весов связей после процесса обучения НС), которая обеспечивает реализацию логической функции, заданной набором данных обучающей выборки (файл *.train).

Во второй части лабораторного практикума проводится исследование нейронной сети, аналогичной представленной на рис. 5, а именно: нейронной сети с конфигурацией 2-1-3-4, 2-1-4-4 или 2-1-5-4, решающей задачу классификации угроз.

Топология нейронной сети для решения задачи классификации угроз, например, для нейронной сети с конфигурацией 2-1-5-4 будет иметь вид (рис. 32).

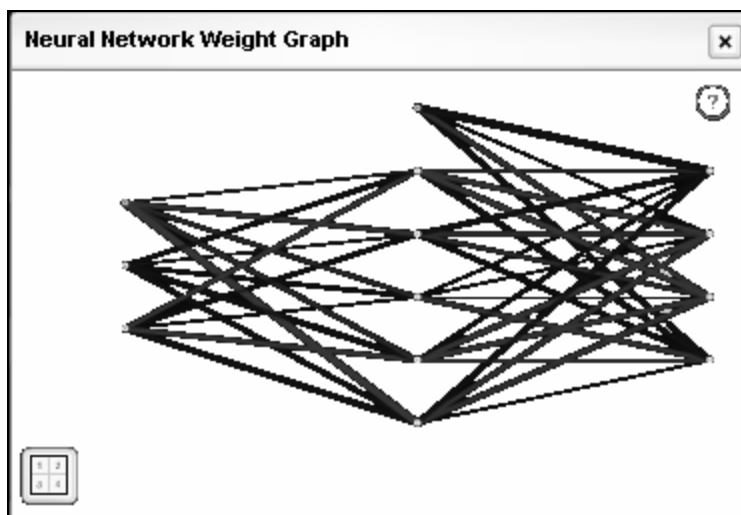


Рис. 32. Топология НС с конфигурацией 2-1-5-4

Набор тестовых данных можно сформировать в виде, аналогичном представленному на рис. 33.

```

Lister - [d:\Lesya\Fann_Lab_rab\fannKernel\net\ANN_2-1-5-4_0.test]
File Edit Options Help
4 2 4
0 0
1 0 0 0
0 1
0 1 0 0
1 0
0 0 1 0
1 1
0 0 0 1

```

Рис. 33. Набор тестовых данных для НС с конфигурацией 2-1-5-4

Набор данных обучающей выборки может содержать значения признаков атаки, выраженные как целочисленными значениями, так и в вещественными числами (рис. 34).

Следует исследовать влияние на время обучения нейронной сети и точность функционирования НС использования только целочисленных данных в обучающей выборке и на аналогичный параметр при использовании вещественных чисел в множестве значений обучающей выборки.

```

Lister - [d:\Lesya\Fann_Lab_rab\fannKernel\net\ANN_2-1-5-4_fl.train]
File Edit Options Help
8 2 4
0 0
1 0 0 0
0 1
0 1 0 0
1 0
0 0 1 0
1 1
0 0 0 1
0.1 0.2
0.9 0.1 0.2 0.3
0.2 0.8
0.1 0.85 0.1 0.2
0.7 0.1
0.1 0.1 0.9 0.1
0.98 0.89
0.1 0.1 0.2 0.98

```

Рис. 34. Набор данных обучающей выборки для НС с конфигурацией 2-1-5-4

Проанализировать влияние объема обучающей выборки на скорость и точность обучения НС, как для случая целочисленных значений, так и для вещественных значений в составе обучающей выборки. Причем при формировании примеров обучающей выборки следует использовать различные значения признаков атаки, находящихся в интервале $[0 - 1]$. Причем для значения качественного показателя «низкий» интервал значений должен быть $[0 - 0.4]$, а для значения качественного показателя «высокий» интервал значений – $[0.6 - 1]$.

4.1. Задания к лабораторным работам

Задание на лаб. раб № 1

1. Ознакомиться с программой *Fann*.
2. Получить от преподавателя одну из логических функций двух переменных и ряд значений для параметра скорости обучения.
3. В соответствии с логической функцией сформировать файлы для обучения, тестирования и конфигурации НС вида 2-1-2-1.
4. Обучить НС на обучающей выборке с ошибкой, не превышающей значение 0,001. Зафиксировать число эпох обучения, необходимых для достижения заданной точности аппроксимации нейронной сетью исследуемой логической функции.
5. Показать преподавателю результаты обучения и файл с весами и порогами НС до и после обучения.
6. Повторить П. 3 - 5 для заданного ряда значений параметра скорости обучения НС с целью набора статистических данных о скорости обучения НС.

7. Сформулировать заключение о зависимости числа эпох от параметра скорости обучения.

Задание на лаб. раб № 2

Для заданной логической функции от двух переменных:

1. Сформировать файлы для обучения, тестирования и конфигурации НС вида 2-1-3-1, 2-1-4-1 и 2-1-5-1, т.е. для НС с тремя, четырьмя и пятью формальными нейронами в скрытом слое.
2. Обучить нейронную сеть на обучающей выборке с ошибкой, не превышающей значение 0,001.
3. Показать преподавателю результаты обучения и файл с весами и порогами НС до и после обучения.
8. Повторить П. 2 - 3 для нейронных сетей с конфигурацией в соответствии с п.1 для набора статистических данных о скорости обучения НС.
4. Сформулировать заключение о зависимости числа эпох обучения от числа ФН в скрытом слое НС.

Задание на лаб. раб № 3

Для заданной логической функции от двух переменных:

1. Сформировать файлы для обучения, тестирования и конфигурации НС с двумя, тремя и четырьмя скрытыми слоями.
2. Обучить НС на обучающей выборке с ошибкой, не превышающей значение 0,001.
3. Показать преподавателю результаты обучения и файл с весами и порогами НС до и после обучения.
9. Повторить П. 2-3 для нейронных сетей с конфигурацией в соответствии с п.1 для набора статистических данных о скорости обучения НС.
5. Сформулировать заключение о зависимости числа эпох обучения от числа скрытых слоев НС.

Задание на лаб. раб № 4

В соответствии с табл. 1:

1. Сформировать файлы для обучения, тестирования и конфигурации НС вида 2-1-3-4, 2-1-4-4 и 2-1-5-4, т.е. для НС с двумя входами, тремя, че-

тырьмя или пятью формальными нейронами в скрытом слое и четырьмя выходами.

2. Обучить НС на обучающей выборке с ошибкой, не превышающей значение 0,001.
3. Показать преподавателю результаты обучения и файл с весами и порогами НС до и после обучения.
4. Повторить П. 2 - 3 для получения статистических данных о скорости обучения НС.
5. Протестировать достоверность распознавания стратегии «Собака» при различных сочетаниях значений показателей «доля рынка» и «степень расширения рынка» в диапазоне значений [0 – 1].
6. Сформулировать заключение о достоверности распознавания угроз при различных сочетаниях значений признаков атаки в зависимости от достоверности исходных данных.

Литературные источники

1. Дюк. В., Самойленко А. Data Mining: учебный курс (+ CD). – СПб: Питер, 2001.
2. Negnevitsky M. Artificial intelligence: a guide to intelligent systems. Addison-Wesley, 2002.
3. Горбань А. Н. Обучение нейронных сетей. - М.: СП ПараГраф. 1991.
4. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика. - М.: Мир, 1992.

Содержание

Введение.....	4
1. Теоретическая часть	4
1.1. Биологический нейрон.....	5
1.2. Формальный нейрон – математическая модель нейрона	5
2. Возможности программы Fann.....	8
2.1. Активация среды моделирования Fann	8
2.2. Пользовательский интерфейс программы FannExplorer.....	9
Описание главного меню.....	9
Опция File	9
Опция View	10
Controller	10
Weight Graph.....	13
Error plot.....	14
Show all и Close All	14
Опция Tools	14
3. Описание среды моделирования	15
3.1. Работа с программой FANN	15
4. Практическая часть	27
4.1. Задания к лабораторным работам	29
Задание на лаб. раб № 1.....	29
Задание на лаб. раб № 2.....	30
Задание на лаб. раб № 3.....	30
Задание на лаб. раб № 4.....	30
Литературные источники.....	31