

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Биологический прототип	4
2 Структура и свойства искусственного нейрона.....	6
<u>2.1</u> История.....	6
<u>2.2</u> Искусственный нейрон	7
<u>2.3</u> Классификация нейронных сетей	9
3. Обучение искусственных нейронных сетей.....	12
<u>3.1</u> Обучение с учителем	12
<u>3.1.1</u> Концепция обучение с учителем	12
<u>3.1.2</u> Алгоритм обратного распространения ошибки	13
<u>3.2</u> Обучение без учителя	16
4. Рекуррентные нейронные сети	18
<u>4.1</u> Общая схема работы рекуррентных нейронных сетей	18
<u>4.2</u> Сеть Хопфилда	18
5. Программное обеспечение для моделирования нейронных сетей	21
<u>5.1</u> Обзор программного обеспечения для моделирования	21
<u>5.2</u> Пакет Fuzzy Logic Toolbox	21
<u>5.2.1</u> Общая характеристика.....	21
<u>5.2.2</u> Создание нечеткой нейронной сети	21
ЗАКЛЮЧЕНИЕ	26
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	27

ВВЕДЕНИЕ

В предложенной курсовой работе исследуется тема искусственных нейронных сетей. В настоящее время данная тема очень актуальна, что подтверждается массой различных применений. Это автоматизация процессов распознавания образов, адаптивное управление, аппроксимация функционалов, прогнозирование, создание экспертных систем, организация ассоциативной памяти и многие другие приложения.

Объектом данной курсовой работы является совокупность знаний об построение искусственного нейрона, алгоритмах построения искусственной нейронной сити, и анализ прикладного программного обеспечения для применения на практике.

Предметом исследования выступает механизм построения искусственной нейронной сети.

Целью работы является создания нечеткой нейронной сети.

Задачей является изучение видов сетей, алгоритмов построения.

Для изучения данной темы использовались такие методы как сравнительной анализ теоритического материала, метод аналогии при изучении видов нейронных сетей, а также моделирование нейронной сети.

В курсовой работе предлагается реализация электронного дидактического ресурса, который позволяет изучить понятие искусственной нейронной сети и алгоритмов построения, а также предлагается реализация построения нечеткой нейронной сети с помощью прикладной программы.

1 Биологический прототип

Развитие искусственных нейронных сетей имеет прямую зависимость с биологией.

Детальное устройство головного мозга стало известным около ста двадцати лет назад благодаря доктору Рамони Кайалу. Опытным путем было показано, что ткани человеческого мозга построены из большого количества связанных между собой однопольных узлов – нейронов. Они соединены между собой нервными волокнами. По ним химическим и электрическим путем передаются сигналы между нейронами. Примерно 10^{11} нейронов принимают участие в около 10^{15} связях, передающих импульсы. Данные связи могут иметь длину более метра.

Биологический нейрон характерен весьма сложной схемой строения. Центральная часть нейрона (сома) соединяется с центральной частью другого нейрона с помощью древовидных отростков (дендрит). Аксон самый крупный отросток, его задача состоит в том, чтобы обеспечивать передачу сигнала активности от сомы иным нейронам [4].

Прием сигналов происходит в точках соединения, принятых называть синапсами. Это небольшой промежуток, порядка 200 нано миллиметров, в месте соединения аксона и дендрит другого нейрона [1].

Протоплазма нейрона заряжена отрицательно в состоянии покоя, с потенциалом около 70 мВ. Заряд из-за поступления возбуждающих или тормозящих сигналов от дендрит деполяризуется до значения 60 мВ, в следствии происходит диффузия в сому положительно заряженных ионов натрия Na^+ . Заряд нейрон резко повышается до положительного, и затем через аксон распространяется возбуждение другим нейронам. После ядро приходит в исходное состояние.

Распространение возбуждения по нейронам головного мозга происходит не одновременно. Частота срабатывания колеблется между 1 и 100 Гц. Сигнал способен распространяется со скоростью от 0,5 до 2 м/с [3].

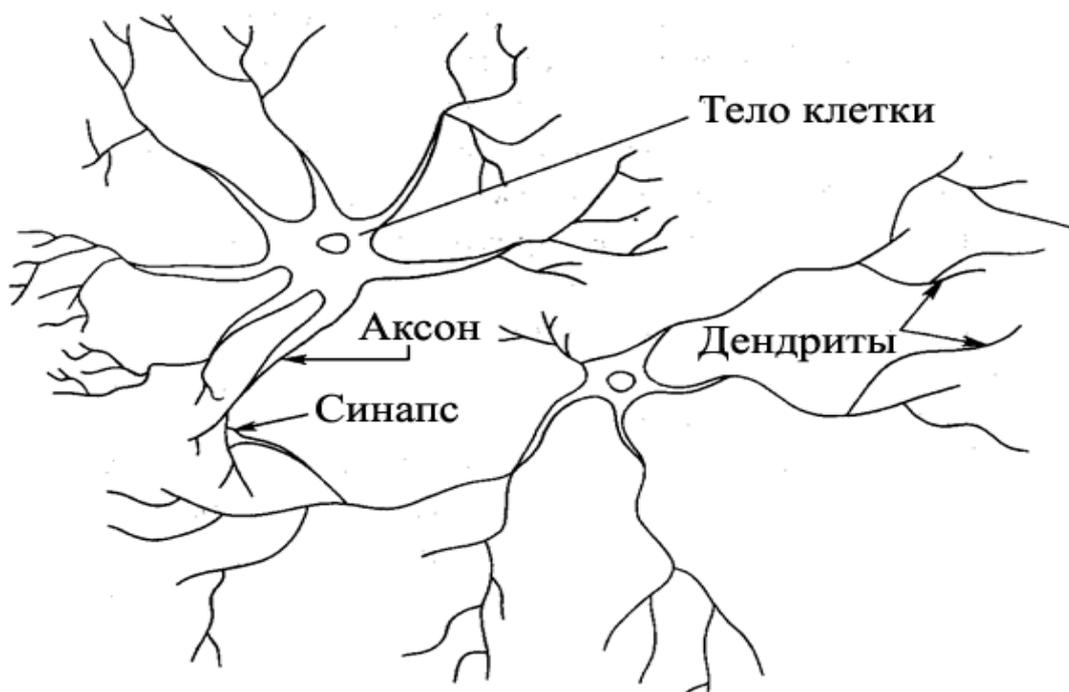


Рисунок 1 – логический нейрон

В действительности, функциональная схема работы нейрона имеет много усложнений и исключений. Большинство искусственных нейронных сетей моделируют только описанные выше свойства, так как знания об устройстве мозговой активности весьма ограничено

Разработчикам, занимающимся построением нейронных сетей, приходится использовать отличные от биологических знаний структуры. В результате искусственные нейронные сети отказываются от биологического правдоподобия[2].

2 Структура и свойства искусственного нейрона

2.1 История

В середине XX века возникает термин «нейронная сеть». Одними из первых работ в данной области были труды МакКаллока и Питтсома. В 1943 году была разработана компьютерная модель нейронной сети, в основу которой легли математические алгоритмы и теории деятельности головного мозга. Было выдвинуто предположение об упрощенной модели нейрона, как устройства, оперирующее двоичными числами. Каждый элемент $i = 1, 2 \dots n$ имеет входы и выходы, принимающие значения 0 или 1. Данная модель получила название «пороговой логикой» [7].

Как и биологической прототип, нейроны Мак-Каллока–Питтса благодаря подстройки параметров, отвечающих за синаптическую проводимость, были способны обучаться.

Эти системы (и множество им подобных) получили название *перцептронов*.

Влиянием остальных нейронов определяется состояние отдельного нейрона как взвешенная линейная комбинация:

$$\sum w_{ij}n_j$$

где n_j – вход нейрона, а w_{ij} – соответствующий этому входу вес.

К сумме затем применяется пороговая функция вида:

$$g(\Sigma < 0) = 0 ; g(\Sigma \geq 0) = 1 .$$

Маккаллох и Питтс показали, что такие сети могут производить произвольные вычисления. При этом единственной нерешенной проблемой оставался подбор весов w_{ij} [2].

В 1949 Д. Хебб предположил, что сила изменения синаптических связей влияет на способность обучения нейронной сети. По предложенному Д. Хеббом алгоритму испытываемая сеть спонтанно обучается выполнять поставленную задачу без вмешательства, происходит процесс самообучения.

В 1986 Дэвидом И. Румельхартом, Дж. Е. Хинтоном и Рональдом Дж. Вильямсом заново открыли и существенно развит метод обратного распространения ошибки. Был предложен эффективный алгоритм для обучения более совершенного, многослойного персептрона [7].

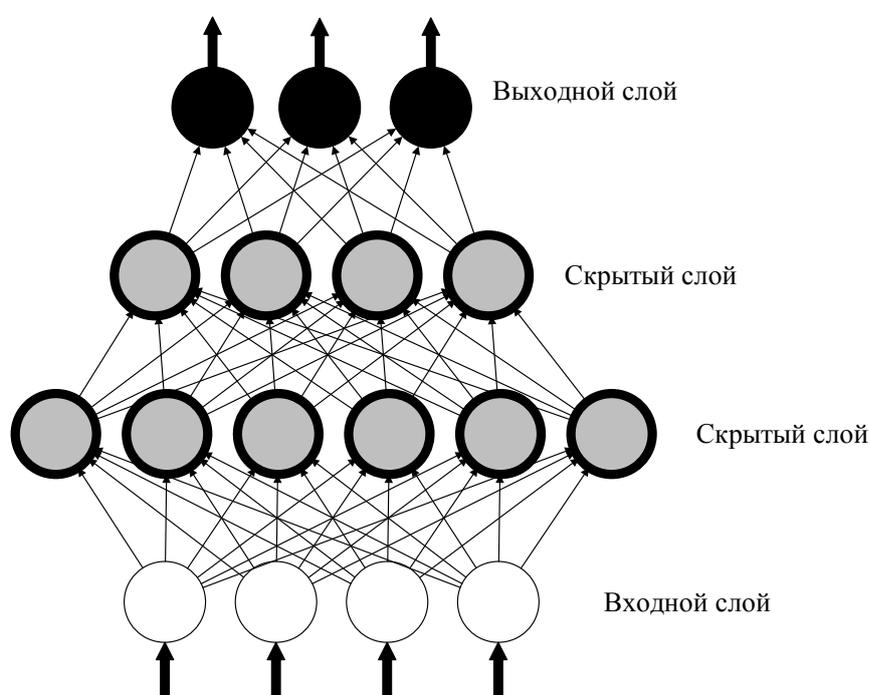


Рисунок 2 – Схема многослойного персептрона

2.2 Искусственный нейрон

Искусственный нейрон имитирует в первом приближении свойства биологического нейрона.

Составляющие - это элементы трех типов: умножителей (синапсов), сумматора и нелинейного преобразователя. Синапсы обеспечивают связь между нейронами, умножают входной сигнал на число, характеризующее силу связи, (вес синапса). Сумматор выполняет сложение сигналов, поступающих

по синаптическим связям от других нейронов, и входных сигналов извне. Функция активации или передаточная функция нейрона – это выход сумматора, который реализует нелинейную функцию одного аргумента [3].

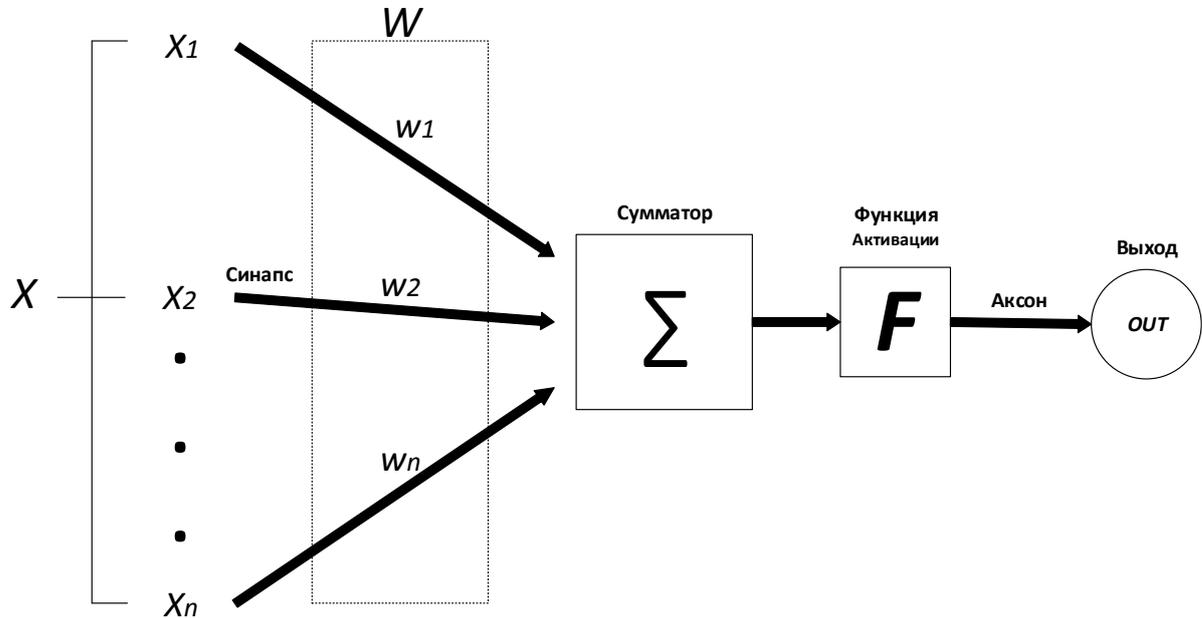


Рисунок 2 – Структура искусственного нейрона

Математическая модель нейрона:

$$s = \sum_{i=1}^n w_i x_i + b ; \quad y = f(s) ,$$

- где w_i - вес синапса;
 b - значение смещения;
 s - результат суммирования;
 x_i - компонент входного вектора (входной сигнал);
 y - выходной сигнал нейрона; n - число входов нейрона;
 f - нелинейное преобразование (функция активации).

Положительные веса в синаптических связях принято называть возбуждающими, а связи с отрицательными весами - тормозящими. В общем случае весовые коэффициенты, так же входной сигнал и смещение могут принимать действительные или фиксированные значения.

Наиболее часто используются следующие функции активации:

- Единичный скачок или жесткая пороговая функция.

Простая кусочно-линейная функция. Если входное значение меньше порогового, то значение функции активации равно минимальному допустимому, иначе – максимально допустимому.

- Линейный порог или гистерезис.

Несложная кусочно-линейная функция. Имеет два линейных участка, где функция активации тождественно равна минимально допустимому и максимально допустимому значению и есть участок, на котором функция строго монотонно возрастает;

- Сигмоидальная функция или сигмоид

Одна из наиболее распространенных функций активации – это нелинейная логистическая функция или сигмоид (функция S-образного вида):

$$f(s) = \frac{1}{1 + e^{-as}}.$$

- Гиперболический тангенс

Гиперболический тангенс является еще одним примером сигмоидальной функции активации, задаётся следующим выражением:

$$f(s) = th\left(\frac{s}{a}\right),$$

где a – это параметр, влияющий на наклон сигмоидальной функции [5].

2.3 Классификация нейронных сетей

Можно выделить три типа нейронных сетей, в зависимости от функций:

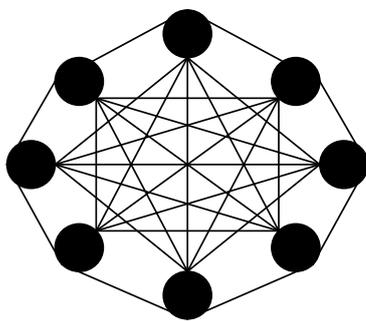
1. входные нейроны, на которые подается вектор, кодирующий входное воздействие или образ внешней среды;
2. выходные нейроны, выходные значения которых представляют выходы нейронной сети;

3. промежуточные нейроны, составляющие основу нейронных сетей.

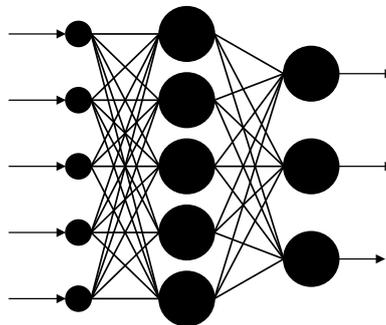
Тип модели нейрона в сети зависит от расположения в ней. Происходит некоторая переработка информации за счет преобразования входного вектора во время функционирования сети. Вид выполняемого преобразования данных зависит так же от особенности архитектуры сети [2].

С точки зрения топологии можно выделить три основных типа нейронных сетей:

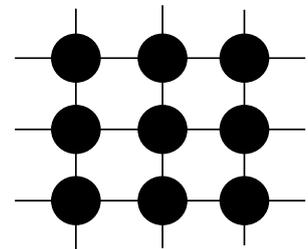
1. Полносвязные — каждый нейрон передает выходной сигнал и самому себе, и остальным нейронам;
2. Многослойные — совокупность нейронных сетей образует слой с единым входным сигналом;
3. Слабосвязные — нейроны располагаются в узлах прямоугольной или гексагональной решетки.



а) полносвязная сеть



б) многослойная сеть с последовательными связями



в) слабосвязная сеть

Рисунок 3 – Архитектуры нейронных сетей

Многослойные нейронные сети можно разделить на следующие типы:

1. Монотонные. В монотонных типах сети каждый слой, за исключением выходного, разбит на два блока. Блоки возбуждающие и блоки, тормозящие так же соединенные между собой либо тормозящими, или возбуждающими связями. Если от блоков одного вида к блокам другого вида ведет только возбуждающие связи, то это означает, что выходной сигнал это монотонно

неубывающая функция, а если сигнал тормозящий, то монотонно не убывающая. Для монотонных сетей необходима монотонная зависимость между входным и выходным сигналом;

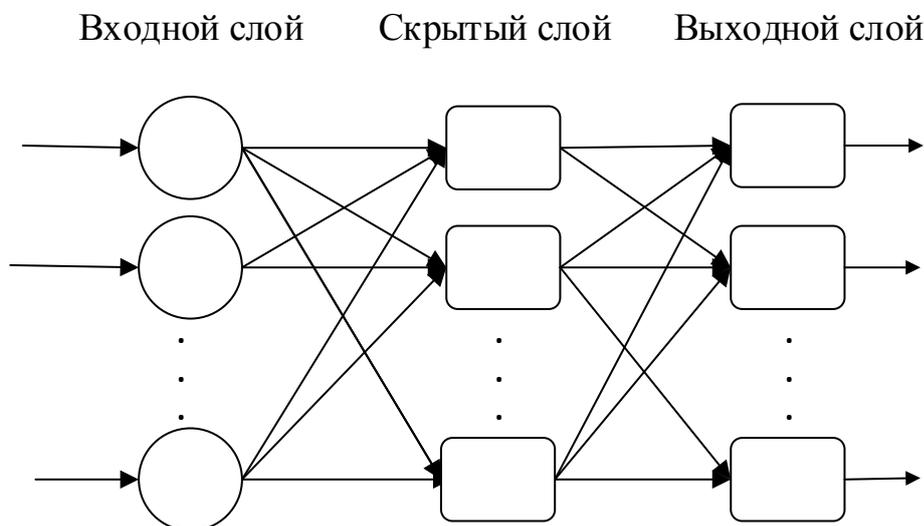


Рисунок 4 – Многослойная (двухслойная) сеть прямого распространения

2. Сети без обратных связей. Данный тип характерен тем, что от первого скрытого слоя сигнал передается последующим слоям, с последующей передачей сигнала интерпретатору;

3. Сети с обратными связями. В данном типе сети информация с последующих слоев передается на предыдущие.

Существует классификация нейронных сетей на синхронные и асинхронные. В синхронных сетях в отдельный момент времени только один нейрон меняет состояние. В асинхронных сетях состояние изменяется у совокупности нейронов [3].

3. Обучение искусственных нейронных сетей

3.1 Обучение с учителем

3.1.1 Концепция обучение с учителем

Обучение с учителем характеризуется тем, что для каждого входного вектора существует целевой вектор, представляющий собой требуемый выход. Вместе они образуют обучающую пару. Обычно сеть обучается на некотором числе таких обучающих пар.

Предъявляется выходной вектор, вычисляется выход сети и сравнивается с соответствующим целевым вектором, разность (ошибка) с помощью обратной связи подается в сеть и веса изменяются в соответствии с алгоритмом, стремящимся минимизировать ошибку. Векторы обучающего множества предъявляются последовательно, вычисляются ошибки и веса подстраиваются для каждого вектора до тех пор, пока ошибка по всему обучающему массиву не достигнет приемлемо низкого уровня [9].

На основе встроенных знаний учитель может сформировать и передать обучаемой нейронной сети желаемый отклик, соответствующий данному входному вектору. Параметры сети корректируются с учетом обучающего вектора и сигнала ошибки [2].

Сигнал ошибки – это разность между желаемым сигналом и текущим откликом нейронной сети. После окончания обучения учителя можно отключить и позволить нейронной сети работать со средой самостоятельно [8].

Системе предъявляется эталонный образ. Если результат распознавания совпадает с заданным, весовые коэффициенты связей не изменяются. Если искусственная нейронная сеть неправильно распознает результат, то весовым коэффициентам дается приращение в сторону повышения качества распознавания.

Существует несколько подходов, один из них - разработка наборов выходных сигналов, соответствующих входным, для каждого слоя нейронной сети. Второй вариант - динамическая подстройка весовых

коэффициентов синапсов, в ходе которой выбираются, как правило, наиболее слабые связи и изменяются на малую величину, а сохраняются только те изменения, которые повлекли уменьшение ошибки на выходе всей сети. Третий вариант - распространение сигналов ошибки от выходов нейронной сети к ее входам, в направлении, обратном прямому распространению сигналов в обычном режиме работы. Этот алгоритм обучения получил название процедуры обратного распространения ошибки (error back propagation) [1].

3.1.2 Алгоритм обратного распространения ошибки

В 1986 г. Румельхарт, Хинтон и Вильямс предложили алгоритм обратного распространения ошибки. Обучение многослойного персептрона основано на минимизации функции ошибки сети $E(w)$. Ошибка определяет отклонение желаемых выходов сети t^n от получившихся y^n . Обычно функция $E(w)$ определяется методом наименьших квадратов:

$$E(w) = \frac{1}{2} \sum_n (t^n - y^n)^2 \quad (1.1)$$

Суть метода заключается в том, что двигаясь в направлении, противоположном градиенту функции Δw_{ji} , приближаемся к минимуму, возможно локальному, функции $E(w)$:

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}}, \quad (1.2)$$

где $0 < \eta < 1$ – задает скорость обучения;
 w_{ji} – коэффициент связи i го нейрона слоя $n - 1$ с j м нейроном слоя n .

Каждый нейрон рассчитывает взвешенную сумму своих входов:

$$a_i = \sum_n w_{ij} z_i \quad (1.3)$$

Где z_i – это входы нейрона и выходы предыдущего слоя нейронов в соответствии с рисунком 5а.

Выход нейрона j – преобразование суммы a_j пороговой функцией g :

$$z_j = g(a_j). \quad (1.4)$$

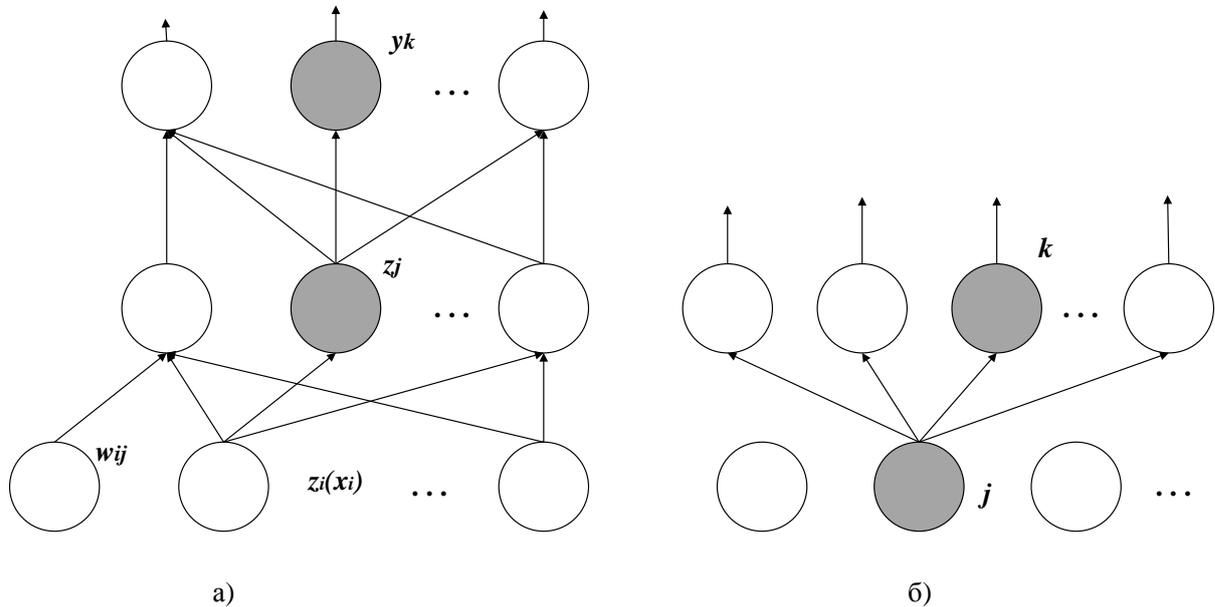


Рисунок 5 – Схема нейронной сети алгоритма обратного распространения

$E(w)$ является сложной функцией, для которой w :

$$w = w(a_j) .$$

Отсюда по правилу вычисления производной сложной функции:

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial a_j} \frac{\partial a_j}{\partial w_{ji}}, \quad \frac{\partial a_j}{\partial w_{ji}} = \frac{\partial \sum w_{ji} z_i}{\partial w_{ji}} = z_i .$$

Для вычисления градиента на каждом слое i вычисляется произведение величины ошибки предыдущего i , находящегося сверху слоя j , на входное значение слоя:

$$\frac{\partial E}{\partial w_{ji}} = \delta_j z_i . \quad (1.5)$$

Для выходного слоя δ_k можно получить:

$$\delta_k = \frac{\partial E}{\partial a_k} = \left| \begin{array}{l} a_k = \sum_j w_{jk} z_j \\ z_k = g(a_k) \end{array} \right| = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial a_k}. \quad (1.6)$$

Для логического сигмоида

$$g(a) = \frac{1}{1 + e^{-a}},$$

такая производная будет иметь довольно простой вид:

$$g'(a) = g(a)(1 - g(a)).$$

Второй сомножитель $\frac{\partial E}{\partial y_k}$ приводится к выражению:

$$\frac{\partial E}{\partial y_k} = y_k - t_k.$$

В итоге формула вычисления ошибки на выходном слое имеет вид:

$$\delta_k = g'(y_k - t_k). \quad (1.7)$$

Для промежуточного слоя δ_j получается аналогичным способом:

$$\delta_j = \frac{\partial E}{\partial a_j} = \left| \begin{array}{l} a_k = \sum_j w_{jk} z_j \\ z_j = g(a_j) \end{array} \right| = \sum_k \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial z_j} \frac{\partial z_j}{\partial a_j}. \quad (1.8)$$

Суммирование происходит по всем k , к которым нейрон j посылает сигнал согласно рисунку 5.

В итоге:

$$\delta_j = g'(a_j) \sum_k \delta_k w_{jk}. \quad (1.9)$$

Ошибка на каждом слое вычисляется рекурсивно через значения ошибки на предыдущих слоях [1].

Алгоритм обратного распространения ошибки:

1. Перед началом работы алгоритма веса w_{ij} инициализируем случайными значениями;
2. Подаем на вход персептрона вектор x^n из обучающей выборки и получаем значение y^n на выходе по формулам (1.4) и (1.3);
3. Вычисляем ошибки δ_k для выходов сети по (1.7);
4. Вычисляем ошибки δ_j для всех скрытых слоев по (1.9);
5. Находим значение градиента по (1.5);
6. Корректируем значения синаптических весов по (1.2);
7. Вычисляем значение ошибки (1.1). Если величина ошибки не устраивает, то повторяем алгоритм.

3.2 Обучение без учителя

Главная черта, делающая обучение без учителя привлекательным, это его самостоятельность. Процесс обучения, как и в случае обучения с учителем, заключается в подстраивании весов синапсов. Некоторые алгоритмы, правда, изменяют и структуру сети, т. е. количество нейронов и их взаимосвязи, но такие преобразования правильнее назвать более широким термином - самоорганизацией. Очевидно, что подстройка весов синапсов может проводиться только на основании информации о состоянии нейронов и уже имеющихся весовых коэффициентов). На этом, в частности, по аналогии с известными принципами самоорганизации нервных клеток, построены алгоритмы обучения Хебба.

Простейшая реализация правила Хебба называется сигнальным методом:

$$\Delta w_{ij} = \eta z_i^{n-1} z_j^n \quad (2.1)$$

Где Δw_{ij} – приращение веса;
 z_i^{n-1}, z_j^n – выходные значения нейронов i и j ;
 $\eta > 0$ – задает скорость обучения.

Более совершенной вариацией правила Хебба является дифференциальный метод:

$$\Delta w_{ij} = \eta (z_i^{n-1} - z_i^{*n-1})(z_j^n - z_j^{*n}) \quad (2.2)$$

Здесь добавляется z_i^{*n-1} и z_j^{*n} – выходные значения нейронов i и j на предыдущей итерации обучения.

Существует третий метод обучения – обучение с забыванием. Введем понятие коэффициента забывания γ – часть синаптического веса, которая «забывается» каждым нейроном на каждой итерации [10]. Тогда обучение выполняется согласно выражению

$$\Delta w_{ij}(t + 1) = (1 - \gamma)w_{ij}(t) + \eta \cdot z_i^{n-1}z_j^n . \quad (2.3)$$

Алгоритм обучения нейронной сети по правилу Хебба:

0. Инициуем веса w_{ij} случайными величинами малой величины;
1. Подаем на входы сети все векторы из $\{x^n\}$ по очереди и получаем выходные значения z для всех нейронов сети;
2. На основании полученных значений z по формулам (2.1) или (2.2), или (2.3) производится изменение весов w_{ij} всей сети;
3. Переходим к началу алгоритма, исключая нулевой шаг, до тех пор, пока выходные значения сети не стабилизируются с заданной точностью.

Так как нейронная сеть – сеть прямого распространения, то она рано или поздно придет в устойчивое состояние [1].

4. Рекуррентные нейронные сети

4.1 Общая схема работы рекуррентных нейронных сетей

В рекуррентных нейронных сетях появляются обратные связи между нейронами от выходов обратно на входы. За счет обратных связей выходы всей сети в различные моменты времени при одних и тех же входах сети могут принимать произвольные состояния [11].

На первом этапе работы рекуррентной сети, согласно рисунку 6, на вход нейрона подаются входные значения x_j и вычисляется выход нейрона z . Полученное выходное значение подается на вход нейрона вместе с другими значениями. Происходит вычисление нового значения на выходе. Процесс повторяется, пока входное значение нейрона будет иметь малое изменение.

Рекуррентные нейронные сети, для которых возможно получить стабилизирующиеся к определенному значению выходы, называются устойчивыми, в другом случае - неустойчивыми. Неустойчивые сети малоприспособны для практического применения.

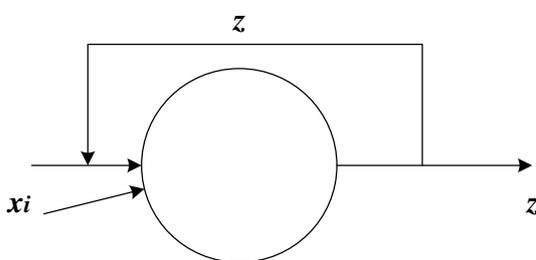


Рисунок 6 – Нейрон с обратной связью

4.2 Сеть Хопфилда

В 1982 г. Д. Хопфилд предложил устойчивую рекуррентную нейронную сеть со схемой согласно рисунку 7. Сеть оперирует только величинами $\{-1; 1\}$. В сети имеется один слой настраиваемых весов w_{ij} .

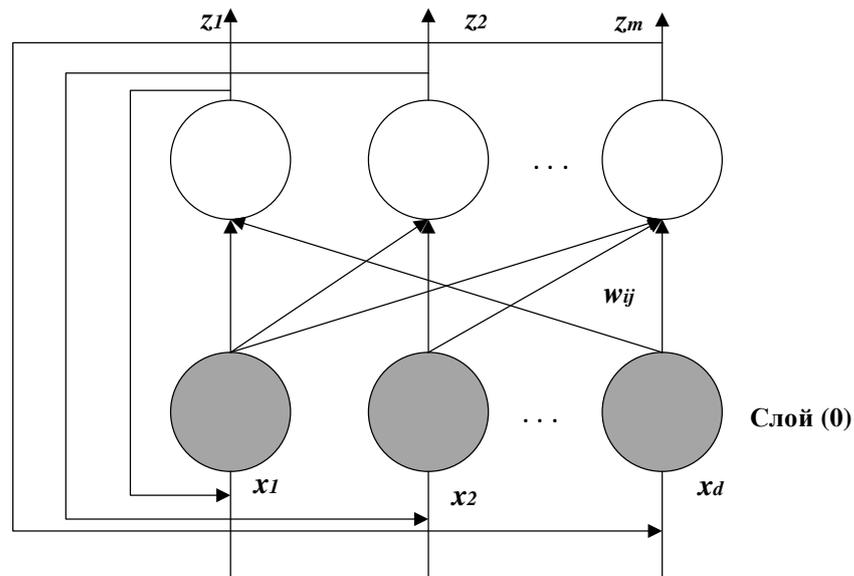


Рисунок 7 – Сеть Хопфилда

Все нейроны единственного слоя возвращают свои выходы на свой вход и входы всех остальных нейронов сети посредством распределителей (не нейронов) слоя (0) [2].

Каждый из нейронов выполняет ниже описанные шаги:

- 1) вычисляет взвешенную сумму a своих входов:

$$a_j = \sum_{i \neq j}^M (w_{ji} z_i) + x_j ;$$

- 2) к сумме применяется нелинейная пороговая функция $g(a)$:

$$z_j = g(a) = \left\{ \begin{array}{l} 1, \quad a_j > T_j \\ -1, \quad a_j < T_j \\ \text{не меняется } a_j = T_j \end{array} \right\}.$$

Функционирование сети происходит таким образом, что входные сигналы подается на вход один раз, а дальнейшая работа происходит за счет

возвращаемых выходов. Изменение состояний всех нейронов происходит одновременно.

Для n выходов соответственно существует 2^n состояний и n -мерный гиперкуб. При поступлении на вход нового входного вектора сеть начинает переходить из вершины в вершину, пока не стабилизируется в какой-либо вершине.

Устойчивая вершина z для данного входа x задается матрицей настраиваемых весов w_{ji} и порогами T_j . Если входной вектор x частично подвергнуть искажению, то сеть Хопфилда приходит именно в состояние z [11].

5. Программное обеспечение для моделирования нейронных сетей

5.1 Обзор программного обеспечения для моделирования

Для моделирования нейронных сетей существуют два вида программ универсальные и прикладные (Neural Network Simulators). Универсальные среды позволяют синтезировать оптимальные нейронные сети, применяемые для решения широкого класса задач. Прикладные среды моделирования ориентированы для синтеза нейронных сетей, применяемых в той или иной отрасли, прикладной области или специфичной задаче [2].

5.2 Пакет Fuzzy Logic Toolbox

5.2.1 Общая характеристика

Fuzzy Logic Toolbox используется в системе MATLAB, является пакетом прикладных программ.

Основные возможности пакета:

- построение адаптивных нечетких систем (нечетких нейронных сетей);
- построение систем нечеткого вывода;
- интерактивное динамическое моделирование в Simulink.

Пакет позволяет работать в режиме графического интерфейса, в режиме командной строки, а также с использованием блоков и примеров пакета Simulink [3].

5.2.2 Создание нечеткой нейронной сети

Графический интерфейс нечетких нейронных систем вызывается функцией `anfisedit` из режима командной строки. С помощью данного редактора осуществляется создание или загрузка структуры нечеткой нейронной сети, просмотр структуры, настройка ее параметров, проверка качества функционирования такой сети.

Создания нечеткой нейронной сети, отображающей зависимость между переменными x и y :

$$y = x^2.$$

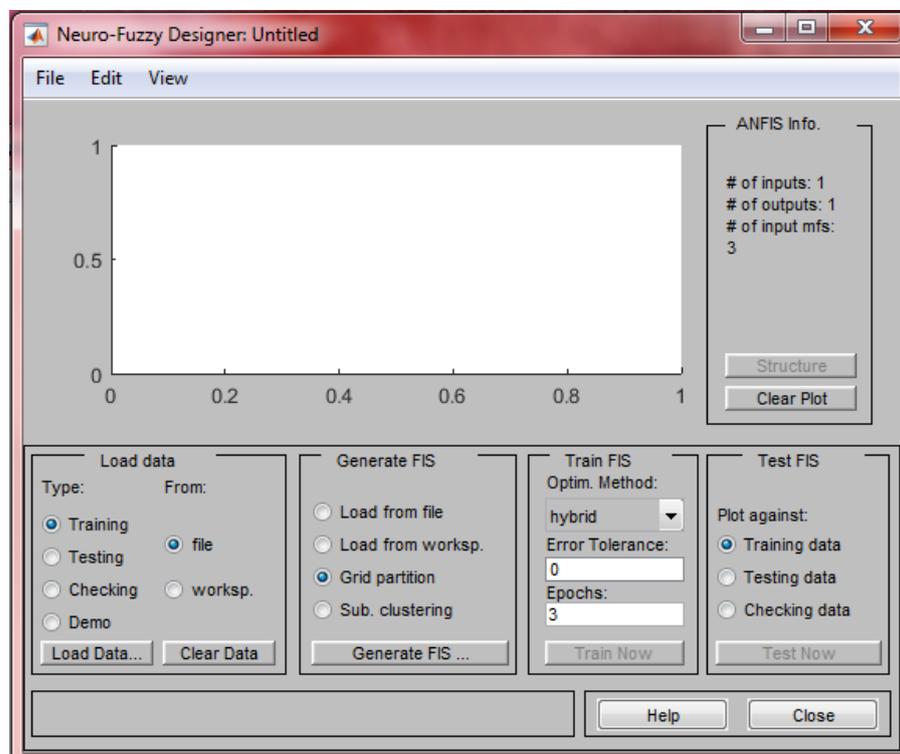


Рисунок 8 – Окно редактора нечеткой нейронной сети

Таблица 1 – Входные данные

x	-1	-0,6	0	0,4	1
y	1	0,36	0	0,16	1

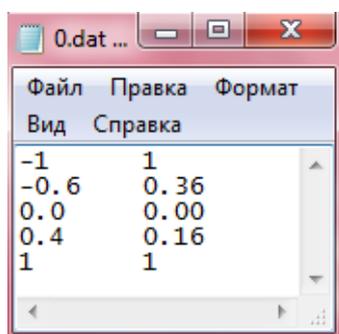


Рисунок 9 – файл с входными данными

Создание структуры, настройка параметров и проверка осуществляются по обучающей (Training), проверочной (Checking) и тестирующей (Testing) выборкам, которые предварительно должны быть представлены в виде текстовых файлов (с расширением .dat и разделителями-табуляциями).

Набор опций Load data (Загрузить данные) включает:

- Type - тип загружаемых данных (Training - для обучения, Testing - для тестирования, Checking - для проверки, Demo - демонстрационные);
- Disk (Диск) или Workspace (Рабочая область) - место, откуда должны загружаться данные.

Группа опций ANFIS-редактора объединена под именем Generate FIS (Создание нечеткой системы вывода).

Группа опций Train FIS (Обучение нечеткой системы вывода) позволяет определить метод обучения (Optim. Method) нечеткой нейронной сети: гибридный (Hybrid) или обратного распространения ошибки (Backpropagation); установить уровень текущей суммарной, по всем образцам, ошибки обучения (Error Tolerance), при достижении которого процесс обучения заканчивается; количество циклов обучения (Epochs).

Опция Train Now позволяет начать процесс обучения нечеткой нейронной сети.

Опции Test FIS позволяют провести проверку и тестирование созданной и обученной системы с выводом результатов в виде графиков (для обучающей выборки - Training data, тестирующей выборки - Testing data и проверочной выборки - Checking data) Кнопка Test Now позволяет запустить указанные процессы.

Создание и проверку нечеткой нейронной сети проведем по этапам.

В окне ANFIS-редактора выберем тип загружаемых данных Training и нажмем кнопку Load data. Открытие файла приводит к появлению в графической части окна редактора набора точек, соответствующих введенным данным.

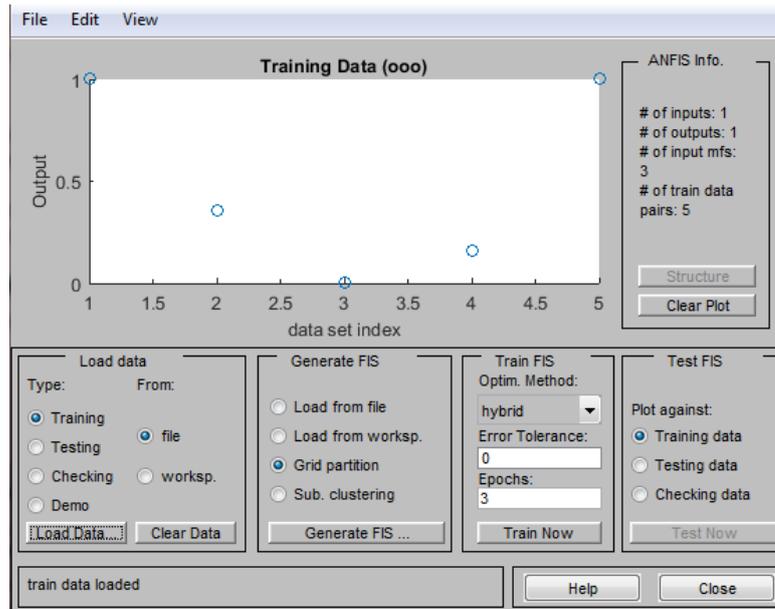


Рисунок 10 – Окно ANFIS-редактора после загрузки обучающей выборки

В группе опций Generate FIS по умолчанию активизирована опция Grid partition

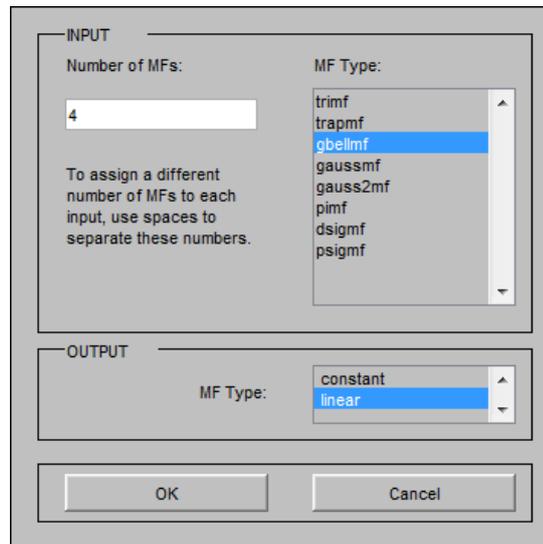


Рисунок 11 – Окно задания функций принадлежности

В опции Train FIS значения оставить по умолчанию, увеличить количество циклов на 40. Результат в виде графика ошибки чети в зависимости от числа циклов обучения. Из которого следует, что обучение фактически закончилось после пятого цикла согласно рисунку 13.

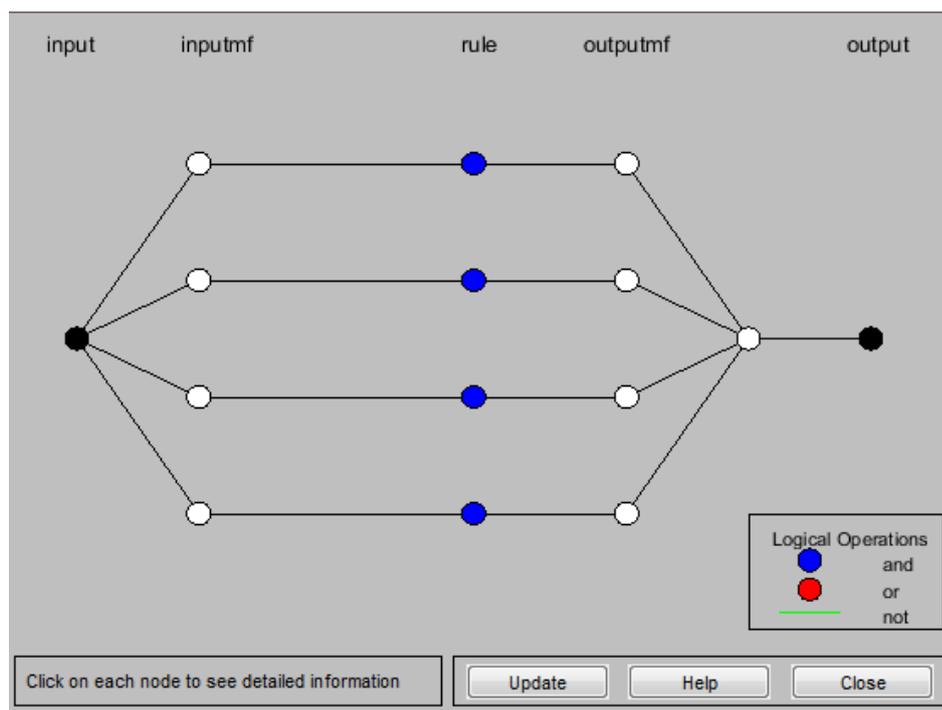


Рисунок 12 – Структура созданной гибридной сети

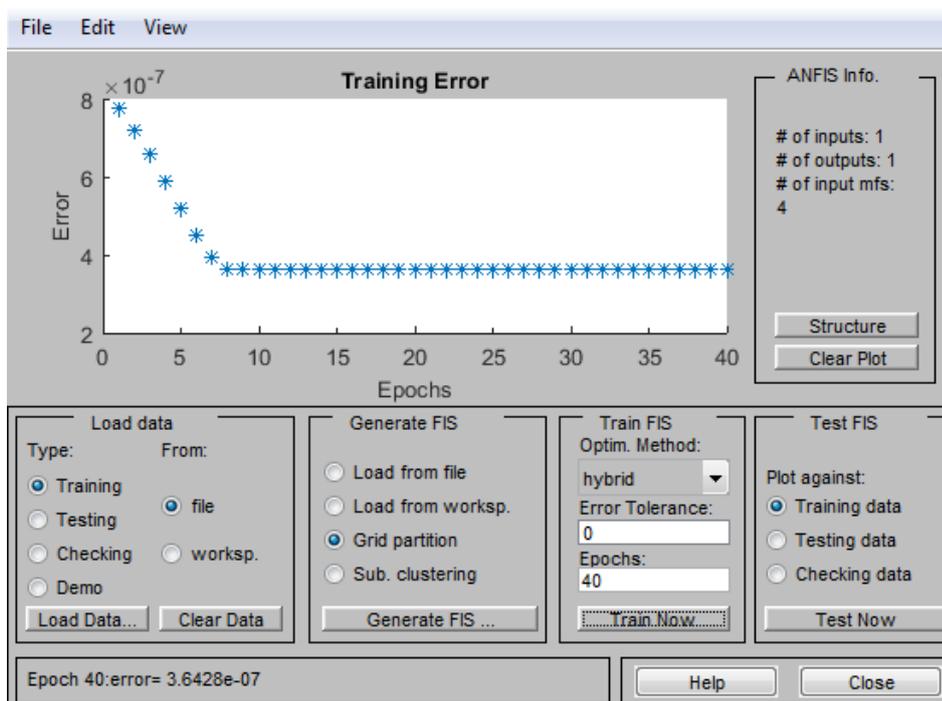


Рисунок 13 – Результат обучения сети

Выходы обученной сети практически совпадают со значениями обучающей выборки.

ЗАКЛЮЧЕНИЕ

Цель курсовой работы – реализация построения нейронной сети с помощью прикладного программного обеспечения – достигнута.

В рамках курсовой работы изучены математический аппарат построения искусственной нейронной сети, основные алгоритмы построения. Реализовано построение нечеткой нейронной сети с помощью прикладного программного обеспечения.

Важно отметить, что тип построения сети зависит от функций, выполняемых этой нейронной сетью. Так же из входных данных немо исключить не относящиеся к решаемой задаче избыточные данные.

В курсовой работе предлагается исследование пакет Fuzzy Logic Toolbox. Проведено успешное построение нечеткой нейронной сети. Данный пакет полностью автоматизирован, что позволяет рассматривать аппарат нечетких нейронных сетей как средство извлечения знаний из баз данных.

Также возможно использование данного теоритического материала для дальнейших исследований в области искусственных нейронных сетей, и применения его для создания программного обеспечения.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Аксенов С.В., Новосельцев В.Б. Организация и использование нейронных сетей (методы и технологии) / Под общ. ред. В.Б. Новосельцева. – Томск: Изд-во НТЛ, 2006. – 128 с.
2. Круглов В.В., Борисов В. В. Искусственные нейронные сети. Теория и практика. - 2-е изд., стереотип. - М.: Горячая линия-Телеком, 2002. - 382 с.
3. Круг П.Г. Нейронные сети и нейрокомпьютеры: Учебное пособие по курсу «Микропроцессоры». – М.: Издательство МЭИ, 2002. – 176 с.
4. Нормальная физиология: учебник / Орлов Р.С., Ноздрачев А.Д. - 2-е изд., исправл. и доп. 2010. - 832 с.
5. Комашинский В.И., Смирнов Д.А. Нейронные сети и их применение в системах управления и связи.-М.:Горячаялиния. Телекомб 2002.-94с.
6. Горожанина, Е.И. Г Нейронные сети. Учебное пособие. [Текст] / Е.И. Горожанина. – Самара. ФГБОУ ВО ПГУТИ, 2017. – 84 с.
7. Розенблатт Ф. Принципы нейродинамики. Перцептрон и теория механизмов мозга. – М.: Мир, 1965. – 480 с.
8. Пилюгина К.Н. Применение нейронных сетей с целью обнаружения вторжений // Современные научные исследования и инновации. 2016. № 2 [Электронный ресурс]. URL: <http://web.snauka.ru/issues/2016/02/63248> (дата обращения: 28.10.2017).
9. Уоссермен Ф. Нейрокомпьютерная техника: теория и практика. – 1992. – 184 с.
10. Hertz J., Krogh A., Palmer R. Wstep do teorii obliczen neuronowych. Wyd. II. – Warszawa: WNT, 1995.
11. Уоссермен Ф. Нейрокомпьютерная техника: теория и практика. – 1992. – 184 с. (<http://www.neuropower.de/rus/>).