

**МИНИСТЕРСТВО ПО РАЗВИТИЮ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН**

**НУКУССКИЙ ФИЛИАЛ ТАШКЕНТСКОГО УНИВЕРСИТЕТА
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ИМЕНИ МУХАММАДА АЛ-
ХОРЕЗМИ**

Кафедра «Компьютер инжиниринг»

Допуск к защите

зав. кафедрой _____

2019 г. «____» _____

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему «Распознавание образов с помощью нейронных сетей в среде
Matlab»

Выпускник:

У. Аминов

Научный руководитель:

д.т.н. К.Сеитназаров

НУКУС - 2019 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
ГЛАВА 1. Математические основы.....	5
1.2. Математическая модель искусственного нейрона.....	5
1.3. Архитектуры искусственных нейронных сетей	7
1.4. Парадигмы обучения искусственных нейронных сетей	8
1.5. Подбор оптимальной архитектуры сети	10
1.6. Сравнительный анализ различных алгоритмов обучения искусственных нейронных сетей.....	11
ГЛАВА2. Области применения искусственных нейронных сетей	13
2.1. Распознавание и классификация образов	13
2.2. Построение модели поведенческого скоринга.....	14
2.3. Другие области использования искусственных нейронных сетей	18
ГЛАВА3. Программное обеспечение для работы с искусственными сетями.....	20
3.1. Multiple Back-Propagation	20
3.2. Пакет STATISTICA Neural Networks	20
3.3. Matlab Neural Network Toolbox	21
3.4. Распознавание образов с помощью нейронных сетей в среде matlabr2009b	
3.5. Распознавание лиц на основе нейронной сети использование Matlab.....	47
Заключения	56
Список литературы.....	59

ВВЕДЕНИЕ

За последние несколько лет интерес к искусственным нейронным сетям возрастает ввиду возможностей их применения в задачах распознавания образов, задачах управления, прогнозирования и т. д. Алгоритмы обучения построены на основе принципов организации и функционирования биологических нейронных сетей. Мозг человека состоит из очень большого числа *нейронов*, соединенных многочисленными связями. Каждый нейрон обладает множеством свойств, общих с другими органами тела. Однако ему присущи абсолютно уникальные способности: принимать, обрабатывать и передавать электрохимические сигналы по нервным путям, которые образуют коммуникационную систему мозга. Искусственные нейронные сети индуцированы биологией, так как состоят из элементов, функциональные возможности которых аналогичны большинству элементарных функций биологического нейрона.

Распознаванием и классификацией образа будем называть его идентификацию и отнесение к соответствующему классу данных. При решении этой задачи нейронная сеть может выполнять функцию как экстрактора свойств, так и классификатора, приписывающего образ конкретному классу. Однако чаще всего экстракция свойств производится на отдельном этапе предварительного преобразования данных. В качестве примера рассмотрим определение достоверности цифровой подписи.

Определение достоверности цифровой подписи - это последовательность действий, в результате которых дается ответ на вопрос: принадлежит данная подпись конкретному человеку или нет. В примере использован пакет STATISTICA Neural Networks.

Области использования искусственных нейронных сетей.

- Прогнозирование финансовых временных рядов (компания LBS Capital Management объявила о значительных успехах в финансовых операциях,

достигнутых за счет прогнозирования цен акций с помощью многослойных персепtronов).

- Повышение эффективности процесса добычи полезных ископаемых (выделение значимых факторов, влияющих на показатели эффективности добычи).

- Оптическое распознавание символов, включая распознавание подписи (например, система идентификации подписи, учитывая не только окончательный ее рисунок, но и скорость движения пера на различных участках, что значительно затрудняет подделку чужой подписи).

ГЛАВА 1. МАТЕМАТИЧЕСКИЕ ОСНОВЫ

1.1 Математическая модель искусственного нейрона

Нейронная сеть - совокупность нейронных элементов и связей между ними. Основной элемент нейронной сети - это формальный нейрон, который представляет собой единицу обработки информации в нейронной сети[1]. На рисунке 1.1 показана модель нейрона, лежащего в основе искусственных нейронных сетей.

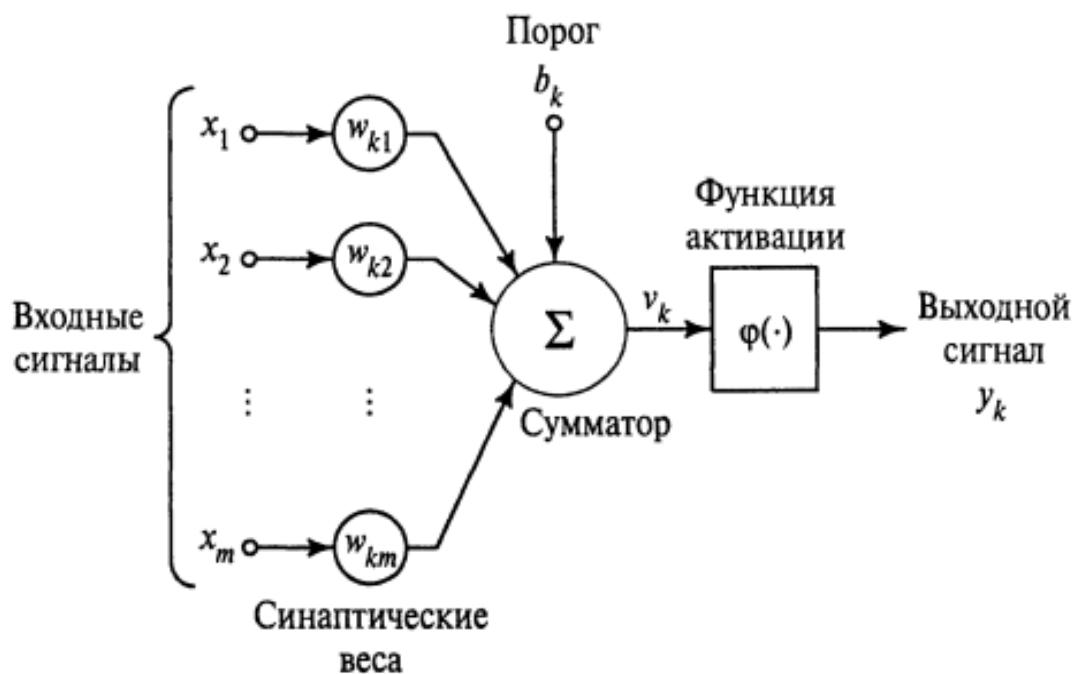


Рисунок 1.1. Нелинейная модель нейрона

В этой модели можно выделить три основных элемента[3]

1) Набор синапсов или связей, каждый из которых характеризуется своим весом или силой. В частности, сигнал x_j на входе синапса j , связанного с нейроном k , умножается на вес w_{kj} . Важно обратить внимание на то, в каком порядке указаны индексы синаптического веса w_{kj} . Первый индекс относится к рассматриваемому нейрону, а второй — ко входному окончанию синапса, с которым связан данный вес. В отличие от синапсов мозга синаптический вес

искусственного нейрона может иметь как положительные, так и отрицательные значения.

2) Сумматор складывает, входные сигналы, взвешенные относительно соответствующих синапсов нейрона. Эту операцию можно описать как линейную комбинацию.

3) Функция активации ограничивает амплитуду выходного сигнала нейрона. Эта функция также называется функцией сжатия. Обычно нормализованный диапазон амплитуд выхода нейрона лежит в интервале $[0,1]$ или $[-1,1]$.

В модель нейрона, показанную на рисунке 1.1, включен пороговый элемент, который обозначен символом $^b k$. Эта величина отражает увеличение или уменьшение входного сигнала, подаваемого на функцию активации.

В математическом представлении функционирование нейрона к можно описать следующей парой уравнений:

$$u_k = \sum_{j=1}^m \omega_{kj} x_j, \quad (1.1)$$

$$y_k = \varphi(u_k + b_k). \quad (1.2)$$

где x_1, x_2, \dots, x_m – входные сигналы;

$\omega_{k1}, \omega_{k2}, \dots, \omega_{km}$ – синаптические веса нейрона;

$\varphi()$ – функция активации;

u_k – линейная комбинация входных воздействий (индуцированное локальное поле);

y_k – выходной сигнал нейрона.

Функции активации $\varphi()$ определяют выходной сигнал нейрона в зависимости от индуцированного локального поля u_k . Можно выделить три основных типа функций активации.

а) Функция единичного скачка, или пороговая функция. Этот тип функции показан на рисунке 1.2, а и описывается следующим образом:

$$\varphi = \begin{cases} 1, & \text{если } v \geq 0; \\ 0, & \text{если } v < 0 \end{cases} \quad (1.3)$$

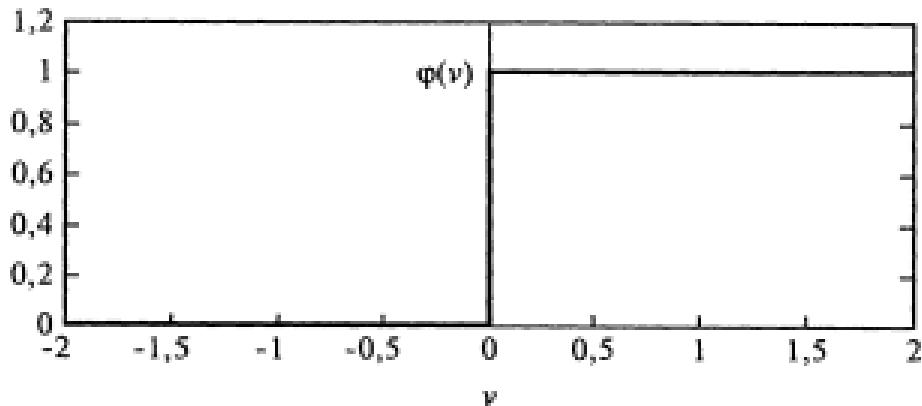


Рисунок 1.2 - Функция единичного скачка

б) Кусочно-линейная функция. Кусочно-линейная функция, показанная на рисунке 1.3, описывается следующим выражением (1.4), где коэффициент усиления в линейной области оператора предполагается равным единице.

$$\varphi = \begin{cases} 1, & \text{если } v \geq \frac{1}{2}; \\ |v|, & \text{если } -\frac{1}{2} > v > -\frac{1}{2}; \\ 0, & v \leq -\frac{1}{2}. \end{cases} \quad (1.4)$$

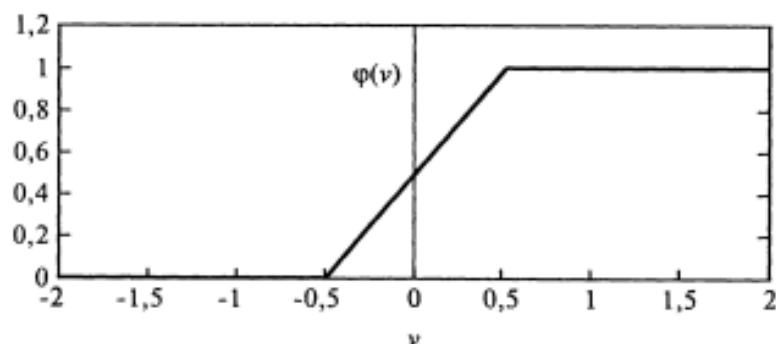


Рисунок 1.3 - Кусочно - линейная функция

в) Сигмоидальная функция, график которой напоминает букву *S*, является, самой распространенной функцией, используемой для создания искусственных нейронных сетей. Это быстро возрастающая функция, которая поддерживает баланс между линейным и нелинейным поведением. Примером

сигмоидальной функции может служить логистическая функция, задаваемая следующим выражением:

$$\varphi(v) = \frac{1}{1 + \exp^{-\alpha v}} \quad (1.5)$$

где α — параметр наклона сигмоидальной функции.

Изменяя этот параметр, можно построить функции с различной крутизной (см. рисунок 1.4). Первый график соответствует величине параметра, равной $\alpha/4$. В пределе, когда параметр наклона достигает бесконечности, сигмоидальная функция вырождается в пороговую. Если пороговая функция может принимать только значения 0 и 1, то сигмоидальная функция принимает бесконечное множество значений в диапазоне от 0 до 1. При этом следует заметить, что сигмоидальная функция является дифференцируемой, в то время как пороговая — нет.

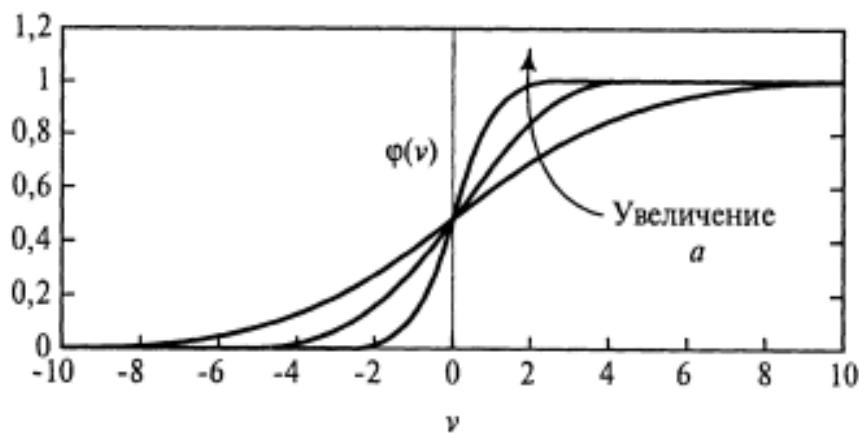


Рисунок 1.4. Сигмоидальная функция для различных значений параметра α

Область значений функций активации, определенных формулами (1.3), (1.4) и (1.5) представляет собой отрезок от 0 до +1. Однако иногда требуется функция активации, имеющая область значений от -1 до +1. В этом случае функция активации должна быть симметричной относительно начала координат. Это значит, что функция активации является нечетной функцией индуцированного локального поля.

1.2 Архитектуры искусственных нейронных сетей

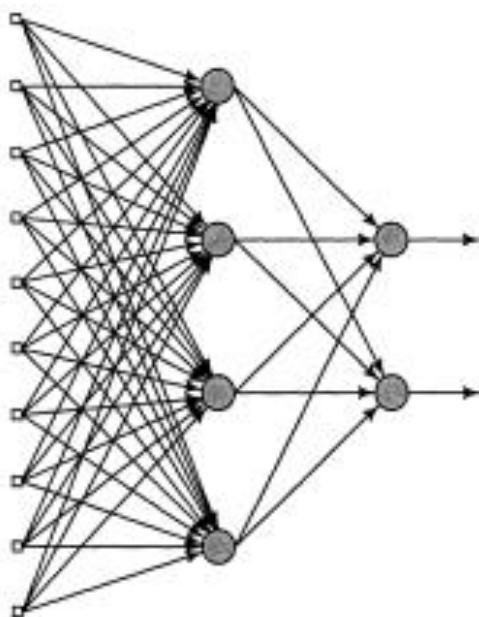
Структура нейронных сетей тесно связана с используемыми алгоритмами обучения. В общем случае можно выделить три фундаментальных класса нейросетевых архитектур[3].

а) Однослойные сети прямого распространения.

б) Многослойные сети прямого распространения. Другой класс нейронных сетей прямого распространения характеризуется наличием одного или нескольких скрытых слоев, узлы которых называются скрытыми нейронами, или скрытыми элементами. Функция последних заключается в посредничестве между внешним входным сигналом и выходом нейронной сети. Добавляя один или несколько скрытых слоев, мы можем выделить статистики высокого порядка.

Сеть, показанная на рисунке 1.5, называется сетью 10-4-2, так как она имеет 10 входных, 4 скрытых и 2 выходных нейрона. Такая сеть считается полносвязной в том смысле, что все узлы каждого конкретного слоя соединены со всеми узлами смежных слоев. Если некоторые из синаптических связей отсутствуют, такая сеть называется неполносвязной.

в) Рекуррентные сети. Рекуррентная нейронная сеть отличается от сети прямого распространения наличием, по крайней мере, одной обратной связи. Наличие обратных связей в сети, показанной на рисунке 1.6, оказывает непосредственное влияние на способность таких сетей к обучению и на их производительность.



Входной слой Слой скрытых нейронов Выходной слой нейронов

Рисунок 1.5 - Полносвязная сеть прямого распространения

Более того, обратная связь подразумевает использование элементов единичной задержки (они обозначены как z^{-1}), что приводит к нелинейному динамическому поведению, если, конечно, в сети содержатся нелинейные нейроны.

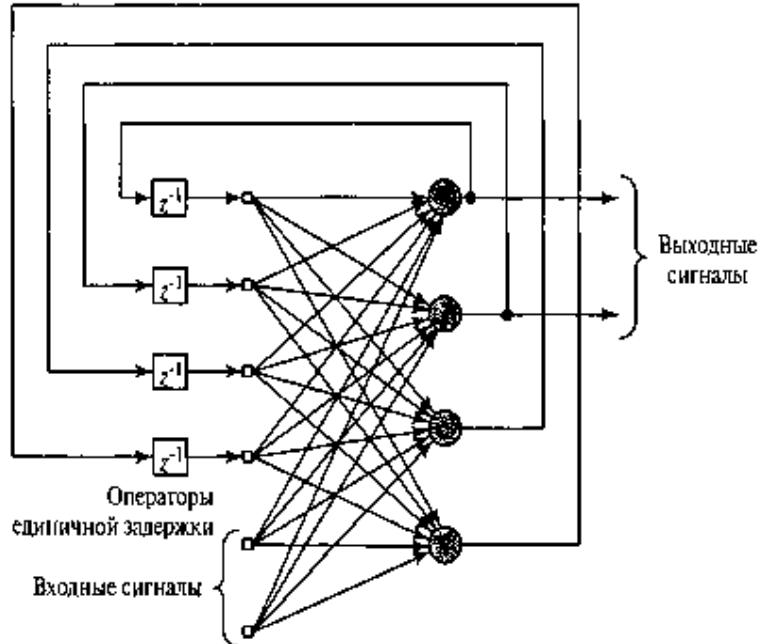


Рисунок 1.6 - Рекуррентная сеть со скрытыми нейронами

1.3 Парадигмы обучения искусственных нейронных сетей

Выделяют две основных парадигмы обучения нейронных сетей [2].

- Обучение с учителем.
- Обучение без учителя.

На рисунке 1.7 показана блочная диаграмма, иллюстрирующая эту форму обучения. Концептуально участие учителя можно рассматривать как наличие знаний об окружающей среде, представленных в виде пар вход-выход. При этом сама среда неизвестна обучаемой нейронной сети.

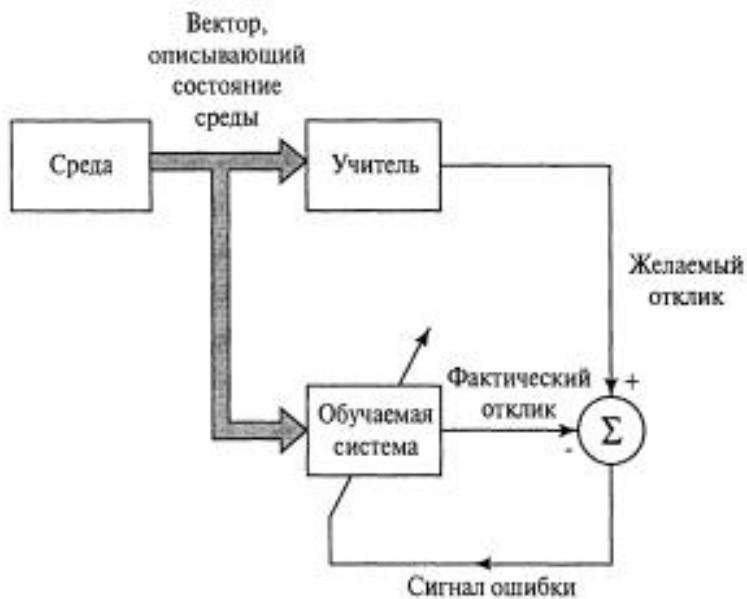


Рисунок 1.7 - Блочная диаграмма обучения с учителем

На основе встроенных знаний учитель может сформировать и передать обучаемой нейронной сети желаемый отклик, соответствующий данному входному вектору. Этот желаемый результат представляет собой оптимальные действия, которые должна выполнить нейронная сеть. Параметры сети корректируются с учетом обучающего вектора и сигнала ошибки. Сигнал ошибки — это разность между желаемым сигналом и текущим откликом нейронной сети. Корректировка параметров выполняется пошагово с целью имитации нейронной сетью поведения учителя. После окончания обучения учителя можно отключить и позволить нейронной сети работать со средой самостоятельно.

Парадигма обучения без учителя (рисунок 1.8) самим названием подчеркивает отсутствие руководителя, контролирующего процесс настройки весовых коэффициентов. При использовании такого подхода не существует маркированных примеров, по которым проводится обучение сети.

Существует лишь независимая от задачи мера качества представления,

которому должна научиться нейронная сеть, и свободные параметры сети оптимизируются по отношению к этой мере. После обучения сети на статистические закономерности входного сигнала она способна формировать внутреннее представление кодируемых признаков входных данных и, таким образом, автоматически создавать новые классы.

Для обучения без учителя можно воспользоваться правилом конкурентного обучения. Например, можно использовать нейронную сеть, состоящую из двух слоев — входного и выходного. Входной слой получает доступные данные. Выходной слой состоит из нейронов, конкурирующих друг с другом за право отклика на признаки, содержащиеся во входных данных.



Рисунок 1.8 - Блочная диаграмма обучения без учителя

В простейшем случае нейронная сеть действует по принципу "победитель получает все". При такой стратегии нейрон с наибольшим суммарным входным сигналом "побеждает" в соревновании и переходит в активное состояние. При этом все остальные нейроны отключаются.

1.4 Подбор оптимальной архитектуры сети

Одно из важнейших свойств нейронной сети - это способность к обобщению полученных знаний. Сеть, натренированная на некотором множестве обучающих выборок, генерирует ожидаемые результаты при подаче на ее вход данных относящихся к тому же множеству, но не участвовавших непосредственно в процессе обучения. Разделение данных на обучающее и тестовое

подмножества представлено на рисунке 1.9.

Множество данных, на котором считается истинным некоторое правило R , разбито на подмножества L , и G , при этом в составе L , в свою очередь, можно выделить определенное подмножество контрольных данных V , используемых для верификации степени обучения сети. Обучение проводится на данных, составляющих подмножество L . Способность отображения сетью элементов L может считаться показателем степени накопления обучающих данных, тогда как способность распознавания данных, входящих во множество C и не использованных для обучения, характеризует ее возможности обобщения знаний. Данные, входящие и в L , и в G , должны быть типичными элементами множества R . В обучающем подмножестве не должно быть уникальных данных, свойства которых отличаются от типичных значений.

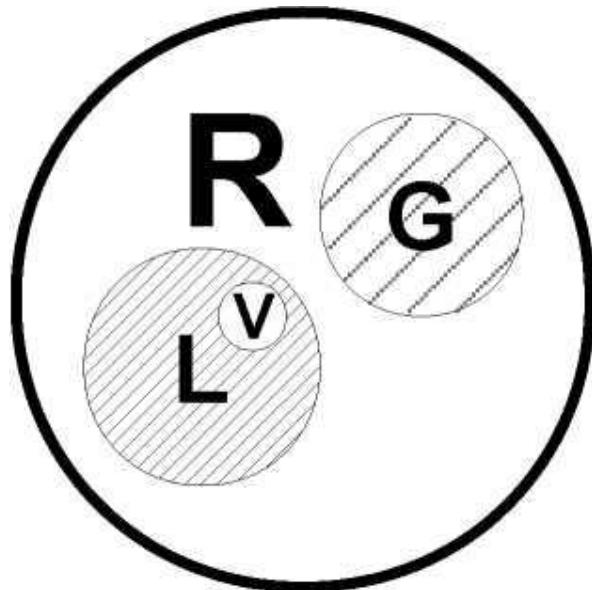


Рисунок 1.9 - Иллюстрация разделения данных, подчиняющихся правилу R , на обучающее подмножество L , тестовое подмножество G и контрольное подмножество V

В [1] приведена мера Вапника-Червонекиса, которая отражает уровень сложности нейронной сети. В качестве ее приближенного значения используется общее количество весов нейронной сети. Рекомендуется, для получения

высоких показателей обобщения подбирать архитектуру сети так, чтобы, количество обучающих выборок превосходило меру Вапника-Червонекиса.

1.5 Сравнительный анализ различных алгоритмов обучения искусственных нейронных сетей

Эффективность алгоритмов обучения проверяется на определенных тестах, соответствующих принятым мировым стандартам. Такими стандартными тестами, в частности, считаются задача логистики, задача парности, кодирование и декодирование двоичных данных, аппроксимация определенного вида нелинейной функции, задача двух спиралей и многие другие. Различные алгоритмы сравниваются по количеству циклов обучения, количеству расчетов значения целевой функции, количеству знакопеременных произведений, чувствительности к локальным минимумам и т.п.

В [1] представлены результаты (таблица 1.1), полученные на компьютере Macintosh Powerbook 1400 при использовании прикладного пакета "Neural Networks" программы Matlab, позволяющие сравнить длительность, количество циклов обучения и вычислительную сложность различных алгоритмов. В ходе экспериментов обучался многослойный персепtron со структурой 1-10-1, предназначенный для аппроксимации 41 пары обучающих одномерных данных. Все алгоритмы были реализованы в инструментальной среде одной и той же программы Matlab, что создало основу для получения объективных оценок.

Таблица 1.1 - Сравнение эффективности алгоритмов обучения

Алгоритм	Время, (с)	Количество циклов	Количество операций, × 106
Наискорейшего спуска с адаптируемым шагом	57,71	980	2,50
Сопряженных градиентов	19,16	89	0,75
Переменной метрики BFGS	10,86	44	1,02
Левенберга-Марквардта	1,87	6	0,46
RPROP	12,96	185	0,56

Получены усредненные результаты по 20 процессам обучения. На малой сети, использованной в ходе тестирования, наибольшую эффективность продемонстрировал алгоритм Левенберга-Марквардта[4] (наименьшее время обучения, наименьшее количество циклов обучения, наименьшая вычислительная сложность). Следующими по количеству циклов и времени обучения идут алгоритмы переменной метрики BFGS и сопряженных градиентов. Самую низкую эффективность в ходе тестирования показал алгоритм наискорейшего спуска (все показатели имеют наихудшие значения). Эвристический алгоритм RPROP в этом соревновании выглядел совсем неплохо - он занял второе место по вычислительной сложности.

По результатам многочисленных и различных тестов сделан общий вывод, что методы переменной метрики и Левенберга-Марквардта, по эффективности доминируют как над методами наискорейшего спуска, так и над методом сопряженных градиентов. Однако это очевидное превосходство исчезает при значительном увеличении размеров сети. Уже при 1000 взвешенных связей наиболее эффективным становится, как правило, метод сопряженных градиентов.

ГЛАВА 2. ОБЛАСТИ ПРИМЕНЕНИЯ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

Однонаправленные нейронные сети с сигмоидальной функцией активации широко применяются на практике, составляя важное звено процесса выработки решений. Далее приводится несколько приложений, позволяющим подчеркнуть универсальность и разнородность функций, которые искусственные нейронные сети могут выполнять.

2.1 Распознавание и классификация образов

Распознаванием и классификацией образа будем называть его идентификацию и отнесение к соответствующему классу данных. При решении этой задачи нейронная сеть может выполнять функцию как экстрактора свойств, так и классификатора, приписывающего образ конкретному классу. Однако чаще всего экстракция свойств производится на отдельном этапе предварительного преобразования данных. В качестве примера рассмотрим определение достоверности цифровой подписи.

Определение достоверности цифровой подписи - это последовательность действий, в результате которых дается ответ на вопрос: принадлежит данная подпись конкретному человеку или нет. В примере использован пакет STATISTICA Neural Networks.

Упростим задачу - рассмотрим одиночный символ, рукописная буква “А”. Как будет видно ниже, сложность данной задачи - в структуре данных и их представления для обучения нейронной сети.

В качестве исходных данных берётся набор картинок. Все картинки разделены на две группы:

- 1) подпись, принадлежащая конкретному человеку;
- 2) подпись, не принадлежащая конкретному человеку (искаженная подпись).

Внутри каждой группы картинки различны. Действительно, ни один человек не может расписаться два раза абсолютно одинаковым образом: изменяются стартовая точка, наклон, форма отдельных элементов. Также присутствуют изображения с “шумами”.

Приведём несколько примеров изображений для каждой группы.



Рисунок 2.1 - "Хорошие" изображения (первая группа)



Рисунок 2.2 - "Плохие" изображения (вторая группа)

Все картинки имеют разрешение 64x64 точек и глубину цвета 1 бит.

Изображения представим в одной двумерной таблице. Для этого применим простую развёртку: двумерный массив значений преобразуется в одномерный. Каждому изображению соответствует одно наблюдение - строка в таблице данных; элементы строки - значения соответствующих пикселей в исходном изображении.

В итоге имеем таблицу с 4096 столбцами ($4096 = 64 \times 64$) и 2275 наблюдениями.

Была добавлена переменная Type, принимающая значение 1, если подпись правильная, 0 - если подпись неправильная.

	Var15/5	Var15/6	Var15/7	Var15/8	Var15/9	Var15/10	Var15/11	Var15/12	Var15/13	Var15/14	Var15/15	Var15/16	Var15/17
1	1	1	1	1	1	1	1	1	1	1	0	0	1
2	1	0	1	1	1	1	1	1	1	1	1	1	0
3	1	1	1	1	1	1	1	1	0	0	1	1	1
4	0	1	0	0	0	0	0	0	0	0	1	1	1
5	0	1	1	1	1	1	1	1	1	1	1	0	0
6	0	1	1	1	1	1	1	1	1	1	1	1	0
7	1	1	1	1	1	1	1	0	0	0	1	1	1
8	0	0	0	0	0	0	0	0	0	0	0	1	1
9	0	0	1	1	0	0	0	0	0	0	0	0	0
10	1	1	1	1	1	1	1	1	0	0	0	1	1
11	1	1	1	1	1	1	1	1	0	0	0	1	1
12	1	1	1	1	1	1	1	1	1	0	0	0	1
13	0	1	1	1	1	1	1	1	1	1	0	0	1
14	0	1	1	1	1	1	1	1	1	1	0	0	1
15	1	0	0	1	1	1	1	1	1	1	1	1	1
16	1	1	1	1	1	1	1	0	0	0	1	1	1
17	1	1	1	1	1	0	0	0	0	0	1	1	1
18	0	0	0	0	0	1	1	1	0	0	1	0	0

Рисунок 2.3 - Исходные данные для классификации

По данным представленным на рисунке 2.3 обучается нейронная сеть, которая после обучения будет выступать в качестве классификатора подписей.

2.2 Построение модели поведенческого скоринга

В примере использован пакет STATISTICA Neural Networks. В данном примере рассмотрим схему построения нейросетевой модели для задачи поведенческого скоринга. Поведенческий скоринг (behavior scoring) используется для принятия решений по уже выданным кредитам.

Основные решения, принимаемые с использованием поведенческого скоринга, можно сформулировать следующим образом:

- Предложение новых услуг и улучшение уже предоставляемых услуг.
- Решение, выдавать ли кредитную карту заново после истечения срока действия, или нет.
- Меньший стартовый кредитный лимит или максимальное значение кредита на кредитной карточке.
- Более строгий сбор платежей с нарушителей или отправка данных о них в агентства сбора платежей.
- Повышение кредитного лимита.
- Помещение под наблюдение ввиду потенциальных мошеннических

действий и т.д.

В данном примере необходимо оценить кредитоспособность существующих заёмщиков на основании данных о графике погашения кредитов и динамики движения средств на счетах клиента.

Каждого клиента будем характеризовать 22 признаками. 20 переменных относятся к анкетным данным, которые заполняются в анкете для получения кредита. К этим переменным относятся:

- Текущий баланс счета;
- Продолжительность в мес.;
- Назначение кредита;
- Сумма кредита;
- Объем сбережений;
- Время работы на данном рабочем месте;
- Семейное положение/пол;
- Длительность проживания по текущему адресу;
- Возраст в годах;
- Число предыдущих кредитов в банке;
- Должность и другие.

На основании перечисленных факторов все клиенты подразделяются на “хороших” и “плохих”. Разбиение на эти группы записано в переменной Кредитоспособность (Creditability).

Всего имеются данные по 1000 клиентов. При этом, 30% относятся к “плохим”, а остальные 70% - к “хорошим”. Процент невыплат по всей совокупности данных около 3% (данная величина относится к одному месяцу). Элемент таблицы данных показан на рисунке 2.4.

	1 Кредито- способность	2 Текущий баланс счета	3 Продолжи- тельность в мес	4 Выплаты по предыдущим кредитам	5 Назначение кредита сумма кредита
1	плохой	нет текущего счета	36	нет проблем	переподготовка
2	хороший	нулевой баланс	48	неуверенное	переподготовка
3	плохой	>=200	36	не было кредитов	поддержанная м.
4	хороший	нет текущего счета	24	выплачены	новая машина
5	хороший	>=200	24	не было кредитов	переподготовка
6	хороший	нулевой баланс	12	не было кредитов	переподготовка
7	плохой	нет текущего счета	30	не было кредитов	поддержанная м.
8	хороший	нулевой баланс	15	выплачены	предметы мебе.
9	хороший	>=200	15	выплачены	предметы мебе.
10	плохой	нулевой баланс	27	выплачены	предметы мебе.
11	плохой	нулевой баланс	24	не было кредитов	поддержанная м.
12	плохой	нет текущего счета	18	не было кредитов	ремонт
13	плохой	нулевой баланс	36	нет проблем	переподготовка
14	хороший	>=200	6	нет проблем	переподготовка
15	хороший	нет текущего счета	12	не было кредитов	другой
16	хороший	>=200	42	не было кредитов	предметы мебе.
17	хороший	нет текущего счета	48	не было кредитов	новая машина
18	плохой	нулевой баланс	24	с проблемами	ремонт
19	хороший	нулевой баланс	24	выплачены	другой
20	хороший	нулевой баланс	48	не было кредитов	бизнес

Рисунок 2.4 - Фрагмент исходной таблицы данных

Поскольку количество наблюдений, относящихся к разным группам (“хороший” и “плохой”), существенно различается, то необходимо задать дополнительную переменную, содержащую веса наблюдений. В противном случае группа “хороших” будет оказывать большее влияние на построение модели, чем группа “плохих”.

Для группы “хороший” зададим вес равный 3, а для группы “плохой” равный 7 (т.е. каждая группа будет оказывать одинаковое влияние на построение модели). Переменную, содержащую веса, назовем w .

Теперь перейдем к построению модели.

Шаг 1. На первом шаге необходимо исключить из анализа переменные, которые не оказывают значимого влияния на принадлежность к тому или иному классу (на зависимую переменную). Использование алгоритмов снижения размерности уменьшает количество независимых переменных до 10.

Шаг 2. В стартовом окне задаем новый набор независимых переменных (рисунок 2.5).

В качестве инструмента построения сети будем использовать *Мастер*

решений. После того, как установки проделаны, нажимаем *OK*.

Шаг 3. Выбор модели.

В начале проведем анализ для невзвешенных переменных. На вкладке *Быстрый Мастера решений* (рисунок 2.6) изменим величину длительности анализа и установим ее равной 100.



Рисунок 2.5 - Стартовое окно модуля Нейронные сети

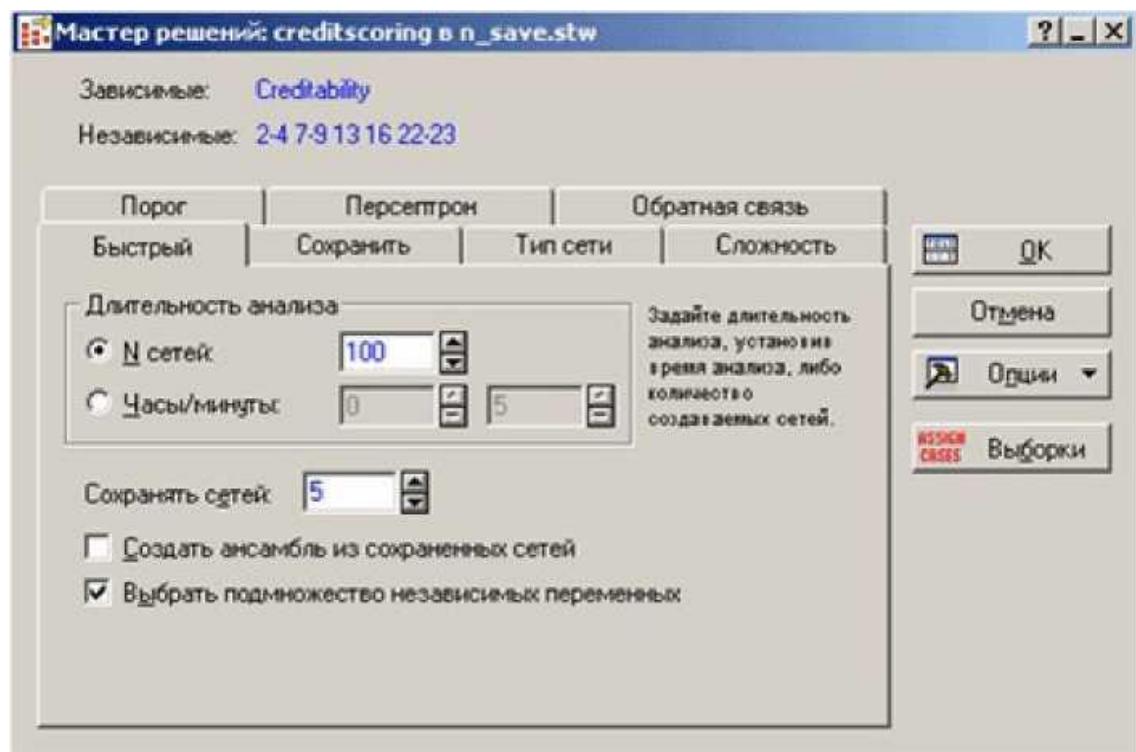


Рисунок 2.6 - Окно Мастера решений, вкладка Быстрый

Для начала, нам необходимо выяснить, какие модели будут работать эффективнее, поэтому, не изменяя остальных опций, нажимаем *OK*.

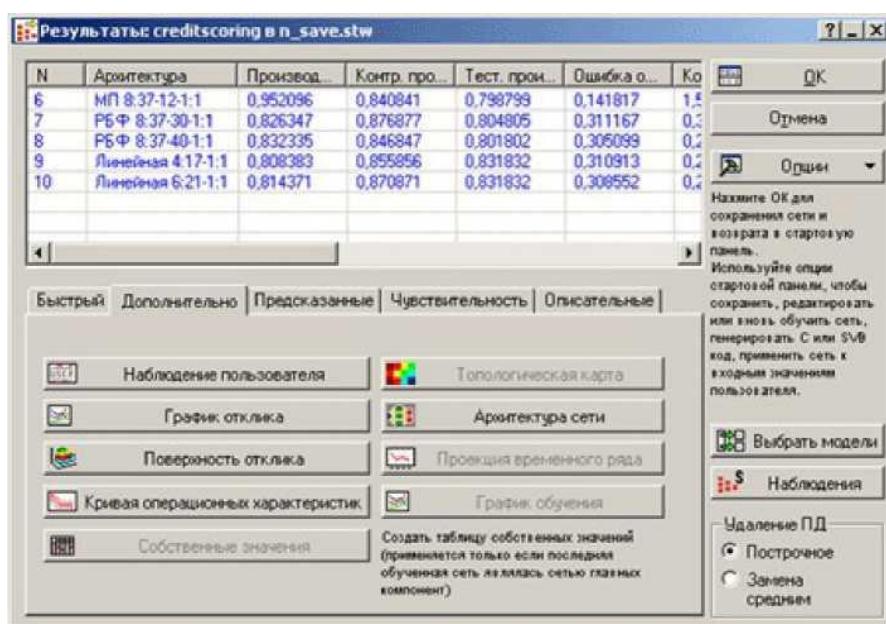


Рисунок 2.7 - Окно результатов

Анализируя величину производительности на тестовом множестве, делаем вывод, что точность прогноза с помощью построенных моделей находится на уровне 80%.

Не углубляясь в дальнейшее изучение построенной модели, попробуем улучшить ее, включив в анализ переменную с весами. Здесь приведём только матрицу классификации.

		Классификация (1)	
		Creditability.bad.1	Creditability.good.1
Всего		2100,000	2100,000
Правильно		2058,000	2031,000
Ошибочно		42,000	69,000
Неизвестно		0,000	0,000
% правильных		98,000	96,714
% ошибочных		2,000	3,286
% неизвестно		0,000	0,000

Рисунок 2.8. Таблица описательных статистик классификации

Процент правильно классифицированных в категорию “плохой” равен 98%, а в категорию “хороший” - 96.7%.

2.3 Другие области использования искусственных нейронных сетей

— Прогнозирование финансовых временных рядов (компания LBS Capital Management объявила о значительных успехах в финансовых операциях, достигнутых за счет прогнозирования цен акций с помощью многослойных персепtronов).

— Повышение эффективности процесса добычи полезных ископаемых (выделение значимых факторов, влияющих на показатели эффективности добычи).

— Оптическое распознавание символов, включая распознавание подписи (например, система идентификации подписи, учитывающая не только

окончательный ее рисунок, но и скорость движения пера на различных участках, что значительно затрудняет подделку чужой подписи).

—Обработка изображений (например, система сканирует видеоизображения станций метро и определяет, насколько станция заполнена людьми, причем работа системы не зависит от условий освещенности и движения поездов).

—Медицинская диагностика (например, прогнозирование эпилептических припадков, определение размеров опухоли простаты).

—Синтез речи (знаменитая экспериментальная система Nettalk, способная произносить фонемы из написанного текста).

—Лингвистический анализ (пример: сеть с неконтролируемым обучением используется для идентификации ключевых фраз и слов в языках туземцев Южной Америки).

2.4. Распознавание образов с помощью нейронных сетей

За последние несколько лет интерес к искусственным нейронным сетям возрастает ввиду возможностей их применения в задачах распознавания образов, задачах управления, прогнозирования и т. д. Алгоритмы обучения построены на основе принципов организации и функционирования биологических нейронных сетей. Мозг человека состоит из очень большого числа *нейронов*, соединенных многочисленными связями. Каждый нейрон обладает множеством свойств, общих с другими органами тела. Однако ему присущи абсолютно уникальные способности: принимать, обрабатывать и передавать электрохимические сигналы по нервным путям, которые образуют коммуникационную систему мозга. Искусственные нейронные сети индуцированы биологией, так как состоят из элементов, функциональные возможности которых аналогичны большинству элементарных функций биологического нейрона [1]

Остановимся подробней на **распознавании образов**. Это научное направление, связанное с разработкой принципов и построением систем,

предназначенных для определения принадлежности объекта к одному из классов объектов. Под объектами в распознавании образов понимают: различные предметы и явления, процессы и ситуации, сигналы и т. п. [2]. Системы распознавания имеют следующую типичную функциональную схему: входные данные, подлежащие распознаванию, подаются на вход системы и подвергаются предобработке с целью их преобразования в необходимый для следующего этапа вид или для выделения из них необходимых характерных признаков. Далее на этапе принятия решения над обработанным массивом данных производится ряд вычислений и на основе их результатов формируется ответ, содержащий ожидаемые от системы сведения о входных данных. Содержание входных и выходных данных определяется назначением системы [3].

В данной работе автор произвел моделирование однослойной и многослойной

нейронной сети в программе Matlab (рис. 1). Инструментарий распознавания образов использовался для распознавания цифр. Он осуществляется с помощью программы «Qt Creator» (рис. 2). В ней вычисляются синхронные операции.

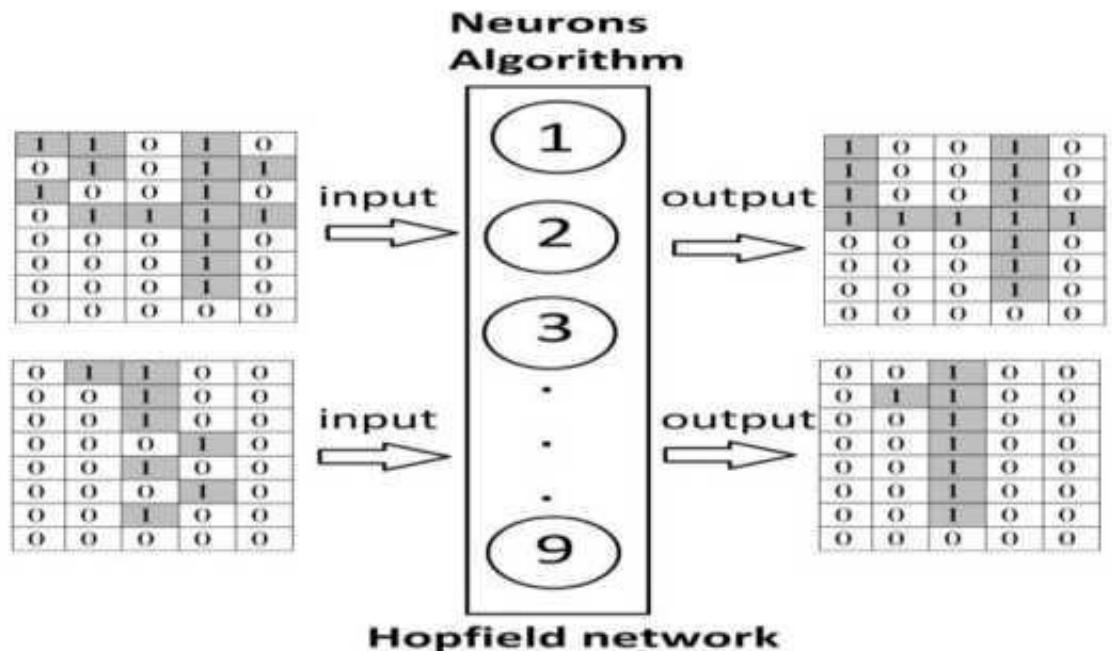


Рис. 1. Цифра, полученная с помощью нейронной сети Хопфилда

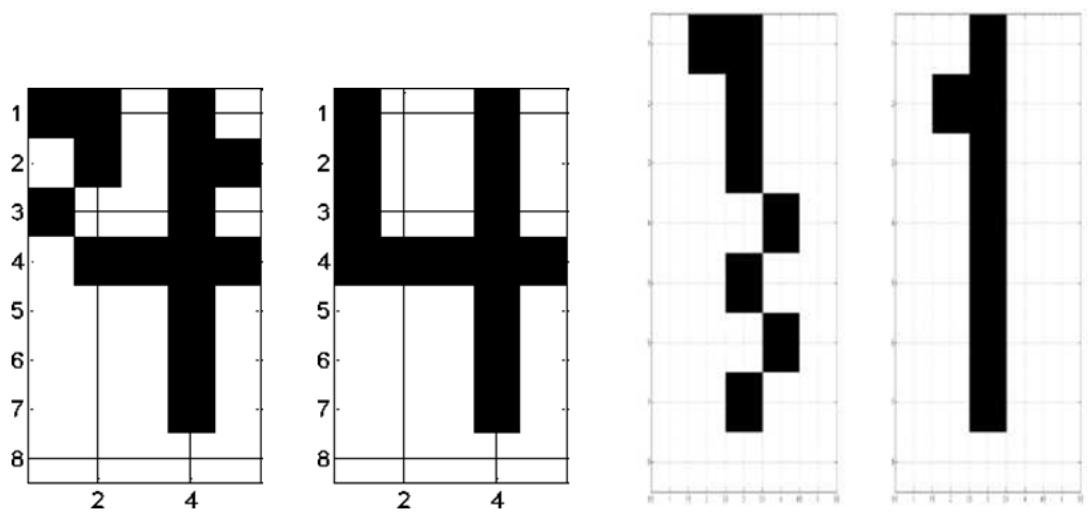


Рис. 2. Полученную с помощью нейронной сети Хопфилда цифру можно

ГЛАВА 3. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ РАБОТЫ С ИСКУССТВЕННЫМИ СЕТЬЯМИ

3.1 Multiple Back-Propagation

Данное ПС является свободно распространяемым приложением для обучения нейронных сетей с помощью алгоритма обратного распространения ошибки. На рисунке 3.1 представлено главное окно программы с созданной моделью нейронной сети прямого распространения, имеющей топологию

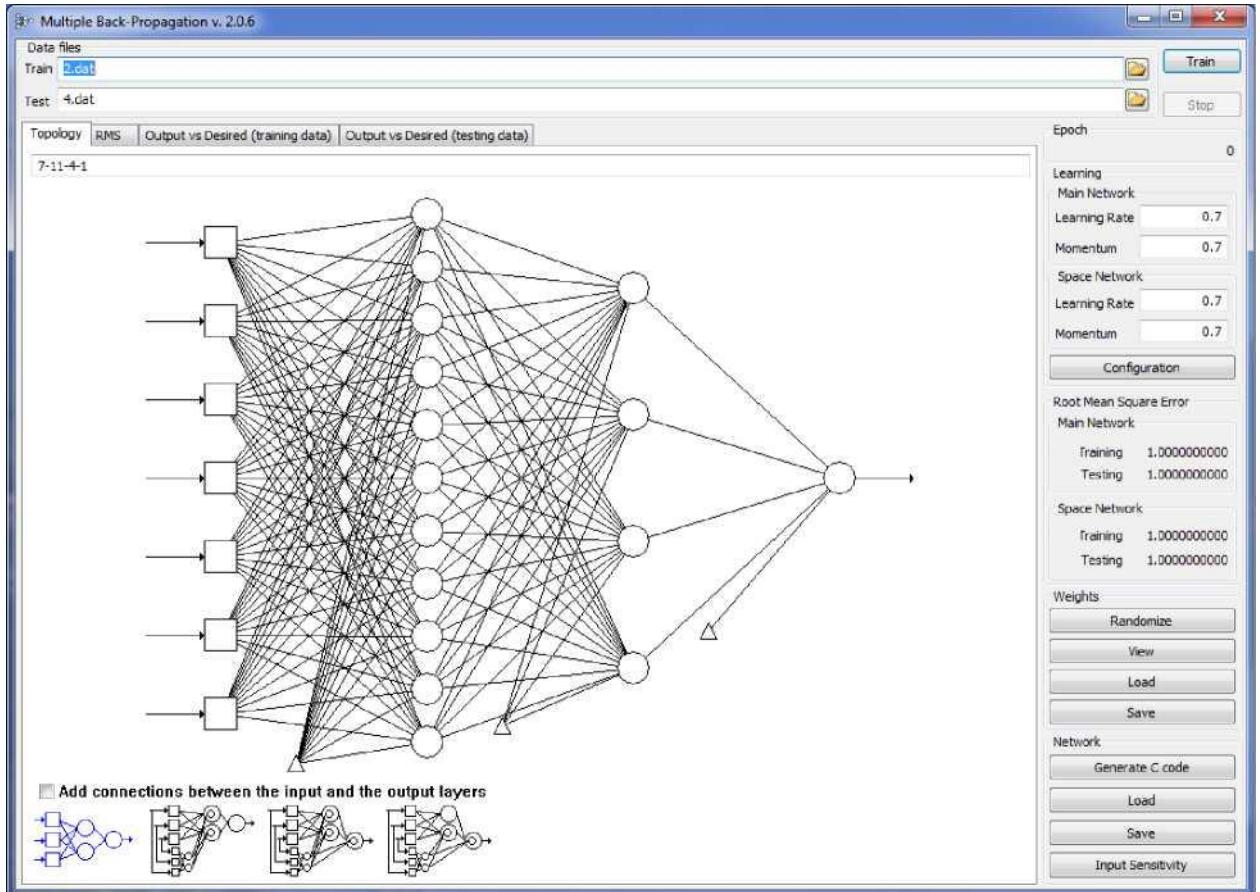


Рисунок 3.1 - Главное окно программы

3.2 Пакет STATISTICA Neural Networks

STATISTICA Neural Networks является богатой, мощной и чрезвычайно быстрой средой анализа нейросетевых моделей, предоставляющей следующие возможности:

- Пре- и пост-обработка, включая выбор данных, кодирование номинальных значений, шкалирование, нормализация, удаление пропущенных

данных с интерпретацией для классификации, регрессия и задачи временных рядов;

- Исключительная простота в использовании плюс непревзойденная аналитическая мощность;
- Мощные аналитические технологии, в том числе Анализ главных компонент и Понижение размерности для выбора нужных входных переменных в нейросетевом анализе данных;
- Самые современные, оптимизированные и мощные алгоритмы обучения сети (включая методы сопряженных градиентов и Левенберга-Марквардта);
- полный контроль над всеми параметрами, влияющими на качество сети, такими как функции активации и ошибок, сложность сети;
- Поддержка каскадов нейросетей и нейросетевых архитектур практически неограниченного размера, созданных в Наборах сетей - Network Sets; выборочное обучение нейросетевых сегментов; объединение, и сохранение наборов сетей в отдельных файлах;
- Полная интеграция с системой STATISTICA; все результаты, графики, отчеты и т.д. могут быть в дальнейшем модифицированы с помощью мощных графических и аналитических инструментов STATISTICA (например, для проведения анализа предсказанных остатков, создания подробного отчета и т.п.);
- Полная интеграция с мощными автоматическими инструментами STATISTICA; запись полноценных макросов для любых анализов; создание собственных нейро-сетевых анализов и приложений с помощью STATISTICA Visual Basic, вызов STATISTICA Neural Networks из любого приложения, поддерживающего технологию СОМ (например, автоматическое проведение нейросетевого анализа в таблице MS Excel или объединение нескольких пользовательских приложений, написанных на C++, C#, Java и т.д.).

3.3 Matlab Neural Network Toolbox

Neural Network Toolbox пакет для работы с нейронными сетями представляет собой полноценную среду MATLAB для решения прикладных задач. Пакет обеспечивает всестороннюю поддержку проектирования, обучения и моделирования множества известных сетевых парадигм, от базовых моделей персептрона до самых современных ассоциативных и самоорганизующихся сетей. Пакет может быть использован для исследования и применения нейронных сетей к таким задачам, как обработка сигналов, нелинейное управление и финансовое моделирование.

Основные свойства:

- Неуправляемые сети: Хэбб, Кохан, конкурентные, карты признаков и самоорганизующиеся карты.
- Неограниченное число элементов и взаимосвязей.
- Настраиваемые на пользователя архитектуры и функции активации.

Пакет Neural Network предоставляет доступ к полному набору средств для исследования, проектирования и моделирования нейронных сетей.

Средства анализа и моделирования MATLAB и SIMULINK позволяют быстро оценивать поведение сети и ее качество в смысле окончательного результата проектирования. С помощью Real-Time Workshop можно генерировать С код для автономных приложений и встроенных систем. Гибкий импорт данных и функции преобразования упрощают первичную подготовку входных данных.

3.4. Распознавание образов с помощью нейронных сетей в среде matlabr2009b

Искусственные нейронные сети представляют собой математическую модель функционирования биологических нейронных сетей - сетей нервных клеток живого организма. Как и в биологической нейронной сети, основным элементом искусственной нейронной сети является нейрон. Соединенные между собой нейроны, образуют слои, количество которых

может варьироваться в зависимости от сложности нейронной сети и решаемых ею задач. Теоретические основы программирования таких нейронных сетей, описываются во многих работах [7, 10, 11].

Одной из актуальных задач является распознавание визуальных образов. Машины, способные распознавать росписи на бумагах, символы на банковских карточках или чеках, существенно облегчают человеческий труд и ускоряют рабочий процесс, при этом снижают риск ошибки за счет отсутствия человеческого фактора [6].

Цель разрабатываемого шаблона, создать нейронную сеть, которая сможет распознавать визуальные образы букв русского алфавита. Программный код, который решает подобную задачу, присутствует в системе Matlab, как демонстрационная программа с названием appcr1. Подробно этот код разобран и пояснен в книге «Нейронные сети» В. С. Медведева, В. Г. Потемкина [6], а так же описан в работе И. С. Миронова, С. В. Скурлаева [7]

В системе Matlab также присутствует инструмент NNtool, имеющий графический интерфейс пользователя, который существенно облегчает задачу и может быть легко использован даже неопытным пользователем. Подробно этот инструмент описан в работах В. Иванников, А. Ланнэ [5] и П. А. Сахнюка [8] и др. [12]. В работе А. И. Шеремет, В. В. Перепелицы, А. М. Денисовой показан пример разработки нейронной сети для распознавания визуальных образов символов латинского алфавита с помощью NNtool [9]. Зарубежные ученые также применяют искусственные нейронные сети в своих исследованиях [13, 14].

Рассмотрим встроенную функцию Matlab prprob, которая представляет собой матрицу, содержащую набор признаков букв латинского языка. Каждая буква имеет размерность 7 на 5 пикселей.

Создадим подобную матрицу с буквами русского алфавита. Для этого создадим в графическом редакторе шаблон каждого символа такой же размерностью (рис. 1, 2).



Рис. 1. Шаблон буквы А созданный в графическом редакторе



Рис. 2. Шаблон буквы Б созданный в графическом редакторе После того, как созданы шаблоны для каждой буквы, необходимо написать функцию, которая будет считывать необходимые признаки символов с графического файла в нужном нам формате.

Для этого выберем в командном меню: File->New^Function M-file. Откроется графический редактор, в который необходимо вставить приведенный ниже код.

Код функции ImgRead:

```
function y = Imgread(x)  
%UNTITLED Summary of this function goes here  
% Detailed explanation goes here  
img=imread(x);  
img1=img(:,:,1);  
img2=reshape (img1,[35, 1]);
```

```
for i=1:35
if
(img2(i,1)==0)
img2(i,1)=1; end
end
for i=1:35 if (img2(i,1)==255) img2(i,1)=0;
end end
y=img2;
end
```

Использую данную функцию, создадим матрицу признаков русского алфавита, введя в окно команд следующий код:

```
images1=Imgread('C:\alphabet\A.png');
```

```
images33=Imgread('C:\alphabet\H.png');
RA=[images1,images2,images3,images4,images5,images6,images7,images8
,... images9,images 10,images 11 ,images12,images 13,images14,images 15,images
16,... images 17,images 18,images 19,images20,images21 ,images22,images23,...
images24,images25,images26,images27,images28,images29,images30,images31,...
images32,images33];
```

Заметим, что для того, чтобы код работал, шаблоны должны располагаться в каталоге C:\alphabet\, в качестве имени сам символ, с разрешением png, или внести соответствующие изменения в код.

Теперь есть матрица RA, которая содержит в себе набор признаков русского алфавита, и будет использоваться в роли входных данных при создании нейронной сети.

В качестве матрицы целей создадим единичную матрицу размерностью 33 на 33.

Для этого введем код. Так же объявим переменные, содержащие в себе количество строк и столбцов.

```
%Создаем переменные для создания нейронной сети в NNtool
```

```
P=double(RA);  
T=eye(33);  
[R,Q] = size(P);  
[S2,Q] = size(T);
```

Для обучения сети нам понадобятся данные с шумом. Создадим эти данные, введя следующий код в окно команд:

```
%Создаем переменные для обучения на зашумленных данных  
нейронной
```

сети в

%NNtool

P1=P;

T1=T; for i=1:100

P1=[P1 ,P+rand(R,Q)*0.1 ,P+rand(R,Q)*0.2];

T1=[T1,T,T];

end

Для симуляции сети нам понадобится переменная, содержащая в себе набор признаков одной буквы, например, буквы И.

Введем соответствующую команду, заодно выведем на экран получившийся зашумленный образ (рис.3).

```
noisy10 = P(:,10) + randn(35,1)*0.2; plotchar(noisy10); %
```

Зашумленный символ И

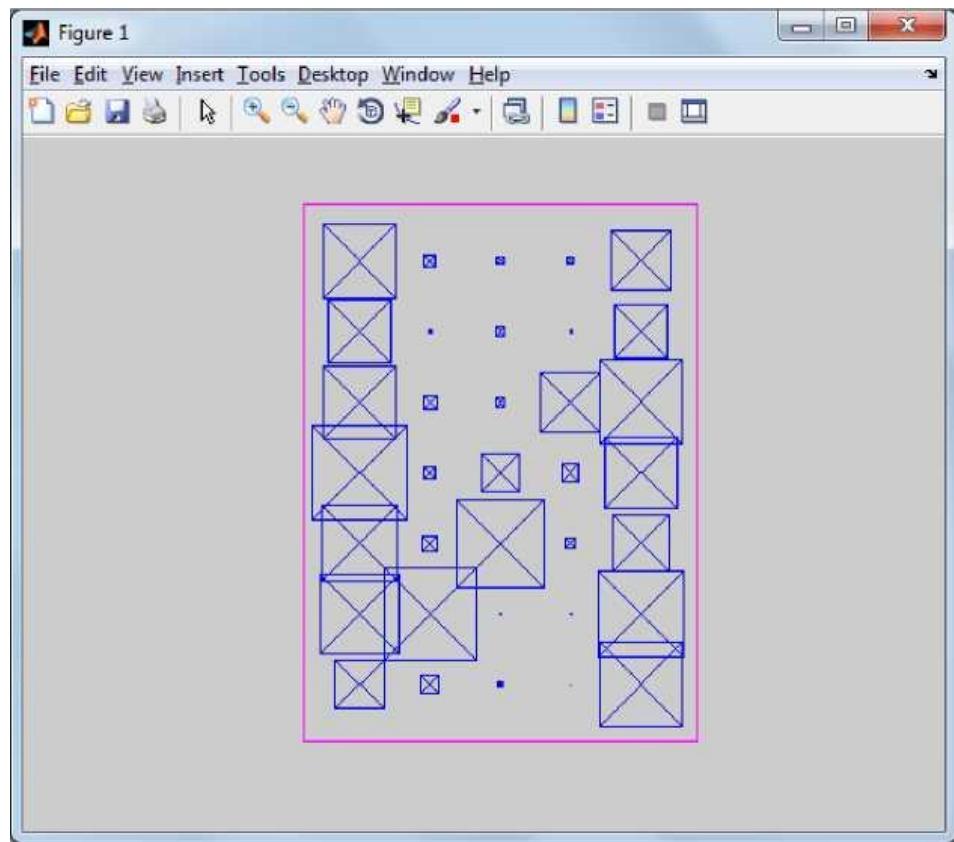


Рис. 3. Зашумленный образ буквы И

Теперь вызовем инструмент NNtool, где с помощью графического интерфейса создадим нейронную сеть. Сделать это можно с помощью соответствующей команды nntool (рис.4).

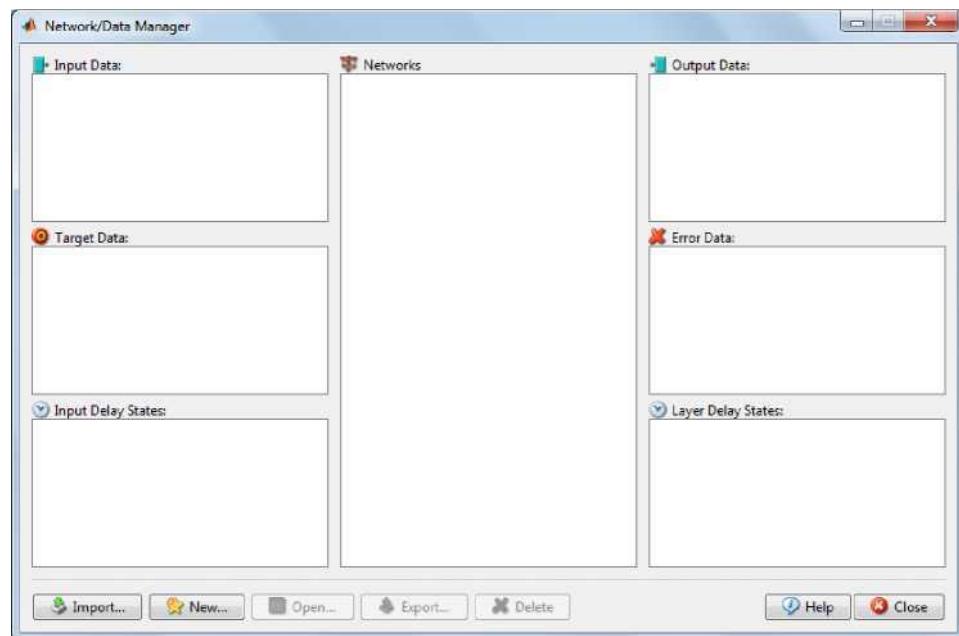


Рис. 4. Окно менеджера данных нейронной сети

С помощью кнопки Import добавляем необходимые нам переменные (рис.5).

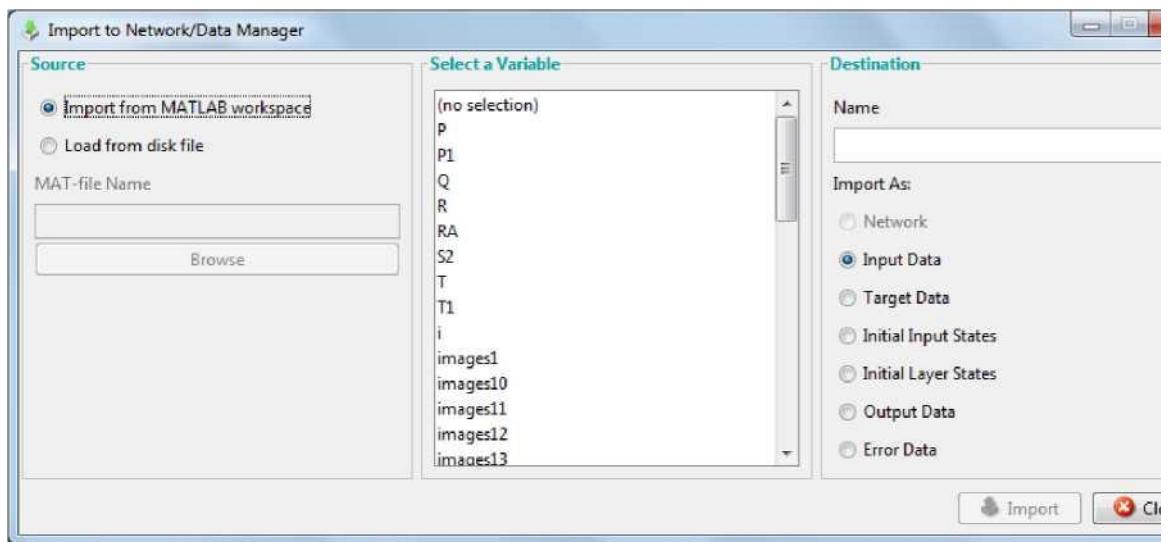


Рис. 5. Окно импорта данных

Переменные P,P1 и noisy10 добавить как input data.

Переменные T,T1 добавить как Target data.

После импорта всех переменных окно должно выглядеть как на рис.6.

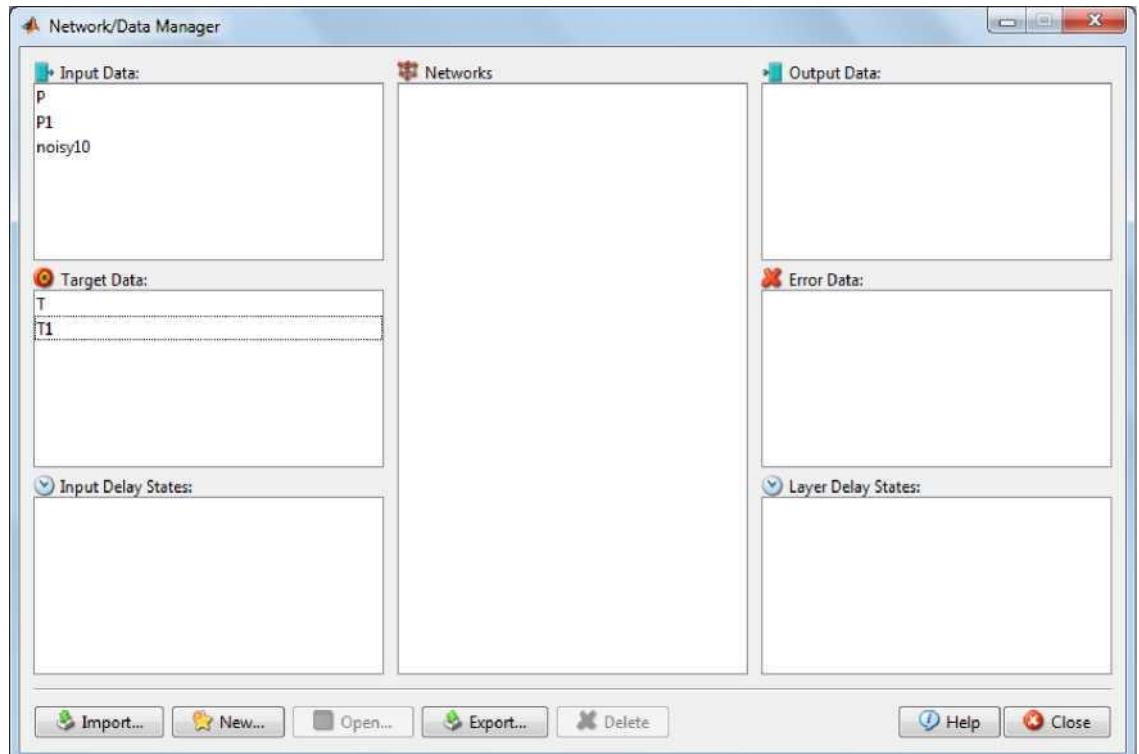


Рис. 6. Окно менеджера данных нейронной сети после импорта переменных

Нажав кнопку New, приступим к созданию нейронной сети. В окне параметров нейронной сети введем настройки (рис.7).

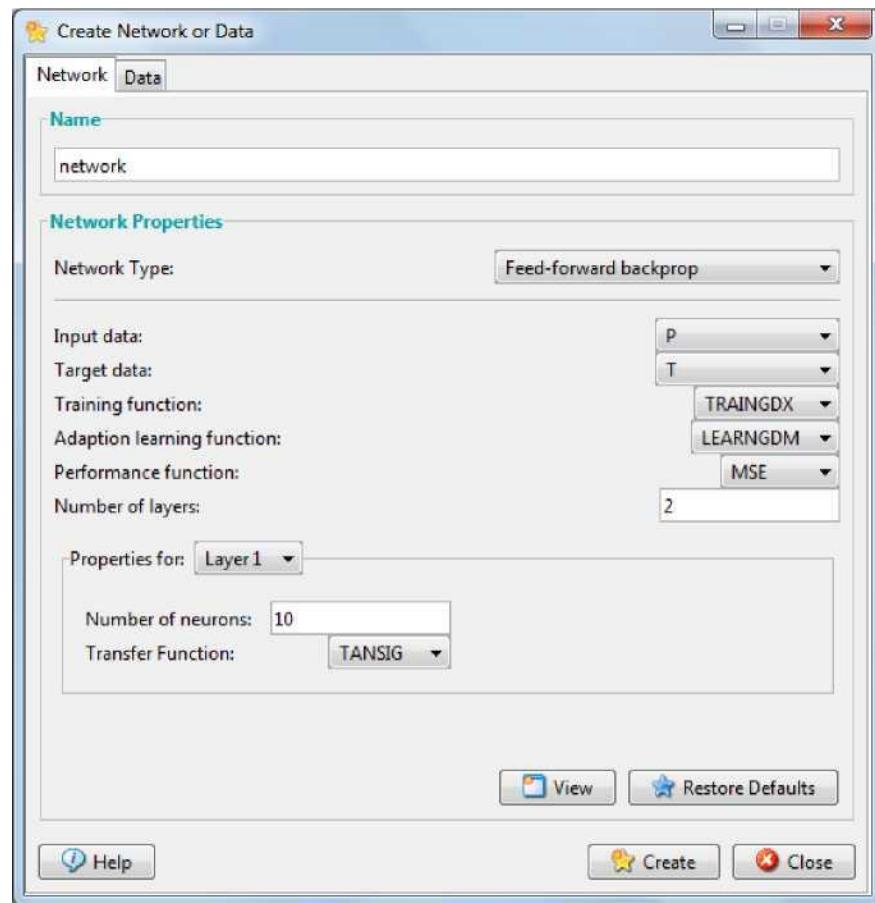


Рис. 7. Окно создания нейронной сети

После выбранных параметров создаем сеть, нажав кнопку Create.

После этого сеть должна появиться в окне менеджера данных нейронной сети в разделе networks (рис.8).

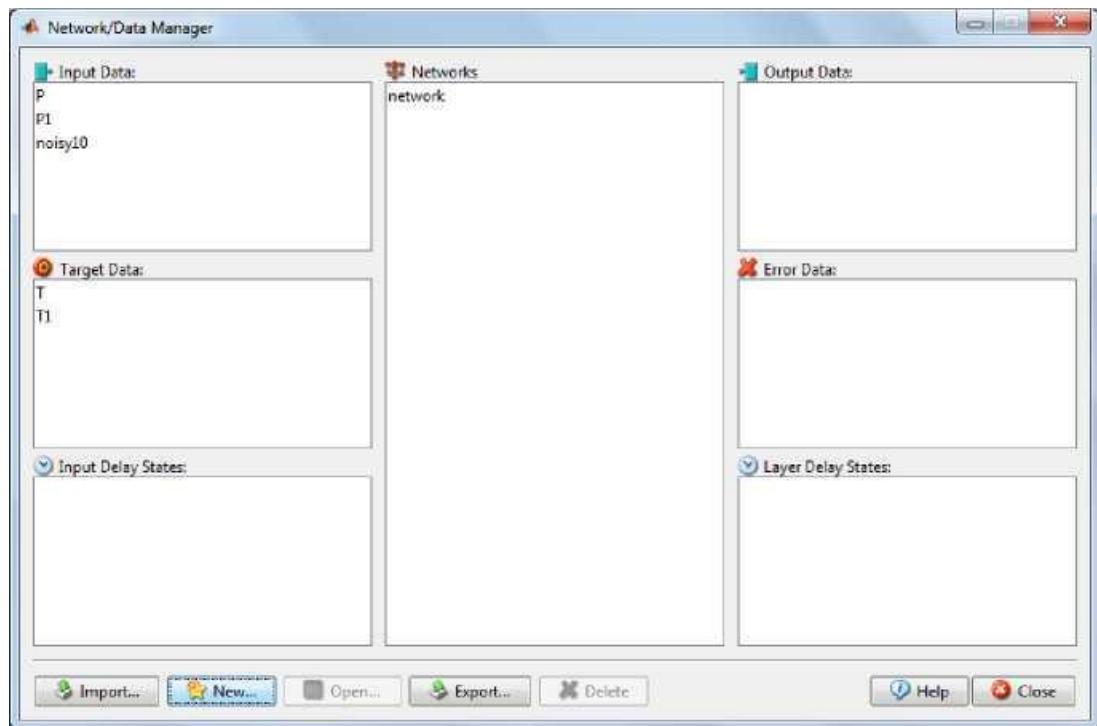


Рис 8. Окно менеджера данных нейронной сети после создания нейронной сети

Нажав дважды на созданную сеть, откроем нейронную сеть. Появится окно с вкладками (рис.9), в которых можно посмотреть структуру нейронной сети, обучить ее, провести симуляцию, изменять веса входных данных.

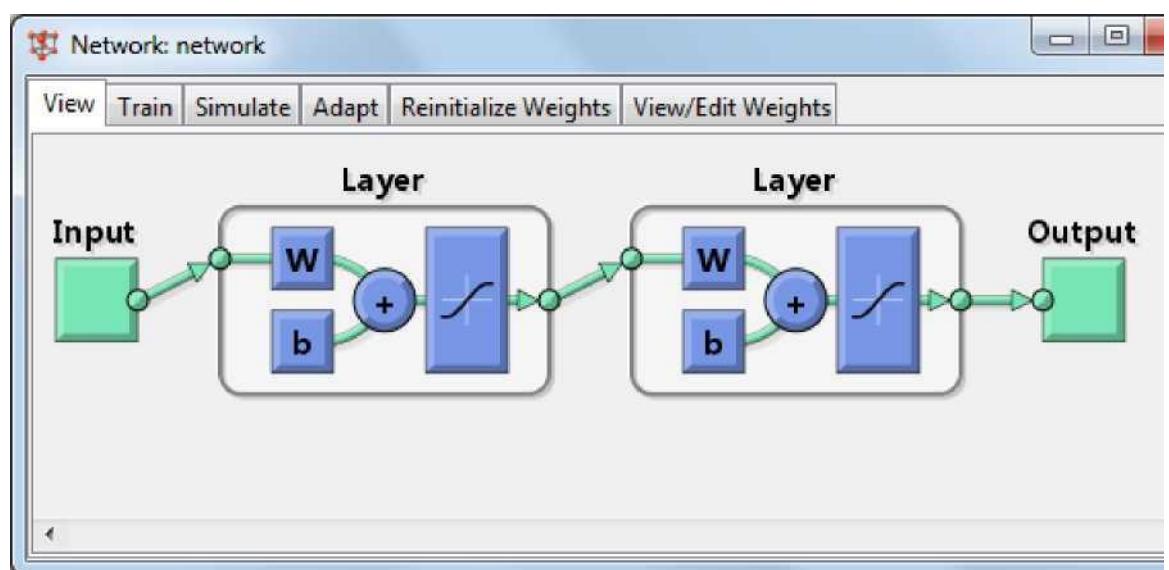


Рис. 9. Структура нейронной сети

Рассмотрим вкладку Train, где будет проводиться обучение нейронной

сети. В начале проведем обучение на идеальных данных. Введем соответствующие параметры (рис.10).

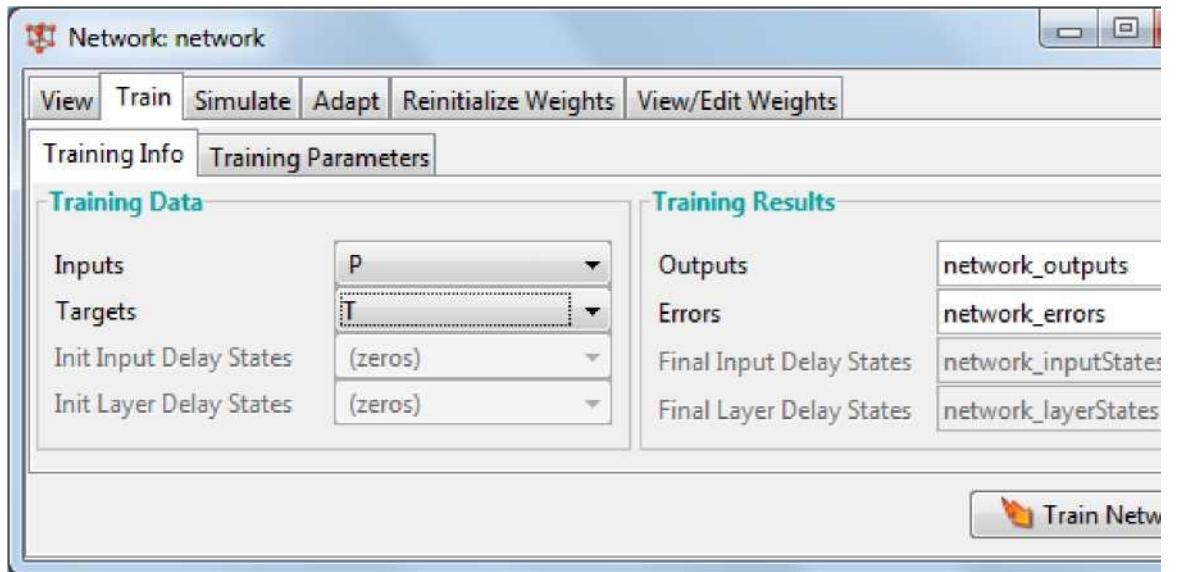


Рис. 10. Настройка данных, с помощью которых будет происходить обучение

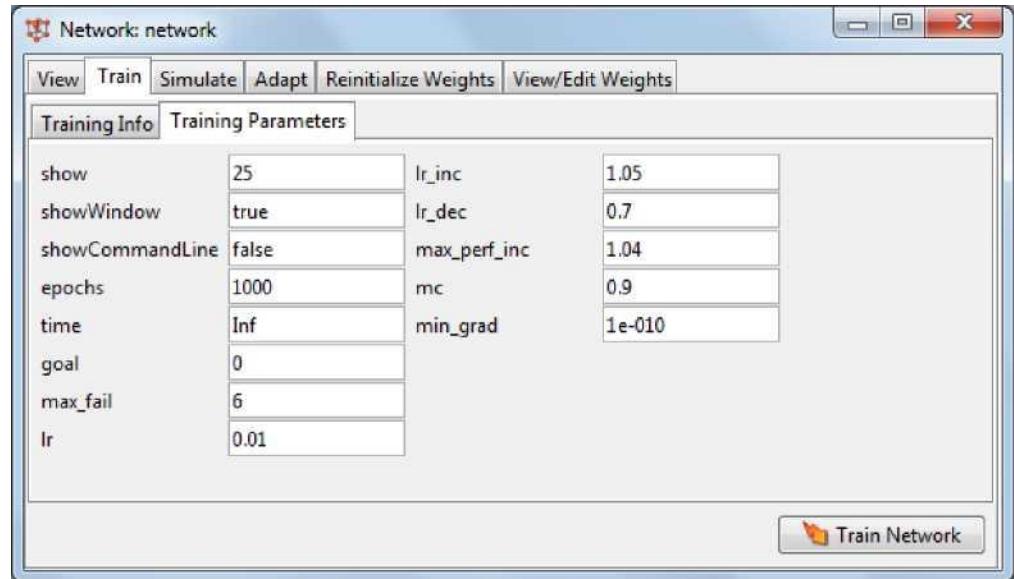


Рис. 11. Параметры обучения нейронной сети

После установки всех параметров запускаем обучение, нажав кнопку Train Network (рис.11). В появившемся окне (рис.12) можем наблюдать процесс обучения нейронной сети.

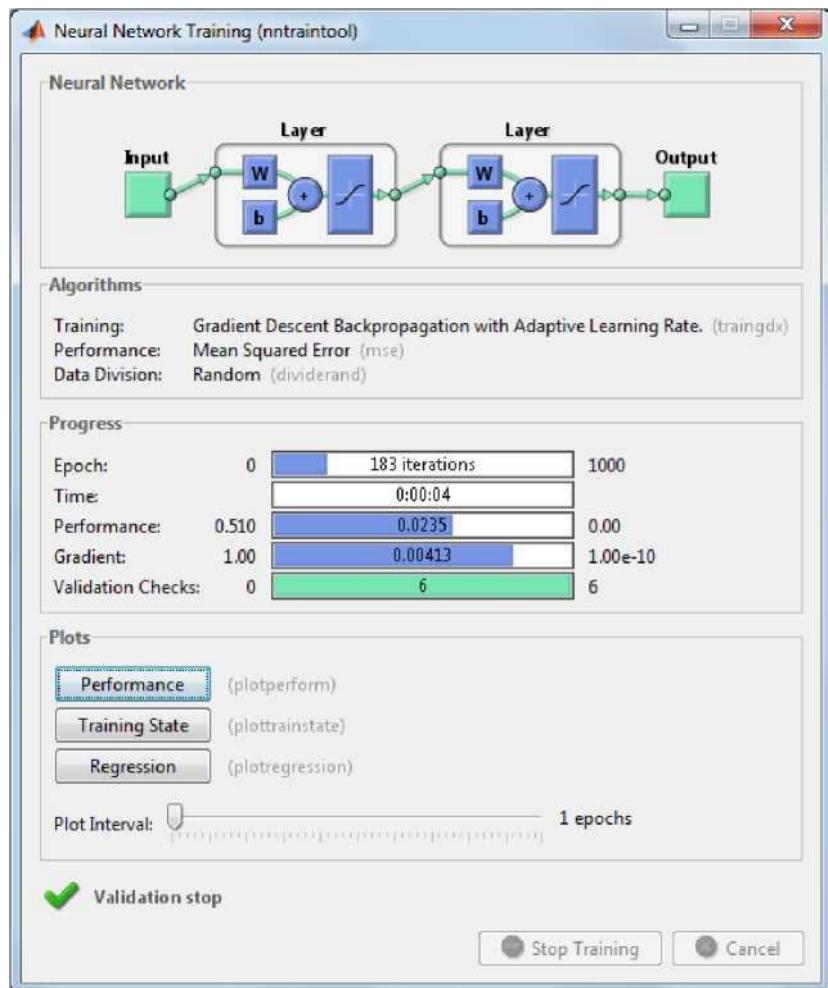


Рис. 12. Процесс обучения

С помощью кнопки Performance можно просмотреть процесс обучения с помощью графика (рис.13).

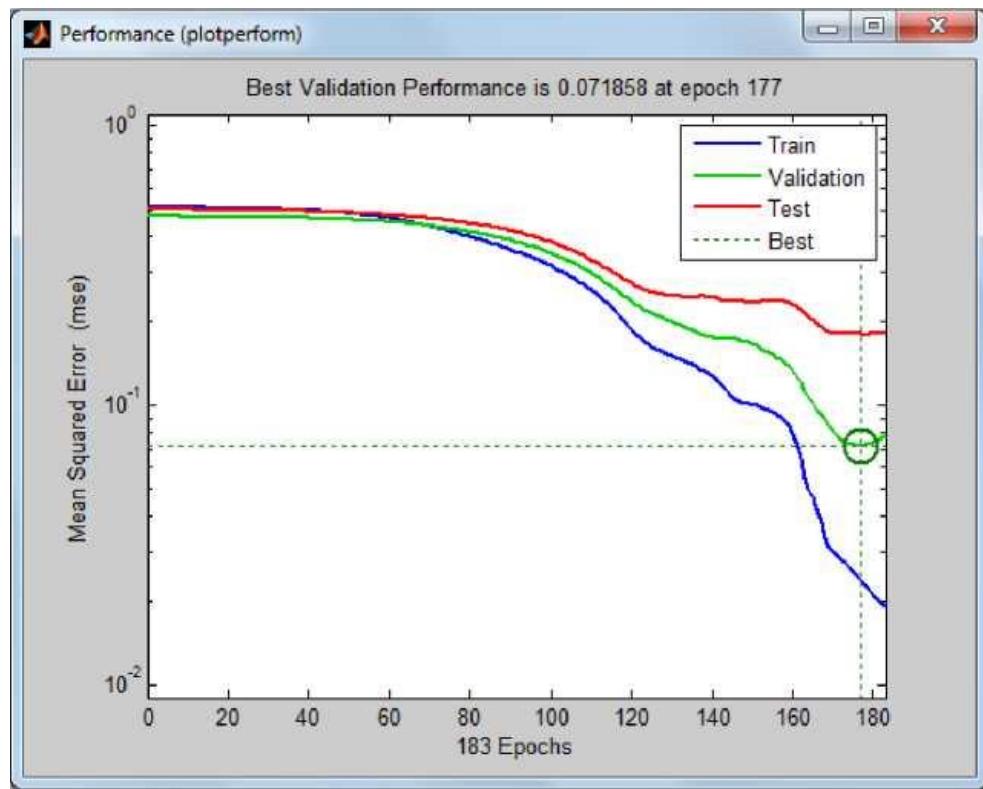


Рис. 13. График обучения нейронной сети

Теперь необходимо провести обучение на данных с шумом. Для этого изменим параметры во вкладке train (рис.14, 15).

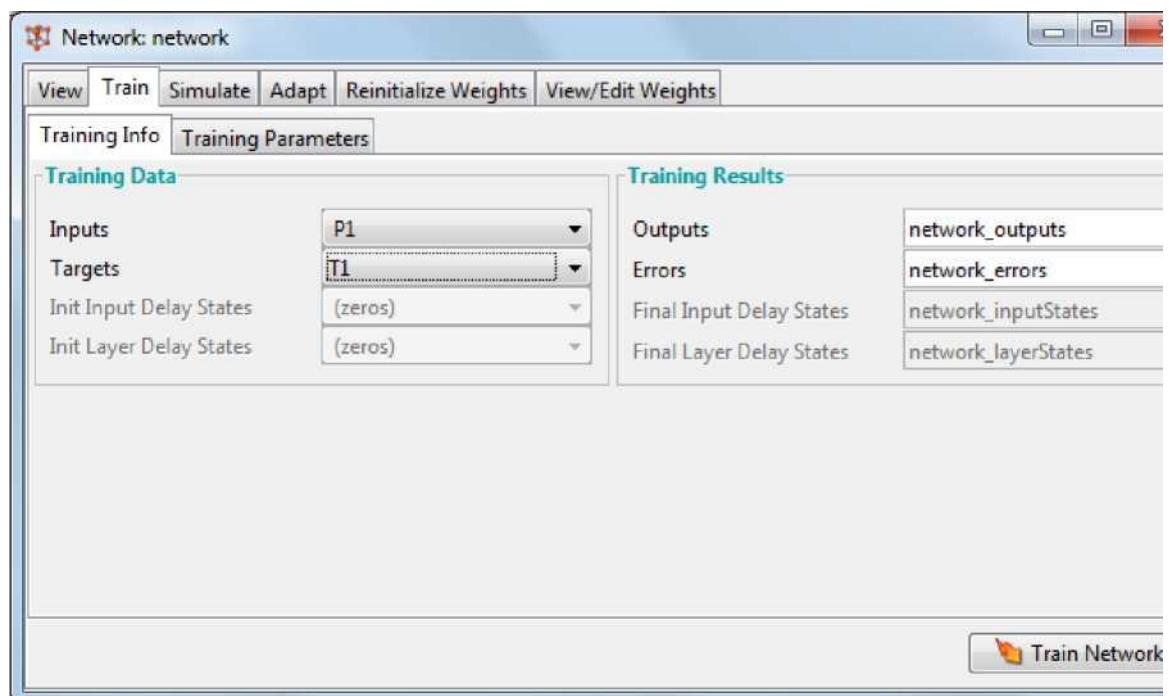


Рис. 14. Параметры обучения на данных с шумом

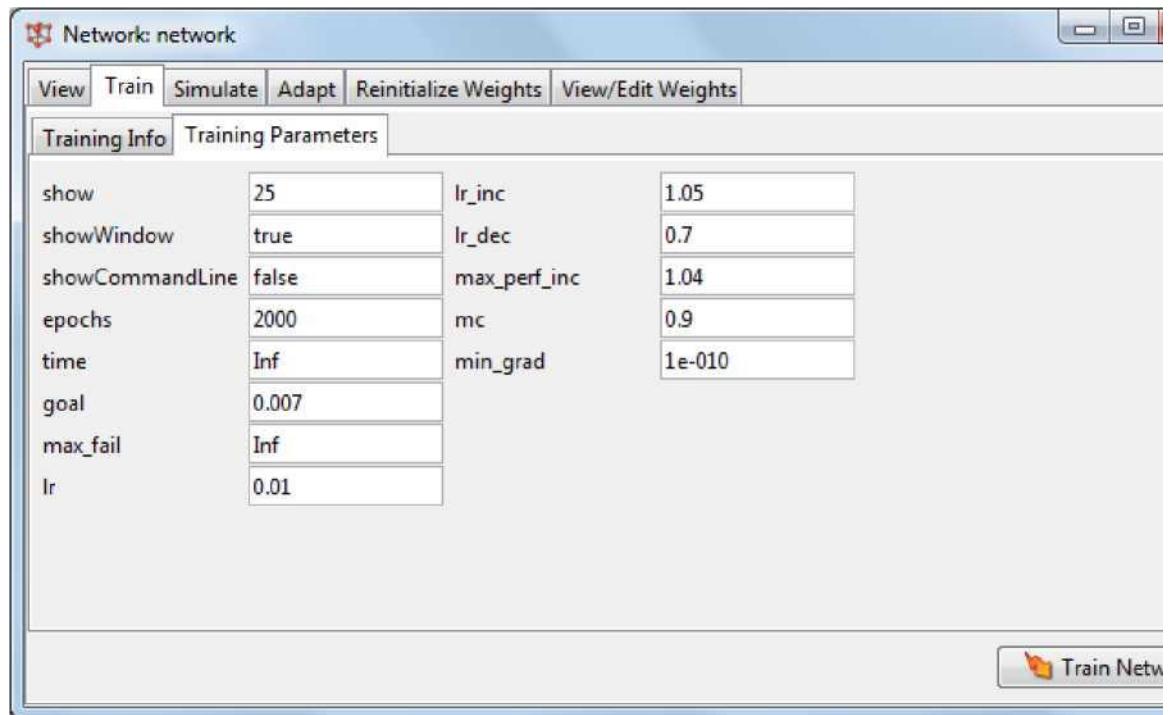


Рис. 15. Параметра обучения на данных с шумом
Окно процесса обучения будет выглядеть как рис.16.

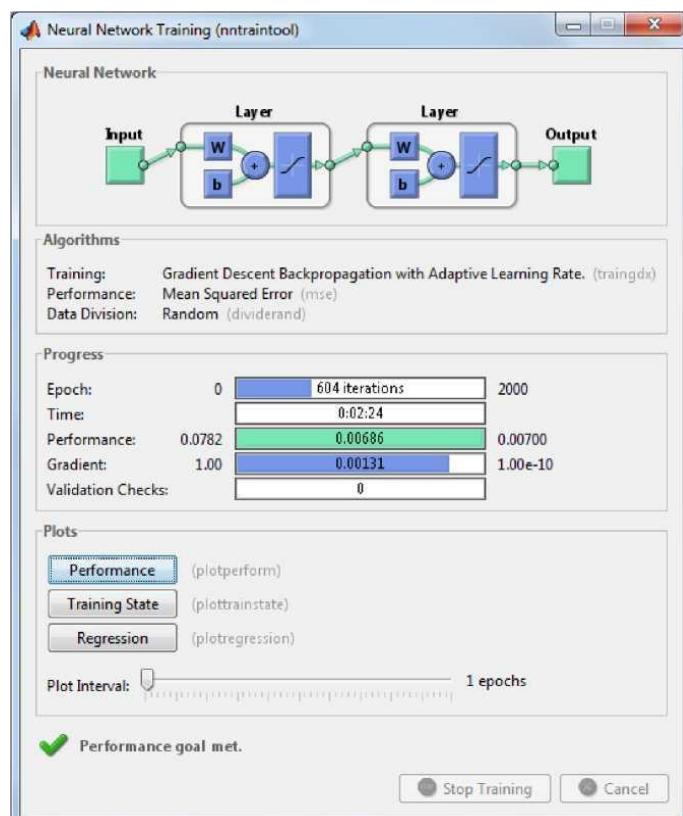


Рис. 16. Процесс обучения сети на данных с шумом

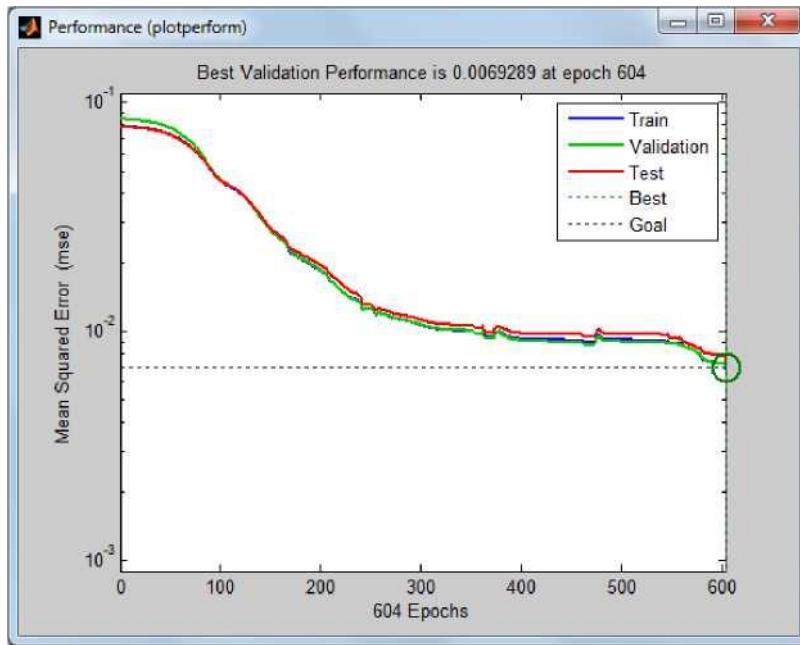


Рис. 17. График обучения нейронной сети на данных с шумом

В результате сеть обучена. Теперь необходимо проверить нейронную сеть. Для этого создана переменная noisy10, которая содержит в себе символ «И» с шумом.

Перейдем на вкладку Simulate, в окне созданной сети (рис.18).

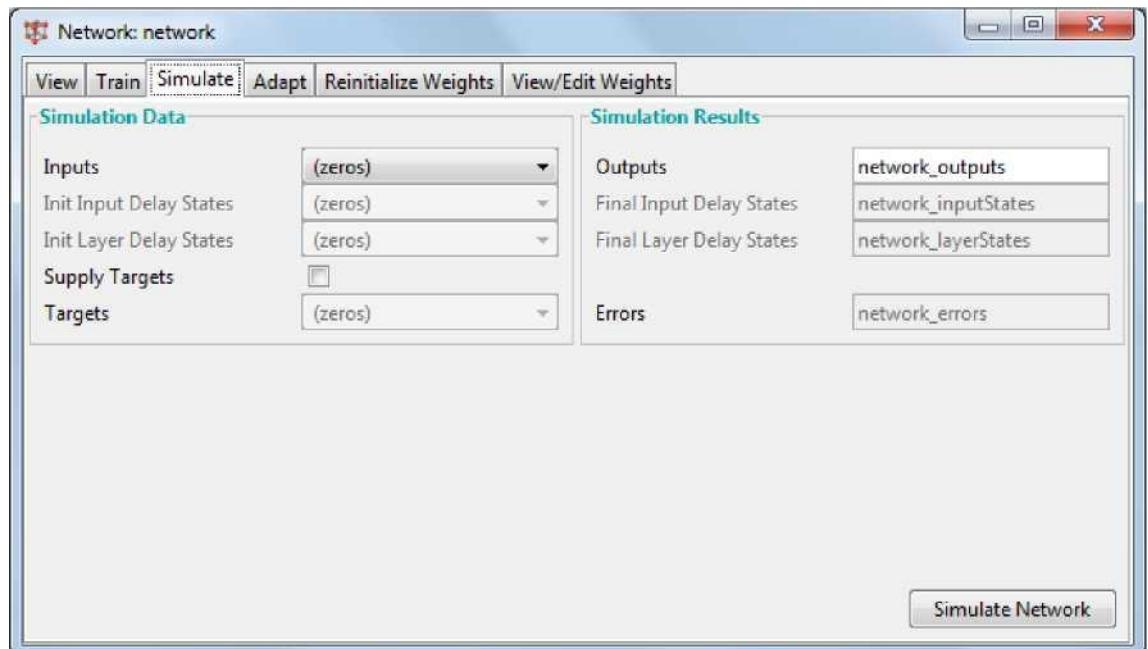


Рис. 18. Вкладка Simulate

Выберем в качестве входных данных переменную noisy10, а в качестве

выходных данных напишем переменную Ans (рис. 19).

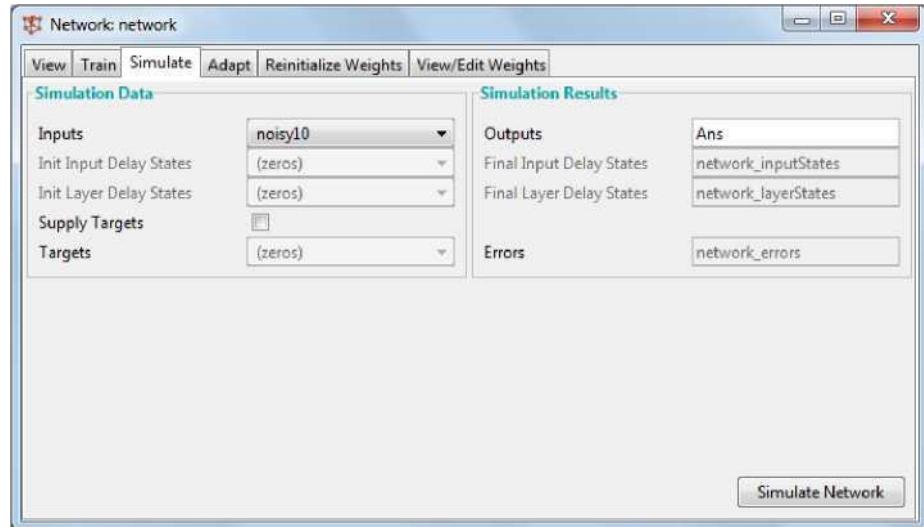


Рис. 19. Вкладка Simulate с выбранными параметрами

Для начала процесса симуляции необходимо нажать кнопку Simulate Network.

После этого в окне менеджера данных нейронной сети появится переменная Ans (рис.20).

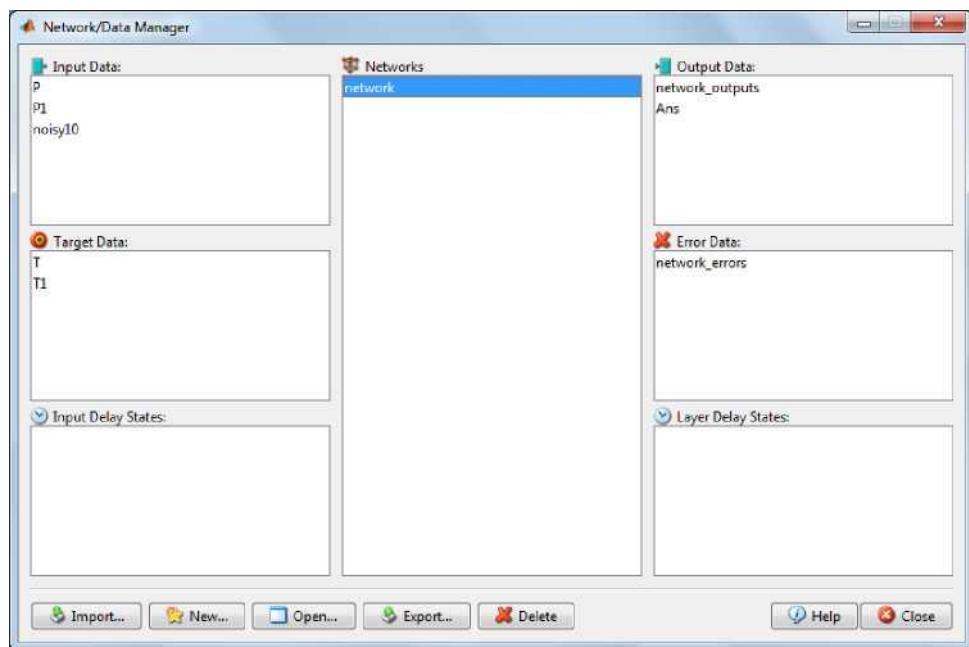


Рис. 20. Окно менеджера данных нейронной сети после проведенной симуляции

Экспортируем нейронную сеть и переменную Ans. Для этого

необходимо нажать кнопку Export и в появившемся окне выбрать необходимые нам переменные и еще раз нажать кнопку Export (рис.21).

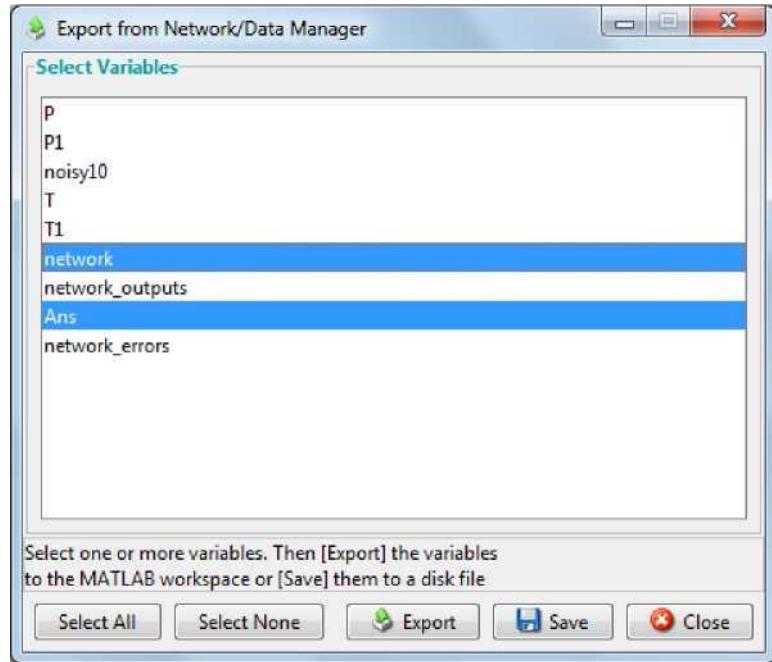


Рис 21. Экспорт переменных

После экспорта в окно команд Matlab вводим следующий фрагмент кода:

```
%Проверяем результат распознания
```

```
Ans = compet(Ans);
```

```
answer = find(compet(Ans) == 1)
```

```
plotchar(P(:,answer)); % Распознанный символ И
```

Перед нами появится окно (рис.22), отображающее распознанный символ.

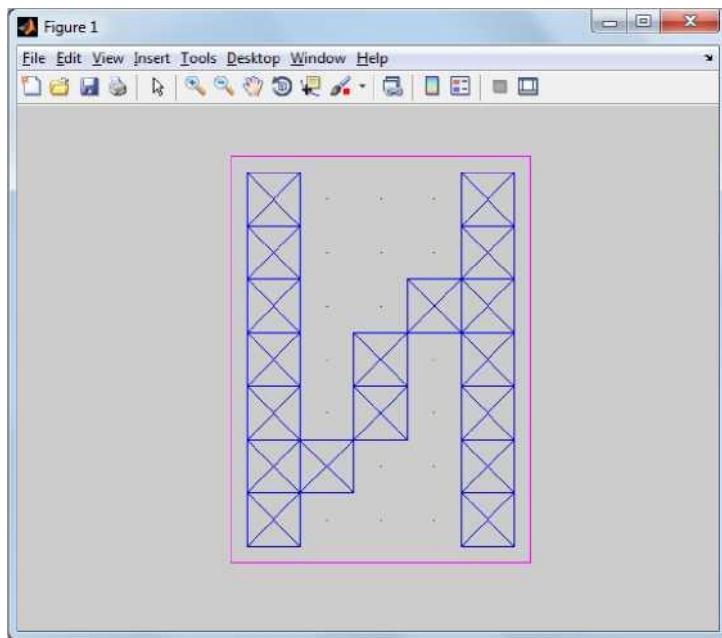


Рис. 22. Распознанный символ И

А в окне команд появится строчка:

```
answer =  
10
```

Означающая, что поступивший символ - это символ под номером 10 в нашем алфавите.

Таким образом, созданная нейронная сеть выполняет поставленную задачу.

Данная работа знакомит с GUI-интерфейсом NNtoot и может в дальнейшем использоваться студентами для создания более сложных, по своей структуре и задаче, нейронных сетей в различных курсах [1-4].

3.5. Распознавание лиц на основе нейронной сети использование Matlab

В этой работе мы предлагаем обозначить самоорганизующуюся карту (SOM) для измерения сходства изображений. Чтобы справиться с этой

задачей, мы подаем изображения лица, связанные с интересующими регионами, в нейронную сеть. В конце этапа обучения каждая нейронная единица настраивается на определенный прототип изображения лица. Распознавание лиц затем выполняется по правилу вероятностного решения. Эта схема предлагает очень многообещающие результаты для идентификации лица, касающегося изменения освещенности, а также поз и выражений лица. Эта статья представляет новую самоорганизующуюся карту (SOM) для распознавания лиц. Метод SOM обучен на изображениях из одной базы данных. Новизна этой работы заключается в интеграции изображений из входной базы данных, обучения и картирования. Распознавание лиц с использованием SOM в режиме без присмотра в нейронной сети. Среди архитектур и алгоритмов, предложенных для искусственной нейронной сети, самоорганизующаяся карта обладает особым свойством эффективного создания пространственно организованного «внутреннего представления» различных характеристик входных сигналов и их абстракций. После тщательной настройки весовых векторов самоорганизующаяся карта оказалась особенно успешной в различных задачах распознавания образов, связанных с очень шумным сигналом. Каждый разрабатывает реалистичные корковые структуры, когда в качестве входных данных даны приближения визуальной среды, и является эффективным способом моделирования развития способности распознавания лиц.

I. Здесь мы разработали и проиллюстрировали систему распознавания человеческих лиц с использованием новой поисковой системы на основе самоорганизующейся карты Кохонена (SOM). SOM обладает хорошими характеристиками извлечения благодаря своему топологическому упорядочению. Результаты Facial Analytics для 400 изображений базы данных AT & T показывают, что частота распознавания лиц с использованием одного из алгоритмов нейронной сети SOM составляет 92,40% для 40 человек.

II. Ключевые слова

III. SOM (самоорганизующаяся картография), PCA (анализ главных компонентов), ICA (независимый анализ компонентов).

Введение

В современном сетевом мире необходимость поддерживать безопасность информации или физических свойств становится все более важной и все более сложной. Время от времени мы слышим о преступлениях, связанных с мошенничеством с кредитными картами, взломом компьютеров хакерами или нарушениями безопасности в компании или правительственном здании. Само собой разумеется, что если кто-то украл дубликаты. Недавно стали доступны технологии, позволяющие подтвердить «истинную» индивидуальную личность. Эта технология основана на области, называемой «биометрия», как показано на рис. 1 и рис. 2. Распознавание лиц является одним из немногих биометрических методов [5], которые обладают преимуществами как высокой точности, так и низкой интрузивности. Он обладает точностью физиологического подхода, не будучи навязчивым.

IV. По этой причине с начала 70-х годов (Kelly, 1970) распознавание лиц привлекло внимание исследователей в таких областях, как безопасность, психология и обработка изображений, к компьютерному зрению. Для распознавания лиц было предложено множество алгоритмов, таких как PCA, ICA [1] [2].

V. Что такое СОМ СОМ [3]

- VI. • Конкурентное обучение ИНН
- VII. • Каждая единица карты уменьшает идентичные входные данные.
- VIII. • Единицы борются за выбор
- IX. • Модификация выбранного узла и его соседей.

Способность человека распознавать отдельных людей исключительно по человеческому лицу весьма примечательна. Эта способность сохраняется даже в течение времени, изменений во внешнем виде и частичной окклюзии. Из-за этой замечательной способности генерировать почти идеальные положительные идентификационные данные значительное внимание

уделялось методам, с помощью которых эффективное распознавание лиц может быть воспроизведено на электронном уровне. Конечно, если такой сложный процесс, как идентификация человека, основанный на неинвазивном методе, таком как распознавание лиц, может быть достигнут с помощью электроники, то такие области, как безопасность банков и аэропортов, могут быть значительно улучшены, кража личных данных может быть дополнительно уменьшена и частная безопасность сектора может быть повышена.

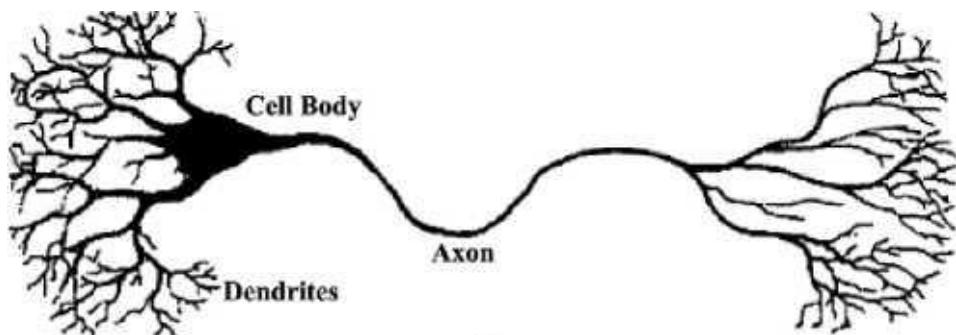
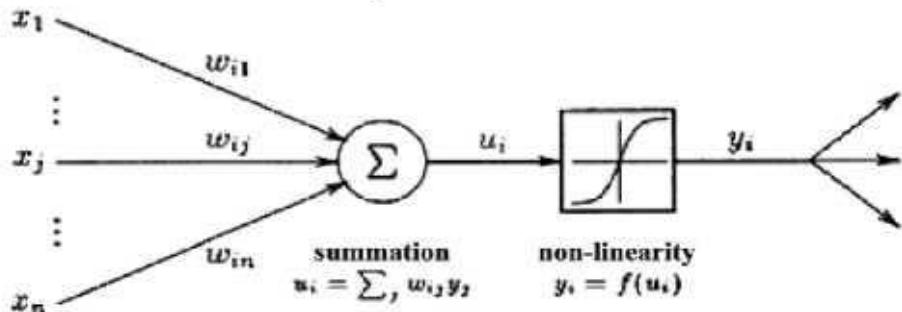


Fig. 1 Neuron Cell



Поза, Освещение, Выражение лица, Состояние изображения, Размер лица.

A. Классификация распознавания лиц

Сценарии распознавания лиц можно разделить на два типа: рис. 3: проверка лица (или аутентификация) и идентификация лица (или распознавание).

1) Проверка лица. Это сопоставление «один-к-одному», при котором изображение лица запроса сравнивается с изображением лица шаблона, личность которого запрашивается. Чтобы оценить эффективность проверки, отображается график частоты проверки (скорость, с которой законным

пользователям предоставляется доступ) в сравнении с частотой ложных приемов (скорость, с которой предоставляется самозванцам), называемая кривой ROC. Хорошая система проверки должна уравновешивать эти два показателя с учетом эксплуатационных потребностей.

2) Идентификация лица: это процесс сопоставления «один ко многим», который сравнивает изображение лица запроса со всеми шаблонными изображениями в базе данных лиц, чтобы определить идентичность лица запроса. Идентификация тестового изображения выполняется путем размещения изображения в базе данных, которое имеет наибольшее сходство с тестовым изображением [4].

Процесс идентификации - это «закрытый» тест, который означает, что датчик проводит наблюдение за человеком, который, как известно, находится в базе данных. (Нормализованные) функции объекта тестирования сравниваются с другими функциями в базе данных системы, и для каждого сравнения определяется показатель сходства. Эти оценки сходства затем численно ранжируются в порядке убывания. Процент случаев, когда наивысший показатель сходства является правильным соответствием для всех лиц, называется «максимальным результатом совпадения».

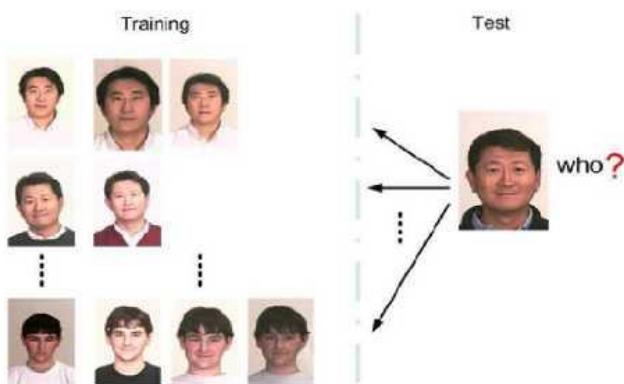


Рис.3. Сравнение нового изображения с базой данных

A. Требования к оборудованию

- 1) Компьютерная система P-4: процессор Intel @ 2,93 ГГц Core 2 Dual / D2C, 2 ГБ оперативной памяти DDR2, гигабитная материнская плата G41, встроенная звуковая карта и AGP-карта, жесткий диск 60 ГБ, шкаф с SMPS, загруженный с QC Windows XP / Vista / Windows 7.

- A. 2) Веб-камера, сканер
- B. Требования к программному обеспечению
- C. 1) Matlab 7.8, Matlab 2008, Панель инструментов обработки изображений
- D. 2) Аутсорсинговая база данных
- E. 3) АТ & Т База данных изображений
- F. XIV. СОМ АЛГОРИТМ И ФЛОХАРТ
- G. А. Шаги алгоритма
- H. Шаг 1 - Инициализация весов. Шаг 2 - Получение наилучшего подходящего элемента Шаг 3 - Соседи по шкале:
 - a) Определение соседей
 - b) Обучение
- I. Шаг 4 - Картрирование: Определение качества СДЛ.
- K. В. Блок-схема

Блок-схема последовательности двух шагов А) Обучение В) Тестирование

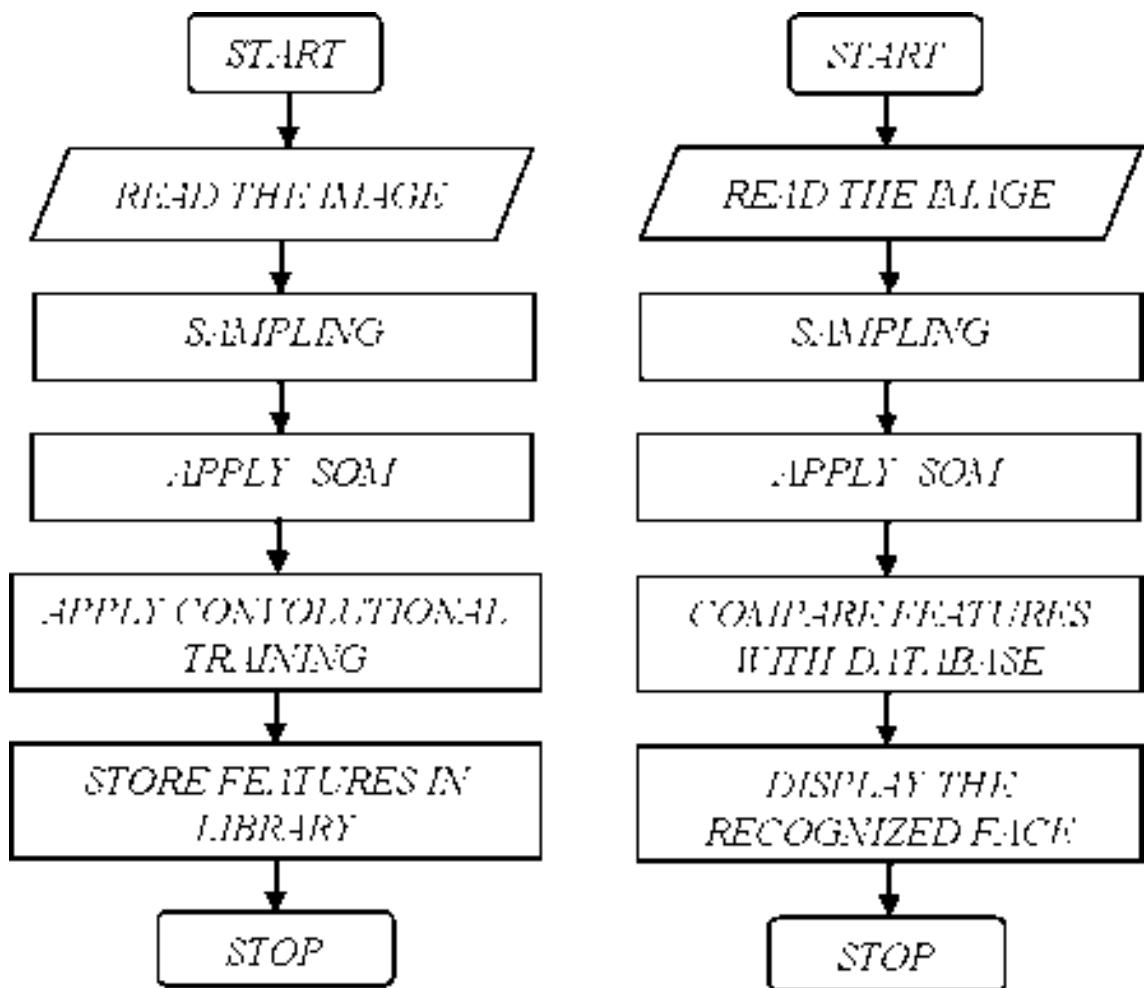
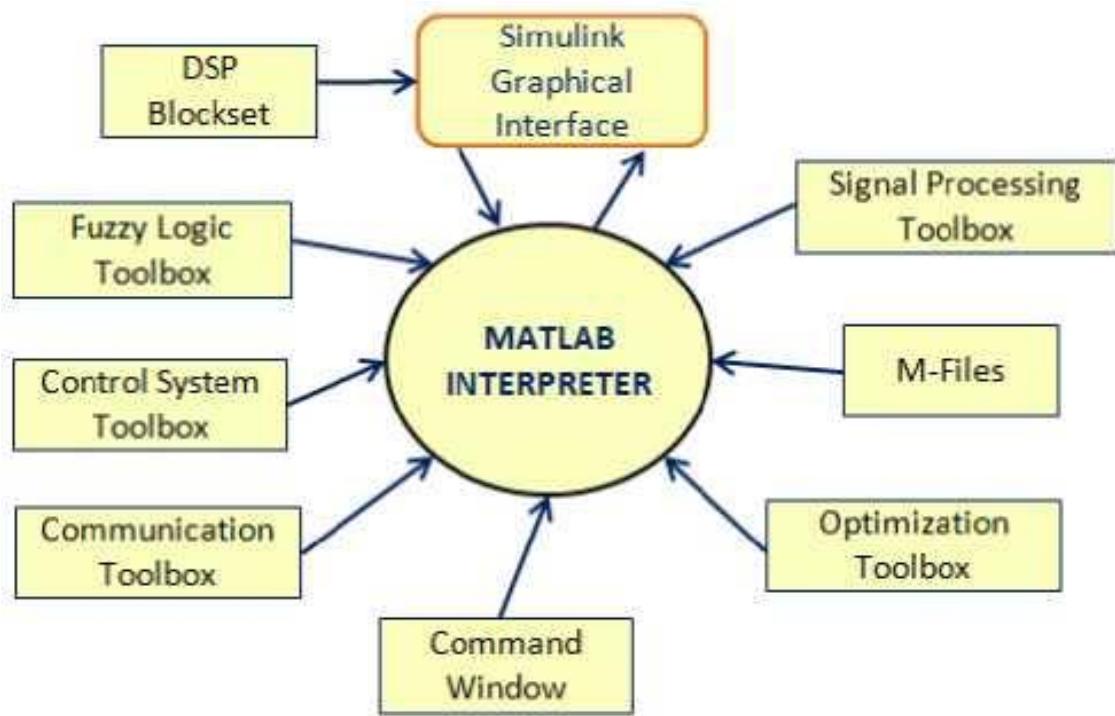


Рис. 4 Блок-схема для распознавания лиц

Название MATLAB означает матричную лабораторию, изначально написанную для обеспечения легкого доступа к программному обеспечению матрицы, разработанному проектами LINPACK и EISPACK. Сегодня движки MATLAB включают библиотеки LAPACK и BLAS, внедряя современное состояние в программное обеспечение для матричных вычислений. Рис. 5.

MATLAB - это интерактивная матричная система для научных и инженерных численных расчетов и визуализации. Основной элемент данных - это массив, который не требует измерения. Он используется для решения многих технических вычислительных задач, особенно с матричной и векторной формулировками, за долю времени, которое потребуется для написания



ЭКСПЕРИМЕНТАЛЬНЫЕ ДАННЫЕ

Публично доступная база данных AT & T [6] используется для экспериментов по распознаванию. В базе данных рассматриваются 10 различных изображений каждого из 40 человек (всего 400 изображений) с различиями в углах лица, выражениях лица и деталях лица. Предварительное изображение базы данных лиц показано на рисунке 6

программы на скалярном неинтерактивном языке, таком как С или FORTRON.



Рис. 6 Набор данных AT & T

РЕЗУЛЬТАТЫ

Вычислительная эффективность метода SOM показана в таблице 1 и на рисунке 7.

TABLE I
RESULT OF SOM ON ORL DATABASE

SRN	DB Image	Accuracy (Recognition Rate)
1	3	81.80%
2	4	83.70%
3	5	86.00%
4	6	92.00%
5	7	92.40%

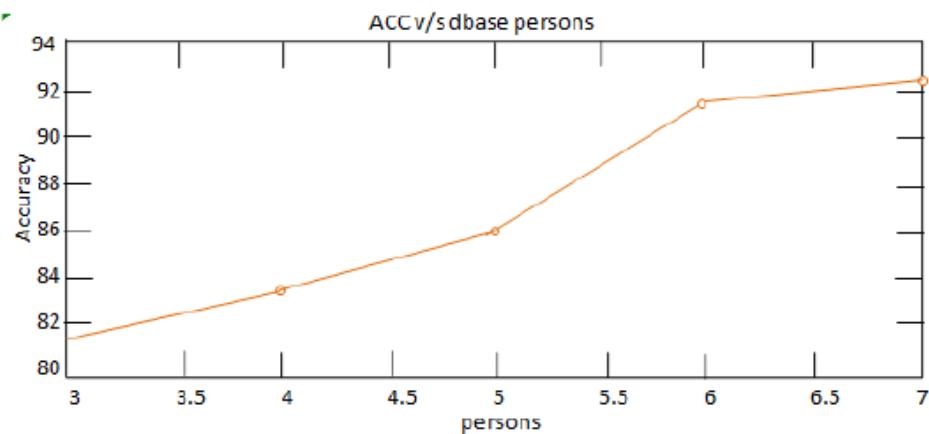


Рис.7 Точность в сравнении с персоналом базы данных

Преимущество распознавания лиц заключается в том, что это неинтрузивная техника, которая может быть эффективной без сотрудничества или знаний участника; это делает его особенно подходящим для целей наблюдения. Системы, например ATR (автоматическое распознавание целей), безопасность переписей людей и криминальная идентификация, используют методы распознавания лиц.

Мы видели метод самоорганизующейся карты (SOM) для распознавания лиц. Новизна этого документа заключается в интеграции входного изображения, извлечения функций, обучения и картирования. SOM

- это листовая искусственная нейронная сеть, клетки которой специально настраиваются на различные шаблоны или классы входного сигнала в процессе обучения без присмотра. Каждая ячейка или локальная группа ячеек действуют как отдельный декодер для одного и того же входа, таким образом, это наличие или отсутствие активного отклика в местоположении, а не точная трансформация сигнала ввода-вывода или величина отклика, что обеспечивает интерпретацию входной информации. СДЛ уменьшить размеры и отображать сходства.

Самоорганизующиеся карты упорядочены топологически, что приводит к хорошим возможностям извлечения. Самый высокий средний показатель распознавания, достигнутый с помощью SINGLE ALGORITHM SOM, составляет 92,40%, полученный для 400 изображений из 400 изображений базы данных AT & T. Таким образом, наши экспериментальные результаты показывают, что сложность системы распознавания лиц резко уменьшается при использовании SOM.

ЗАКЛЮЧЕНИЕ

Однонаправленные нейронные сети с сигмоидальной функцией активации широко применяются на практике, составляя важное звено процесса выработки решений. В работе приводятся несколько приложений, позволяющим подчеркнуть универсальность и разнородность функций, которые искусственные нейронные сети могут выполнять.

Полная интеграция с мощными автоматическими инструментами STATISTICA; запись полноценных макросов для любых анализов; создание собственных нейро-сетевых анализов и приложений с помощью STATISTICA Visual Basic, вызов STATISTICA Neural Networks из любого приложения, поддерживающего технологию COM (например, автоматическое проведение нейросетевого анализа в таблице MS Excel или объединение нескольких пользовательских приложений, написанных на C++, C#, Java и т.д.).

Цель разрабатываемого шаблона, создать нейронную сеть, которая сможет распознавать визуальные образы букв русского алфавита. Программный код, который решает подобную задачу, присутствует в системе Matlab, как демонстрационная программа с названием appcr1. Подробно этот код разобран и пояснен в книге «Нейронные сети» В. С. Медведева, В. Г. Потемкина [6], а так же описан в работе И. С. Миронова, С. В. Скурлаева [7]

Данная работа знакомит с GUI-интерфейсом NNtoot и может в дальнейшем использоваться студентами для создания более сложных, по своей структуре и задаче, нейронных сетей в различных курсах

СПИСОК ЛИТЕРАТУРЫ

[1 %8B/%D0%9B%D0%B0%D0%B1 %D0%BE%D1 %80%D0%B0%D1 %82%D0%BE%D1 %80%D0%BD%D1 %8B%D0%B5/Lab%202.pdf](#)

9. Шеремет А.И., Перепелица В.В., Денисова А.М. Проектирование нейронной сети для распознавания символов в программной среде MATLAB [Электронный ресурс].

URL:

<http://nauka.zinet.info/13/sheremet.php>

10. Principe J.C., Euliano N.R., Lefebvre W.C. Neural and Adaptive Systems. Fundamentals Through Simulations. New York. John Wiley & Sons Inc. 2000.

11. Luo F-L., Unbehauen R. Applied Neural Networks for Signal Processing. Cambridge University Press. 1998.

12. Demuth H., Beale M. Neural Network Toolbox. For Use with MATLAB. The MathWorks Inc. 1992-2000.

13. Awadalla M. H. A., Ismaeil I. I., Sadek M. A. Spiking neural network-based control chart pattern recognition //Journal of Engineering and Technology Research. 2011. Т. 3. №. 1. С. 5-15.

14. Dede G., Sazli M. H. Speech recognition with artificial neural networks //Digital Signal Processing. 2010. Т. 20. №. 3. С. 763-768.

15. Осовский, С. Нейронные сети для обработки информации / С. Осовский. - М.: Финансы и статистика, 2002. - 344 с.

16. Хайкин, С. Нейронные сети: полный курс, 2-е издание / С. Хайкин. - М.: Издательский дом «Вильямс», 2006. — 1104 с.

17. Каллан, Р. Основные концепции нейронных сетей / Р. Каллан. - М.: Издательский дом «Вильямс», 2001. - 287 с.

18. Marquardt, D. An algorithm for least squares estimation of nonlinear parameters / D. Marquardt // SIAM. - 1963. - С. 431- 442.

19. Bishop, C. M. Neural network for pattern recognition / C.M. Bishop. - Oxford: Oxford University Press, 2005. - 482 с.