

Министерство образования Республики Беларусь  
Учреждение образования «Витебский государственный  
университет имени П.М. Машерова»  
Кафедра информатики и информационных технологий

# **ДОПОЛНИТЕЛЬНЫЕ ГЛАВЫ ИНФОРМАТИКИ**

*Методические рекомендации  
к выполнению лабораторных работ*

*Витебск  
ВГУ имени П.М. Машерова  
2017*

УДК 004.032.26+004.414.23(076.5)

ББК 32.971.92я73+32.972.11я73

Д68

Печатается по решению научно-методического совета учреждения образования «Витебский государственный университет имени П.М. Машерова». Протокол № 1 от 19.10.2017 г.

Составитель: доцент кафедры информатики и информационных технологий ВГУ имени П.М. Машерова, кандидат физико-математических наук **С.А. Прохожий**

Рецензент:

доцент кафедры логистики и менеджмента  
ВФ УО ФПБ «Международный университет “МИТСО”»,  
кандидат педагогических наук *А.С. Слепухина*

**Д68** **Дополнительные главы информатики** : методические рекомендации к выполнению лабораторных работ / сост. С.А. Прохожий. – Витебск : ВГУ имени П.М. Машерова, 2017. – 36 с.

Методические рекомендации содержат задания для лабораторных занятий и самостоятельной работы по предмету «Дополнительные главы информатики», а также краткие теоретические сведения, необходимые для успешного их выполнения. Издание описывает содержание комплекса лабораторных работ по нейросетевым технологиям на базе программного продукта Statistica Neural Networks. Рекомендуется для практического освоения типовых задач интеллектуальных информационных систем, эффективно решаемых с помощью искусственных нейронных сетей (ИНС).

Предназначено для студентов дневной и заочной форм обучения по дисциплине «Дополнительные главы информатики» специальности «Программное обеспечение информационных технологий».

УДК 004.032.26+004.414.23(076.5)

ББК 32.971.92я73+32.972.11я73

© ВГУ имени П.М. Машерова, 2017

## ВВЕДЕНИЕ

Интеллектуальные информационные технологии – это информационные технологии, помогающие человеку ускорить анализ политической, экономической, социальной и технической ситуации, а также синтез управленческих решений. При этом используемые методы не обязательно должны быть логически непротиворечивы или копировать процессы человеческого мышления. На практике наиболее распространены два подхода – либо оперирование с формализованными знаниями человека на основе символического подхода (экспертные системы, нечеткая логика), либо со свойствами человека приемами обучения и мышления (искусственные нейронные сети и эволюционные вычисления).

Во втором случае при построении интеллектуальной информационной системы (далее ИИС) не требуется априорное знание обо всех закономерностях исследуемой предметной области, но необходимо располагать достаточным количеством примеров для настройки разрабатываемой адаптивной ИИС. После обучения на заданных параметрах такая ИИС способна формировать требуемое решение с определенной степенью приближения. В основе указанной адаптивной ИИС содержится искусственная нейронная сеть (далее ИНС).

ИНС представляет собой достаточно новую и перспективную вычислительную технологию, способствующую новым подходам к решению целого ряда задач в науке, технике, экономике и бизнесе. ИНС вошли в практику там, где нужно решать неформализованные задачи, для которых нет известных алгоритмов решения. Для таких задач нейросетевые технологии по эффективности могут существенно превышать традиционные методы.

ИНС наилучшим образом проявляют себя при большом количестве данных, для которых характерны неявные взаимозависимости и закономерности. Здесь ИНС автоматически извлекают и используют различные зависимости, скрытые в данных. К задачам, эффективно решаемым с помощью ИНС, в настоящее время относятся следующие:

- аппроксимация функций (построение моделей плохо формализуемых зависимостей), например, в экономике оценка стоимости недвижимости, контроль качества выпускаемой продукции;
- классификация образов (указание принадлежности входного образа предварительно определенным классам), например, оценка кредитоспособности, оценка вероятности банкротства;
- кластеризация (классификация образа «без учителя»), например, анализ деятельности конкурирующих фирм, сжатие информации в информационных хранилищах;
- прогнозирование, например, экономических параметров и фондовых индексов, объемов продаж;

- ассоциативная память (память, адресуемая по содержанию), например, для создания хранилищ данных, мультимедийных баз данных;
- оптимизация (нахождение решения, удовлетворяющего системе ограничений и макси- или минимизирующее целевую функцию), например, доходности работы компании, задачи ценообразования;
- управление, например, товарными запасами, динамическими объектами (оптимальное).

Но значительная часть задач моделирования, решаемых ИНС, сводится к представлению в виде аппроксимации, а ИНС выполняет роль универсального инструмента для аппроксимации векторных функций нескольких переменных.

Цель настоящих методических рекомендаций – получение навыков построения ИНС и практического решения распространенных нейросетевых задач, в том числе с использованием программного пакета STATISTICA Neural Networks.

Издание состоит из трех глав. Глава 1 затрагивает общие понятия задачи распознавания образов. В главе 2 исследуется архитектура сверточных нейронных сетей. Глава 3 содержит четыре лабораторные работы для практического освоения решения типовых нейросетевых задач с использованием пакета SNN.

# 1. ЗАДАЧА РАСПОЗНАВАНИЯ ОБРАЗОВ

## 1.1 Постановка задачи

Пусть есть некоторое множество  $U$  – множество объектов реального мира. К каждому объекту из этого множества можно отнести какие-то признаки – элементы вектора  $X$ . Также у каждого из этих объектов есть какой-то класс, который необходимо определить, имея признаки объекта. Множество, элементы которого являются классом, обозначим как  $Y$ .

*Задача распознавания образов* - это задача отнесения объектов к определенному классу с помощью выделения существенных признаков, характеризующих эти объекты, что продемонстрировано на рис. 1.1.

### Задача машинного обучения (на примере классификации)

Решающая функция

$$F: X \rightarrow Y$$

Обучающая выборка

$$D \left\{ \begin{array}{|c|c|} \hline x_1 & y_1 \\ \hline x_2 & y_2 \\ \hline \vdots & \vdots \\ \hline x_N & y_N \\ \hline \end{array} \right.$$

Рисунок 1.1 – Задача машинного обучения

Задачей распознавания образов, или задачей машинного обучения в данном случае является построение решающей функции, которая каждому вектору признаков объекта будет ставить в соответствие меру принадлежности к определенному классу. [1]

*Прецедент* – объект, правильная классификация которого известна.

В зависимости от наличия или отсутствия прецедентной информации различают задачи распознавания с обучением и без обучения. Задача распознавания на основе имеющегося множества прецедентов называется *классификацией с обучением* (или с учителем).

В том случае, если имеется множество векторов признаков, полученных для некоторого набора образов, но правильная классификация этих образов неизвестна, возникает задача разделения этих образов на классы по сходству соответствующих векторов признаков. Эта задача называется *кластеризацией*, или *распознаванием без обучения*. [2]

## 1.2 Сравнительная характеристика методов распознавания изображений

Сравнение будет происходить на основе четырех характеристик:

- скорость обучения;
- максимальная достижимая точность распознавания изображений;
- адаптивность (инвариантность к преобразованиям входных данных);

- универсальность (применимость к обработке различного рода данных).

Среди сравниваемых методов распознавания изображений будут присутствовать как классические (вероятностный классификатор, сравнение с эталоном); так и нейросетевые (персептрон, нейронная сеть, сверточная нейронная сеть).

#### Классификация на основе Байесовской теории решений

Байесовский подход исходит из статистической природы наблюдений. За основу берется предположение о существовании вероятностной меры на пространстве образов, которая либо известна, либо может быть оценена.

Цель состоит в разработке такого классификатора, который будет правильно определять наиболее вероятный класс для пробного образа. Тогда задача состоит в определении "наиболее вероятного" класса.

Характеристики:

*Скорость обучения:* плохая, так как производится исключительно силами пользователя, который определяет признаки.

*Максимальная точность распознавания:* хорошая, так как метод основан на теории вероятности.

*Адаптивность:* хорошая, но если классы плохо разделимы (что и происходит в большинстве реальных задач), то адаптивность данного метода резко снижается.

*Универсальность:* плохая, потому что высокая точность распознавания будет достигаться только при данных, максимально похожих на ожидаемые.

#### Классификация на основе сравнения с эталоном

Пусть задано множество образов (эталонов). Задача состоит в том, чтобы для тестируемого объекта выяснить, какой эталон ближе на основе меры сходства ("расстояния между объектами").

Характеристики:

*Скорость обучения:* плохая, так как производится исключительно силами пользователя, который определяет признаки и математическую модель.

*Максимальная точность распознавания:* средняя.

*Адаптивность:* у данного метода средняя адаптивность, потому что это свойство зависит от конкретной математической модели.

*Универсальность:* у данного метода средняя универсальность, потому что это свойство зависит от конкретной математической модели.

#### Линейный классификатор: алгоритм персептрона

В алгоритме персептрона в основу положен принцип действия нейрона. На вход нейрона поступают значения вектора признаков  $X$ , которые умножаются на соответствующие веса синапсов вектора весов  $W$ , после

чего суммируются с учетом прибавки некоторого порогового значения и обрабатывается специальной функцией активации.

Модель нейрона можно увидеть на рис. 1.2.

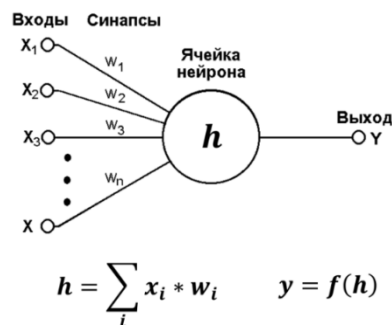


Рисунок 1.2 – Модель нейрона

Затем вычисляется специальная переменная ошибки путем вычитания получившегося отклика нейрона из ожидаемого, после чего на основе переменной ошибки происходит корректировка весов. [3]

Перцептрон может состоять из нескольких нейронов. Логическая схема типичного перцептрона приведена на рис. 1.3:

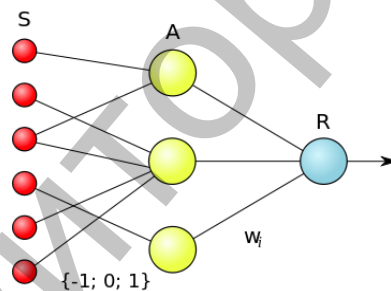


Рисунок 1.3 – Модель перцептрона

Веса S—A связей могут иметь значения  $-1$ ,  $+1$  или  $0$  (то есть отсутствие связи). Веса A—R связей  $W$  могут быть любыми.

Характеристики:

*Скорость обучения:* перцептрон обучается самостоятельно, но в то же время количество связей между нейронами зачастую очень высоко.

*Макс. точность распознавания:* хорошая из-за постоянной корректировки весов в процессе обучения.

*Адаптивность:* у данного метода средняя адаптивность.

*Универсальность:* плохая из-за свойства линейности данного метода, а, следовательно, в неспособности производить новые, обобщенные признаки на основе уже определенных.

Нелинейный классификатор: многослойная нейронная сеть

Используется нейронная сеть, состоящая из слоев, которые в свою очередь состоят из множества нейронов. Нейроны соседних слоев связаны

с друг другом синаптическими связями, а коррекция весов происходит с помощью функции обратного распространения ошибки. [3]

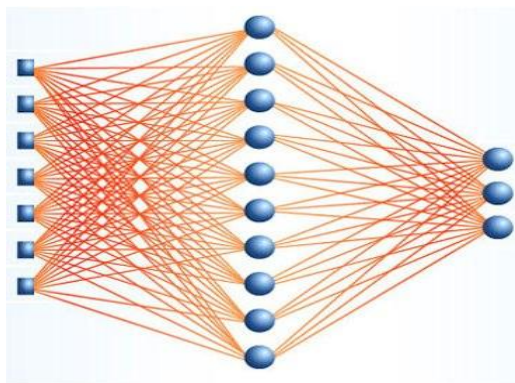


Рисунок 1.4 – Модель многослойной нейронной сети

Крайний левый и крайний правый вертикальные ряды объектов на Рисунке 1.4 – входной и выходной слою соответственно. Средний ряд – скрытый слой, скрытых слоев может быть несколько.

При определенных обстоятельствах нейронная сеть с одним скрытым слоем будет являться персептроном, поэтому рассматривается именно многослойная нейронная сеть – т.е. нейронная сеть с несколькими скрытыми слоями.

Характеристики:

*Скорость обучения:* средняя; нейронная сеть обучается самостоятельно, но в то же время количество связей между нейронами очень высоко.

*Макс. точность распознавания:* хорошая из-за постоянной корректировки весов с помощью обратного распространения ошибки в процессе обучения.

*Адаптивность:* у данного метода хорошая адаптивность, лежащая в сути самого алгоритма.

*Универсальность:* средняя.

*Нелинейный классификатор: сверточная нейронная сеть*

Сверточная сеть – это нейронная сеть со специфической структурой, заключающейся в чередующихся операциях извлечения и сжатия карт признаков, уменьшая таким образом скорость обучения, чувствительность к исходным данным и делая возможным выработку более обобщенных правил.

Характеристики:

*Скорость обучения:* хорошая, так как нейронная сеть обучается самостоятельно; также из-за особенностей ее архитектуры количество синаптических связей значительно снижено по сравнению с другими нейронными сетями.

*Макс. точность распознавания:* отличная из-за постоянной корректировки весов с помощью обратного распространения ошибки в процессе обучения.



*Адаптивность:* у данного метода отличная адаптивность из-за использования карт признаков.

*Универсальность:* у данного метода отличная универсальность из-за свойства нелинейности данного метода, а, следовательно, способности производить новые, обобщенные признаки на основе уже определенных.

### Итоговое сравнение

Результаты итогового сравнения методов распознавания изображений можно увидеть в Таблице 1.1.

<b>Метод</b>	<b>Качество</b>	<b>Скорость обучения</b>	<b>Макс. точность</b>	<b>Адаптивность</b>	<b>Универсальность</b>
<i>Байесовский классификатор</i>		плохая	хорошая	хорошая	плохая
<i>Сравнение с эталоном</i>		плохая	средняя	средняя	средняя
<i>Перцептрон</i>		средняя	хорошая	средняя	плохая
<i>Многослойная НС (не перцептрон)</i>		средняя	хорошая	хорошая	средняя
<i>Сверточная нейронная сеть</i>		хорошая	<b>отличная</b>	<b>отличная</b>	<b>отличная</b>

Таблица 1.1 – Результаты итогового сравнения

Из таблицы видно, что среди всех остальных методов лидирует сверточная нейронная сеть (далее – СНС).

## 2. АРХИТЕКТУРА СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

### 2.1 Понятие сверточной нейронной сети

*Сверточные нейронные сети* (Convolutional Neural Networks, CNN) являются представителями класса моделей, поводом к созданию которых послужили исследования зрительного аппарата кошек, проведенные Хубелем и Вейселем в 1960-х [4].

Их результатом было открытие двух типов клеток, влияющих на зрительную восприимчивость: первые обладают свойством локальной чувствительности и предназначены для выделения элементарных характеристик образов (ориентированных краев, конечных точек, углов), вторые же путем их комбинирования осуществляют построение высокоуровневых признаков.

Известны различные реализации сверточных нейронных сетей, отличающиеся топологией слоев, способом организации процесса обучения и другими признаками.

Типичный пример структуры сверточной нейронной сети представлена на рис. 2.1.

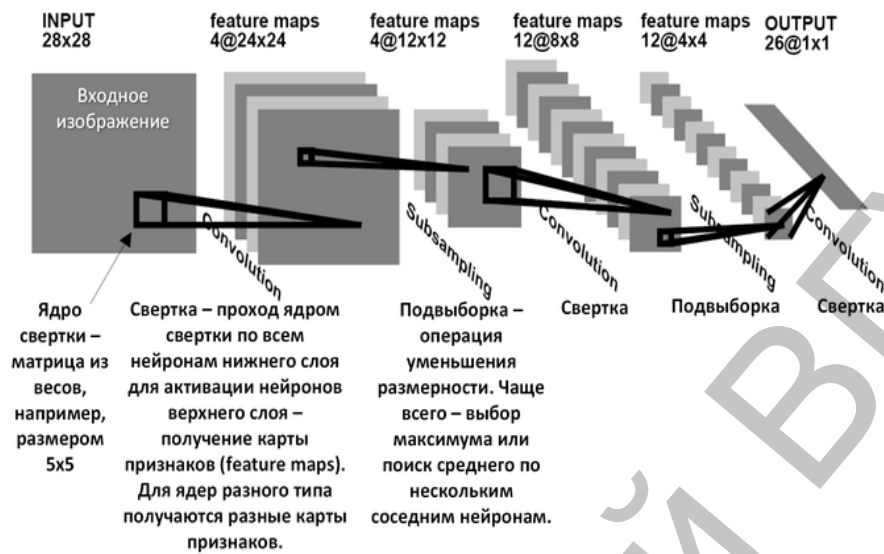


Рисунок 2.1 – Пример структуры сверточной нейронной сети

## 2.2 Слой свертки

Операция *свертки изображения* – это операция между матрицей изображения и матрицей-ядром (фильтром), при которой каждый элемент (пиксель) в выходном изображении есть сумма произведений значения элемента ядра на значение соответствующего покрываемого элемента матрицы входящего изображения.

В зависимости от коэффициентов фильтра значения тех или иных пикселей из окрестности некоторого пикселя во входном изображении будут иметь большее или меньшее влияние на значение пикселя в новом изображении. [5]

Оперция свертки наглядно показана на рис. 2.2.

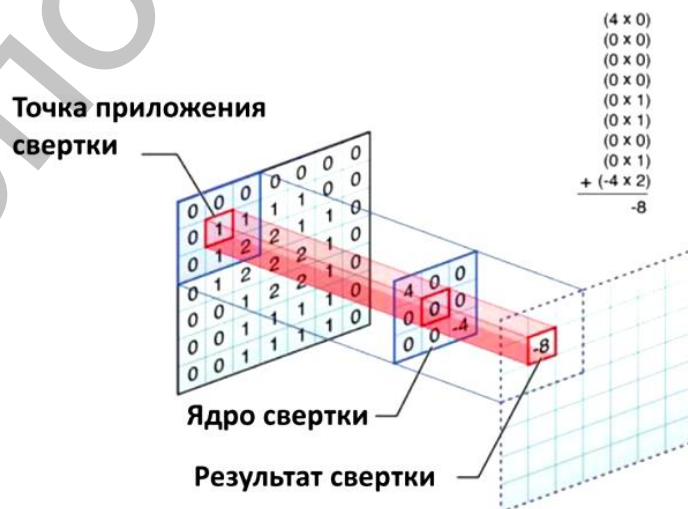


Рисунок 2.2 – Операция свертки

Операция свертки используется в СНС для составления карт признаков. Рассмотрим на примере то, как это происходит: пусть имеем фильтр с ядром, в котором закодирован признак скругленной вправо и вверх линии.

Фильтр и его визуализация приведены на рис. 2.3.

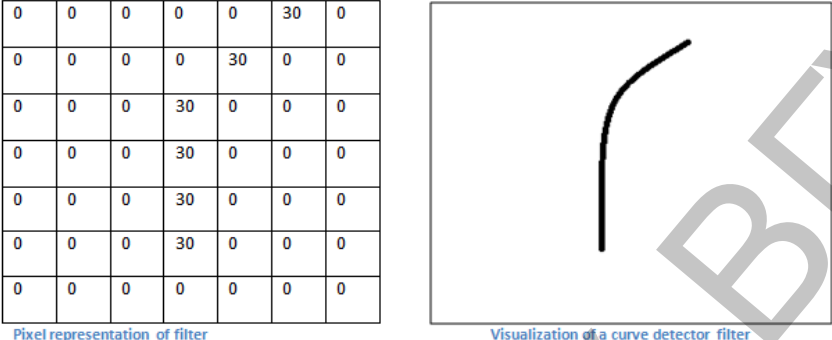


Рисунок 2.3 – Фильтр и его визуализация

Применим окно фильтра в определенной области рисунка мыши, которая продемонстрирована на рис. 2.4.

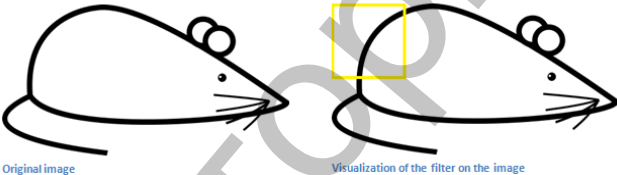


Рисунок 2.4 – Область применения фильтра

В результате применения фильтра на данную область, было получено большое число, что показано на рис. 2.5. Это означает, что признак распознан.

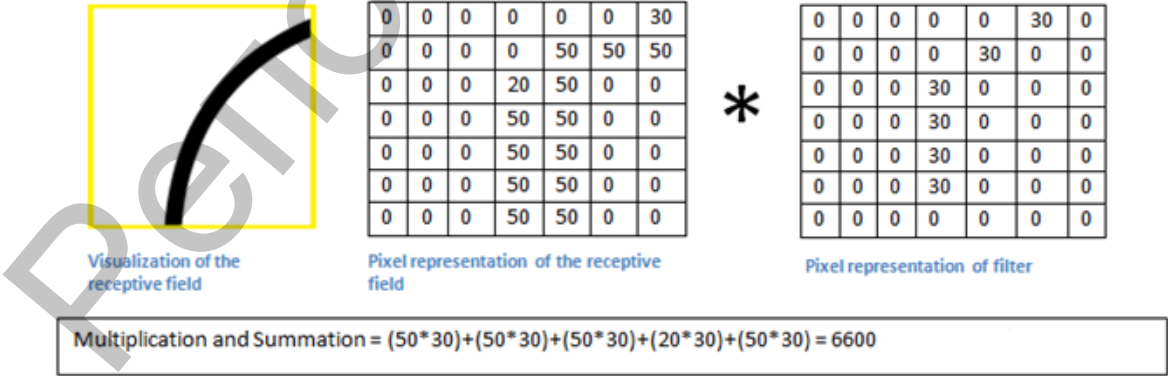


Рисунок 2.5 – Результат применения фильтра (наличие признака)

Если же произвести свертку другой области, не похожую на визуализацию фильтра, то получим маленькое число, что и показано на рис. 2.6. Это будет свидетельствовать о том, что признак не распознан.

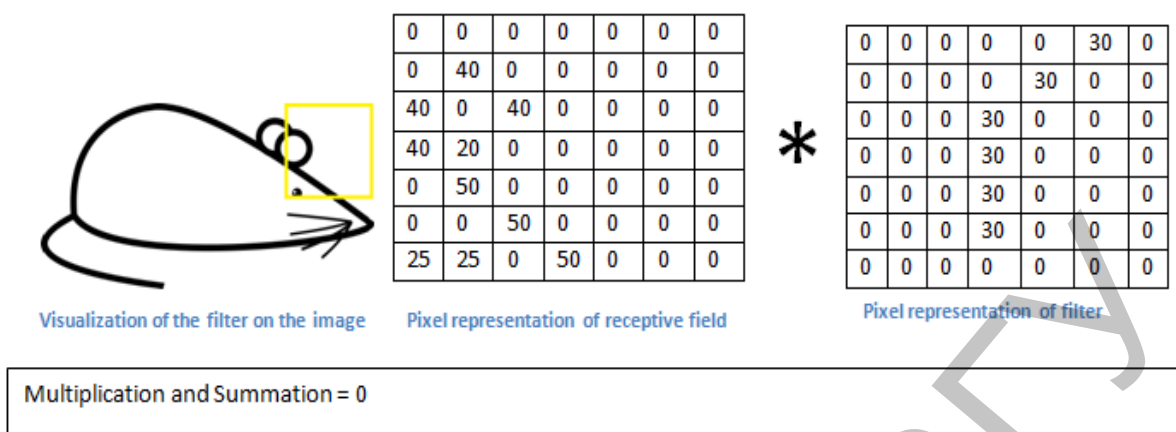


Рисунок 2.6 – Результат применения фильтра (отсутствие признака)

### 2.3 Суть слоя свертки

В свёрточной нейронной сети в операции свёртки используется лишь ограниченная матрица весов небольшого размера, которую «двигают» по всему обрабатываемому слою (в самом начале — непосредственно по входному изображению), формируя после каждого сдвига сигнал активации для нейрона следующего слоя с аналогичной позицией. Процесс формирования первого слоя продемонстрирован на рис. 2.7.

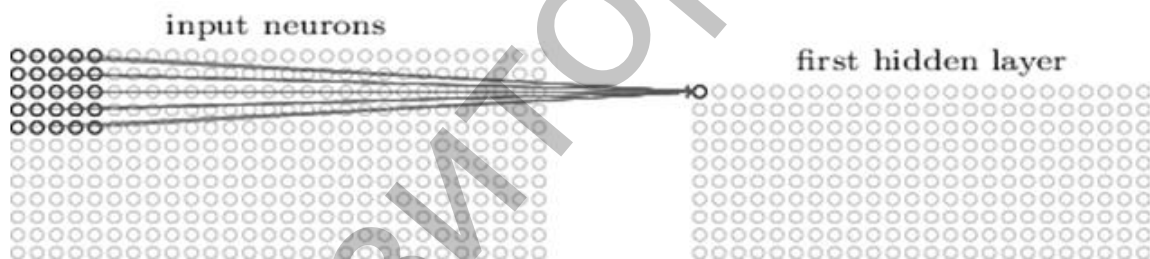
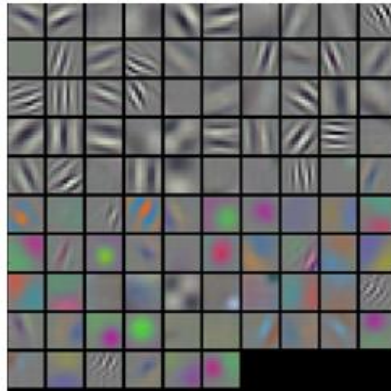


Рисунок 2.7 – Пример свертки с ядром размера 5x5

То есть для различных нейронов выходного слоя используются одна и та же матрица весов, которую также называют ядром свёртки.

Её интерпретируют как графическое кодирование какого-либо признака, например, наличие наклонной линии под определенным углом. Тогда следующий слой, получившийся в результате операции свёртки такой матрицей весов, показывает наличие данного признака в обрабатываемом слое и её координаты, формируя так называемую карту признаков.

В свёрточной нейронной сети присутствует целая гамма таких фильтров, кодирующих элементы изображения (например линии и дуги под разными углами). На рис. 2.8 продемонстрирована визуализация фильтров первого свёрточного слоя некоторой обученной СНС. [6]



Visualizations of filters

Рисунок 2.8 – Визуализация фильтров первого сверточного слоя

При этом ядра свертки не закладываются пользователем заранее, а формируются самостоятельно путём обучения сети классическим методом распространения ошибки.

Проход каждым набором весов формирует свой собственный экземпляр карты признаков, делая нейронную сеть многоканальной (много независимых карт признаков на одном слое).

Также следует отметить, что при прохождении фильтром изображения, фильтр передвигают обычно не на полный шаг (размер матрицы весов фильтра), а на небольшое расстояние. Так, например, при размерности матрицы весов  $5 \times 5$  соответствующий фильтр каждый раз перемещается только на один или два пикселя вместо пяти, чтобы не «перешагнуть» искомый признак.

Особенностью свёрточного слоя является сравнительно небольшое количество параметров, устанавливаемое при обучении. Так например, пусть исходное изображение имеет размерность  $100 \times 100$  пикселей и поступает по трём каналам, а свёрточный слой использует фильтры с ядром  $3 \times 3$  пикселя с выходом на 6 каналов. Для каждого фильтра с таким ядром количество настраиваемых параметров равно количеству элементов в ядре (матрице весов) – 9 параметров.

Тогда общее количество настраиваемых параметров, учитывая сочетания всех каналов, будет равно  $3 \times 6 \times 9 = 152$ , что существенно меньше количества искомых параметров в случае, если бы для той же задачи использовалась полносвязная нейронная сеть.

#### 2.4 Слой подвыборки

*Слой подвыборки* (иначе - слой пулинга, субдискретизации) представляет собой нелинейное уплотнение карты признаков, при этом группа пикселей, (обычно размера  $2 \times 2$ ) уплотняется до одного пикселя, проходя нелинейное математическое преобразование. Чаще всего в качестве такого преобразования используется функция максимума.

Преобразования затрагивают непересекающиеся по сторонам прямоугольники или квадраты, каждый из которых ужимается в один пиксель, при этом выбирается пиксель, имеющий максимальное значение.

На рис. 2.9 представлена подвыборка с функцией максимума и фильтром  $2 \times 2$  с шагом, равным 2.

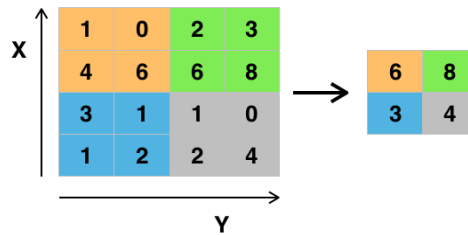


Рисунок 2.9 – Пример подвыборки

В данной архитектуре сети считается, что информация о факте наличия искомого признака важнее точного знания его координат.

Операция подвыборки позволяет существенно уменьшить пространственный объём изображения. Если на предыдущей операции свёртки уже были выявлены некоторые признаки, то для дальнейшей обработки настолько подробное изображение не нужно и поэтому оно уплотняется с помощью подвыборки до менее детализированного.

Это также помогает снизить чувствительность СНС к входящему изображению, а следственно, способствует большей точности распознавания образов при искажении. Сеть становится более инвариантной к масштабу входного изображения.

## 2.5 Обучение и тестирование

*Обучение* как многослойной полносвязной нейронной сети, так и сверточной нейронной сети, происходит с помощью метода обратного распространения ошибки. Данный метод можно разделить на 4 отдельных блока: прямое распространение, функцию потерь, обратное распространение и обновление веса.

Один из способов визуализировать идею минимизации потерь — это трёхмерный график, где веса нейронной сети (пример упрощен для двух весов) это независимые переменные, а зависимая переменная — это потеря. Задача минимизации потерь — отрегулировать веса так, чтобы снизить потерю. Необходимо визуально приблизиться к самой нижней точке чаше-подобного объекта, изображенного на рис. 2.10.

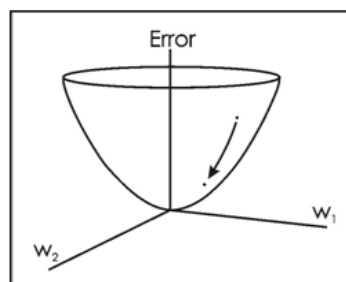


Рисунок 2.10 – Формула вычисления СКО в качестве функции потерь

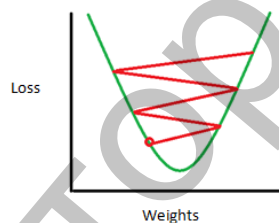
Чтобы добиться этого, необходимо найти производную потери, что значит в рамках изображенного на рис. 2.10 графика рассчитать угловой коэффициент в каждом направлении с учётом весов.

Это математический эквивалент  $dL/dW$ , т.е. производной ошибки по всем весам, где  $L$  – ошибка, а  $W$  – веса определенного слоя. Теперь необходимо выполнить обратное распространение через сеть, которое определяет, какие веса оказали большее влияние на потери найти пути их корректировки с целью уменьшения потерь.

Для этого достаточно лишь обновить все веса так, чтобы они менялись в направлении градиента – то есть вычислить веса по формуле (2.1):

$$w = w_i - \eta \frac{dL}{dW} \quad (2.1)$$

где  $w_i$  – изначальный вес;  $\eta$  – скорость обучения, которая выбирается программистом. Высокая скорость обучения приводит к более крупным шагам в обновлениях веса, при этом образцу может потребоваться меньше времени, чтобы набрать оптимальный набор весов. Однако слишком высокая скорость обучения может привести к очень крупным и недостаточно точным скачкам, которые помешают достижению оптимальных показателей. [6]



**Рисунок 2.11 – Проблема баланса между скоростью обучения и потерями**

Данная проблема между скоростью обучения и потерями проиллюстрирована на рис. 2.11.

Совокупность процесса прямого распространения, вычисления функции потери, обратного распространения и обновления весов обычно называют *одним периодом дискретизации*, или *эпохой*.

Программа будет повторять этот процесс фиксированное количество периодов для каждого тренировочного изображения. После того, как обновление параметров завершится на последнем тренировочном образце, сеть должна быть достаточно хорошо обучена и веса слоёв настроены правильно.

### 3. ЛАБОРАТОРНЫЕ РАБОТЫ

Программа Statistica Neural Networks (SNN) является обширной, мощной и быстрой средой построения и анализа нейросетевых моделей, предоставляющей разнообразные функциональные возможности для работы с очень сложными задачами, включающими не только новейшие архитектуры нейронных сетей и алгоритмы обучения, но также и новые подходы в отборе входных данных и построении сети.

Для построения нейронной сети в программе необходимо выбрать переменные, которые влияют на результат, числовые или номинальные. Переменные других типов следует преобразовать в указанные типы либо объявить незначимыми.

Часть приведенных лабораторных работ взята из [7]. Для изучения работы программного пакета SNN и выполнения практических заданий введены дополнительные файлы Вентилятор.stw и Вуз.sta, представляющие собой общие таблицы с данными параметров и составляющие базу данных для выполнения лабораторных работ № 2-4.

#### Лабораторная работа № 1 ПРОГРАММИРОВАНИЕ НЕЙРОННОЙ СЕТИ

**Цель работы** – изучить математические основы работы нейронной сети и реализовать задачу распознавания образов на языке программирования.

##### Порядок работы

1. Начертите в тетради 10 квадратов размером  $10 \times 10$  клеток. В каждом квадрате от руки нарисуйте десять вариантов написания цифры (рисующая цифра выбирается по последней цифре номера студенческого билета). Старайтесь, чтобы различные изображения немного отличались друг от друга (наклоном или формой), но их размер был приблизительно одинаковым и занимал большую часть квадрата.

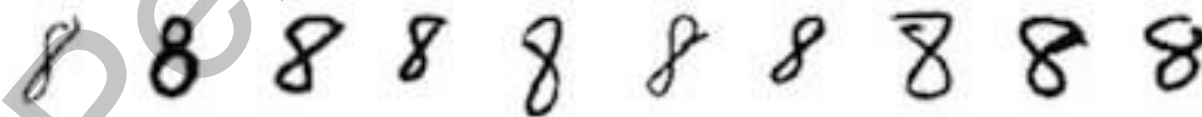


Рисунок 3.1. Образцы

2. Растрируйте полученные изображения, т.е. закрасьте клетки, через которые проходят линии, и оставьте пустыми те, через которые линии не проходят.



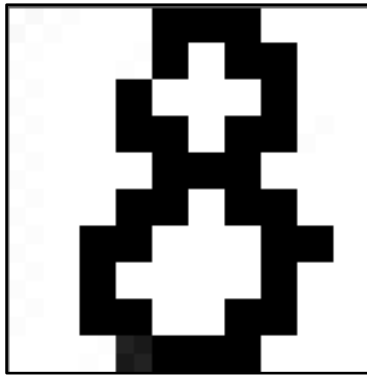


Рисунок 3.2. Растриванный образец

3. Закодируйте изображение каждого квадрата в виде вектора размерностью 100. При этом пустые участки соответствуют значению -1, а закрашенные 1. Например, **эталонный** вектор, описывающий изображение цифры 8 из п.2, имеет координаты:

$$X = \begin{bmatrix} -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 \\ -1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 \\ -1 & -1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 \end{bmatrix}$$

Таким образом, у вас получится десять эталонных изображений

4. Полагая веса каждого эталона  $w_i = 0,1$ , вычислить весовую матрицу

$$W = \sum_{i=1}^{10} w_i X_i^T X_i$$

5. Нарисовать и растривать тестируемую цифру, а затем составить тестируемый вектор  $Y$ .

6. Вычислить выходной вектор  $Y_* = WY$  и нормированный выходной вектор  $Y_0 = f(Y_*)$ . В качестве функции активации взять

$$f(x) = \text{sign}(x) = \begin{cases} 1, & x \geq 0 \\ -1 & x < 0 \end{cases}$$

7. Вычислить расстояния  $r_i = \text{dist}(X_i, Y_0) = \sqrt{\sum_{k=1}^{100} (x_{ik} - y_k)^2}$  от каждого эталонного вектора до нормированного выходного вектора  $Y_0$ . Найти  $r_{\min} = \min\{r_i\}$ .

8. На основании значения  $r_{\min}$  сделать вывод о том, принадлежит ли тестируемое изображение цифры к данному классу.

9\*. Организовать всевозможный перебор всех весов с шагом 0.05.

$$W = \sum_{i=1}^{10} w_i X_i^T X_i$$

Другими словами, нужно составить матрицу для каждого из сочетаний весов [0.55 0.05 0.05 ... 0.05], [0.50 0.10 0.05 ... 0.05], [0.50 0.05 0.10 ... 0.05], ..., [0.50 0.05 0.05 ... 0.10], [0.45 0.05 0.10 ... 0.10], [0.45 0.10 0.05 ... 0.10], ..., [0.45 0.10 0.10 ... 0.05], ..., ..., [0.05 0.05 0.05 ... 0.55].

Для каждого сочетания весов найти расстояние от тестового вектора  $Y_0$  до каждого из эталонных векторов-образцов  $X_i$ . Из полученного множества значений выбрать минимальное расстояние  $r_{\min}$ . Определить экспериментальным путём критическое значение  $r_{\text{крит}}$  такое, что при выполнении неравенства  $r_{\min} < r_{\text{крит}}$  можно утверждать, что распознаваемый вектор  $Y$  принадлежит данному классу образцов  $\{X\}$ .

## Лабораторная работа № 2 ЗНАКОМСТВО С ПРОГРАММОЙ SNN

**Цель работы** – познакомиться с программным продуктом Statistica Neural Networks (SNN), построить нейронную сеть с помощью мастера решений.

### Порядок работы

1. Открыть файл данных Вентилятор.stw (табл.П.1) с помощью команды Файл – Открыть. В этом файле находятся данные о двух типах классах – 1 и 2, наличие и отсутствие перегрева.
2. Выбрать команду **Нейронные сети** в меню **Анализ** для вызова стартовой панели STATISTICA Нейронные сети.

Инструмент:

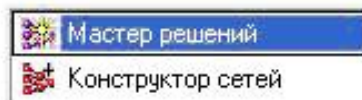


Рисунок 3.3. Выбор инструмента

3. На вкладке **Быстрый** стартовой панели **Нейронные сети** выбрать из списка тип задачи (в данном случае – **Классификация**) и метод решения (в данном случае – **Мастер решений**) и нажать кнопку **ОК** (рис. 3.3). После этого будет отображен стандартный диалог выбора переменных.
4. Выбрать зависимую (выходную) переменную (в данном случае – переменная КЛАСС) (рис. 3.4).

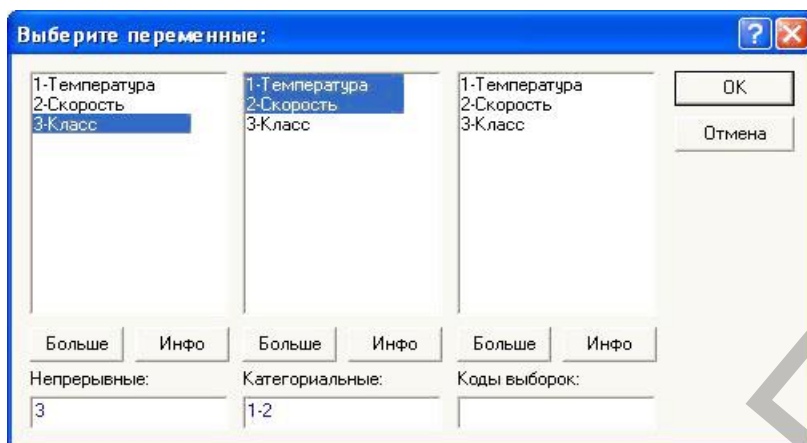


Рисунок 3.4. Входные данные

5. Для отображения **Мастера решений** нажать кнопку **OK** на стартовой панели. На вкладке **Быстрый** (рис. 3.5) отменить выбор опции **Выбрать подмножество независимых переменных**, здесь определены только две независимые переменные, таким образом, обе переменные будут использоваться в качестве входов для всех проверяемых нейронных сетей. В группе **Длительность анализа** находятся опции, определяющие время, которое **Мастер решений** потратит на поиск эффективной нейронной сети. Чем дольше **Мастер решений** будет работать, тем эффективней будет найденное решение. Например, установить 25 сетей.

6. Ввести число 10 для сохранения сетей, чтобы **Мастер решений** сохранил только 10 наилучших вариантов сетей.

Вкладка **Мастер решений – Быстрый** будет иметь вид, отображенный на рис. 3.5.

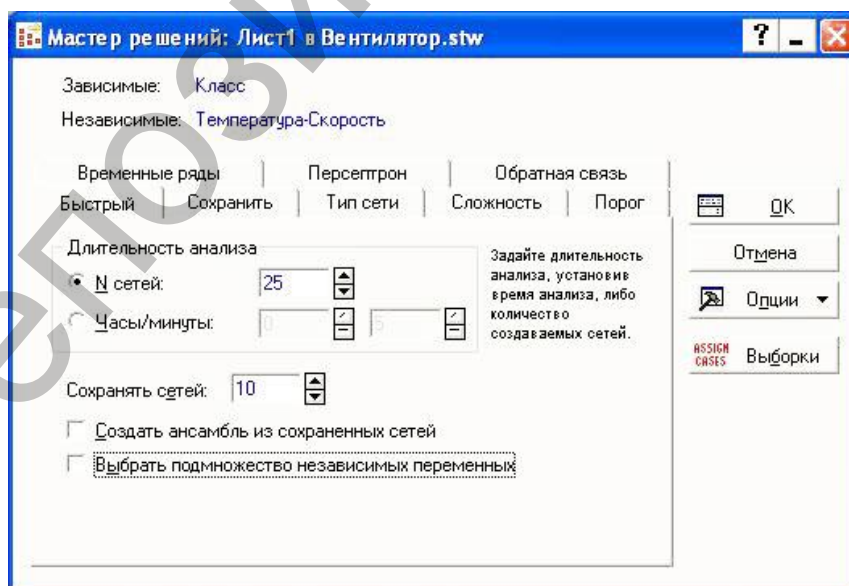


Рисунок 3.5. Настройки для проведения анализа

Нажать кнопку **ОК**, чтобы **Мастер решений** начал построение нейронных сетей. После этого будет отображен диалог **Идет обучение (Мастер решений)**. Каждый раз при обнаружении улучшенной нейронной сети будет добавлена новая строка в информационную таблицу. Кроме этого, в нижней части окна отображается время работы и процент выполненного задания. Если на протяжении длительного времени не произошло никаких улучшений, то нажать кнопку **Готово** в диалоге **Идет обучение**, чтобы завершить процесс поиска сетей. После окончания поиска будет отображен диалог **Результаты**, содержащий информацию о найденных сетях для дальнейшего анализа (рис. 3.6).

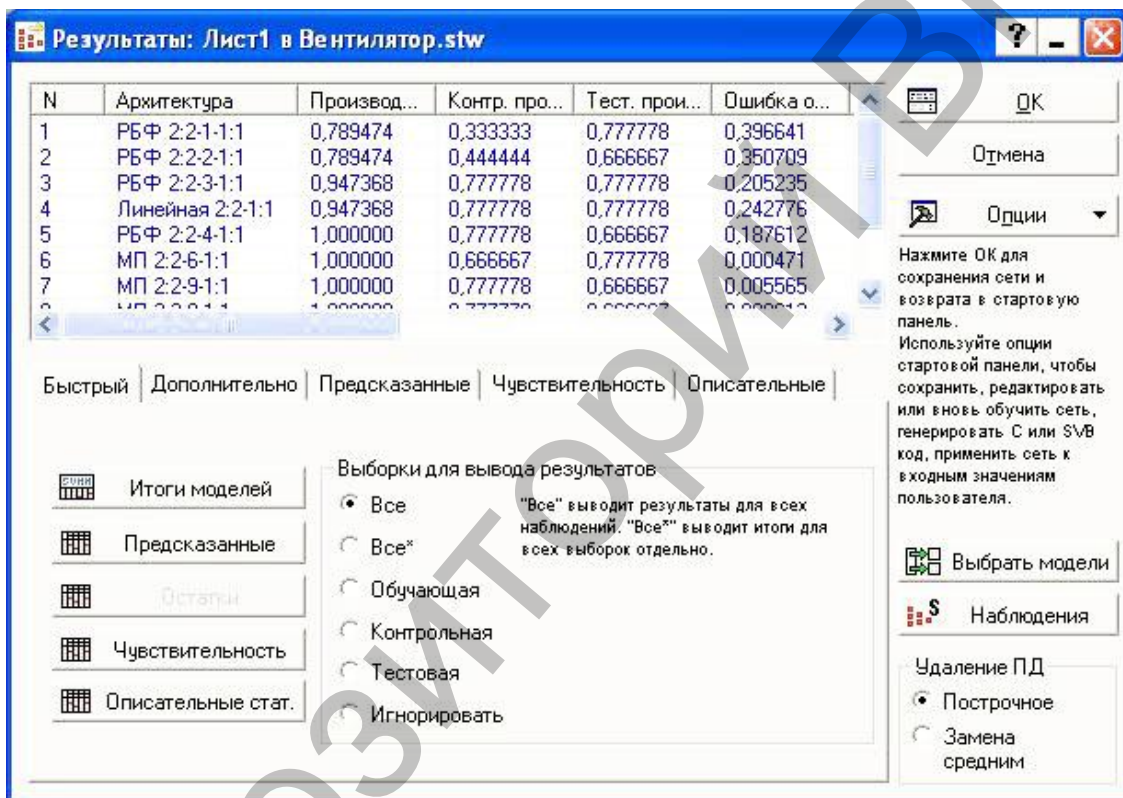


Рисунок 3.6. Результаты обучения

7. Нажать кнопку **Описательные стат.** на вкладке **Быстрый** в диалоге **Результаты**, чтобы отобразить две итоговые таблицы: Классификация и Матрица ошибок.

В таблице классификаций (рис. 3.7) представлена полная информация о решении соответствующей задачи. В этой таблице для каждого выходного класса, предсказанной каждой моделью, существует несколько столбцов. Например, столбец, обозначенный КЛАСС.1.11, соответствует предсказаниям модели 1 в классе ПЕРЕГРЕВА для переменной КЛАСС. В первой строке приводится информация о количестве наблюдений различных типов перегревов в файле данных. Во второй (третьей) строке отображаются данные (для каждого класса) о количестве правильно (неправильно) классифицированных наблюдений.

В четвертой строке приводятся «неизвестные» наблюдения. Матрица ошибок обычно используется в задачах с несколькими выходными классами.

8. Для отображения итоговой статистики необходимо открыть **Анализ** (кнопка **Результаты** в строке **Анализа** или команда **Продолжить** в меню **Анализ**). В группе **Выборки для вывода результатов** выбрать опцию **Все** (отдельно). Затем нажать кнопку **Описательные статистики**. Итоговая таблица классификаций разделена на четыре части. Заголовки столбцов имеют различные префиксы: **О**, **К**, **Т** и **И**, которые соответствуют обучающей, контрольной, тестовой и игнорируемой выборкам соответственно. По умолчанию, наблюдения разделены на три подмножества в отношении 2:1:1. Таким образом, выделено 50 обучающих наблюдений, 25 контрольных наблюдений и 25 проверочных наблюдений. Результаты работы нейронной сети на этих множествах практически совпадают, то есть качество нейронной сети можно считать приемлемым.

	Классификация (11-20 ) (Лист1 в Вентилятор.stw)				
	Класс.1.11	Класс.2.11	Класс.1.12	Класс.2.12	Класс.1.13
<b>Всего</b>	17,00000	20,00000	17,00000	20,00000	17,00000
Правильно	11,00000	13,00000	13,00000	14,00000	17,00000
Ошибочно	6,00000	7,00000	4,00000	6,00000	0,00000
Неизвестно	0,00000	0,00000	0,00000	0,00000	0,00000
% правильных	64,70588	65,00000	76,47059	70,00000	100,00000
% ошибочных	35,29412	35,00000	23,52941	30,00000	0,00000
% неизвестно	0,00000	0,00000	0,00000	0,00000	0,00000

Рисунок 3.7. Таблица классификаций

9. Для завершения **Анализа** нажать кнопку **ОК** в диалоге **Результаты**. На стартовой панели при нажатии на кнопку **Отмена** все построенные нейронные сети будут удалены. Сохранять нейронные сети необходимо для того, чтобы быстрее провести обучение нейронных сетей, соответственно перед этим найти сеть с наилучшей производительностью, и далее построенные нейронные сети сохраняются для дальнейшего использования. Для сохранения нейронной сети выбрать вкладку **Сети/Ансамбли** и нажать кнопку **Сохранить файл сети как...** (файл имеет расширение .snn).

### Задания

1. Построить и обучить нейронную сеть с помощью **Мастера решений** для автоматизации диагностики автомобиля, определяющей необходимость капитального ремонта двигателя на основе следующих параметров: компрессия двигателя, давление масла, расход бензина.

2. Ввести исходные данные в соответствии с табл. 3.1, конкретные значения переменных получить у преподавателя.

3. Построить нейронную сеть в соответствии с настройками:
  - тип задачи: классификация;
  - инструмент: мастер решений;
  - количество сетей: 25;
  - сохранять сетей: 10.
4. Сохранить нейронные сети различных типов с разными показателями производительности и сложности. Определить наилучшую сеть.
5. Сделать анализ построения нейронной сети и отразить в отчете.

Номер наблюдения	Давление масла, кгс/см <sup>2</sup>	Расход бензина, % от нормы	Компрессия двигателя, кгс/см <sup>2</sup>	Вывод
1	0.84-0.91	105	10.3	Не треб.
2	0.56-0.85	127	10.2	Не треб.
3	0.33-0.55	144	9.5	Кап.рем.
4	0.24-0.46	141	10.7	Кап.рем.
5	0.23-0.44	113	9.8	Не треб.
6	0.18-0.79	139	9.4	Кап.рем.
7	0.67-0.84	112	10.1	Не треб.
8	0.75-0.90	140	9.5	Кап.рем.

Таблица 3.1 - Основные данные для определения необходимости капитального ремонта

### Лабораторная работа №3 ТЕСТИРОВАНИЕ НЕЙРОСЕТИ ДЛЯ РЕШЕНИЯ ЗАДАЧИ КЛАССИФИКАЦИИ

**Цель работы** – построить, обучить и провести тестирование нейросети для решения задачи классификации, а также определить зависимость результатов предсказаний от длительности анализа при построении и обучении нейронной сети.

#### Порядок работы

1. Открыть файл данных **Вентилятор.stw** (см.табл.П.1). В этом файле находятся данные о двух типах классах – 1 и 2, наличие и отсутствие перегрева.
2. Выбрать команду **Нейронные сети** в меню **Анализ** для вызова стартовой панели STATISTICA Нейронные сети.
3. На вкладке **Быстрый** стартовой панели **Нейронные сети** выбрать из списка тип задачи (в данном случае – **Классификация**) и метод решения (в данном случае – **Мастер решений**) и нажать кнопку **ОК**.
4. После этого будет отображен стандартный диалог выбора переменных. Выбрать зависимую (выходную) переменную (в данном случае – переменная КЛАСС, которая содержит коды для двух типов перегрева) и за-

тем независимые переменные (в данном случае – переменные **ТЕМПЕРАТУРА** и **СКОРОСТЬ**, которые содержат данные о температуре и скорости вращения вентилятора).

5. Нажать кнопку **ОК** на стартовой панели для отображения диалога **Мастер решений**.

6. На вкладке **Быстрый** нужно отменить выбор опции **Выбрать подмножество независимых переменных**, так как определены только две независимые переменные. Таким образом, обе переменные будут использоваться в качестве входов для всех проверяемых нейронных сетей. В группе **Длительность анализа** находятся опции, определяющие время, которое **Мастер решений** потратит на поиск эффективной нейронной сети. Чем дольше **Мастер решений** будет работать, тем эффективней будет найденное решение. Можно ограничить время работы в терминах количество создаваемых и проверяемых нейронных сетей (опция **N сетей**) или точно указать время работы (опция **Часы/минуты**). Для решения текущей задачи в опции **N сетей** необходимо указать число итераций, равное 25. **Мастер решений** может исследовать очень большое количество нейронных сетей. Обычно рекомендуется изучить несколько наилучших вариантов сетей. Таким образом можно сравнить производительность нескольких различных типов нейронных сетей, поскольку некоторые сети могут иметь меньшую производительность, но им стоит отдать предпочтение по другим причинам (например, меньший размер или меньшее число входов). По умолчанию, программа сохраняет нейронные сети различных типов с различными показателями производительности и сложности для того, чтобы можно было в итоге выбрать наилучшую сеть самостоятельно.

7. Ввести число 10 для сохранения сетей, чтобы **Мастер решений** сохранил только 10 наилучших вариантов сетей. Теперь нажать кнопку **ОК**, чтобы **Мастер решений** начал построение нейронных сетей. После этого будет отображен диалог **Идет обучение (Мастер решений)**. Каждый раз при обнаружении улучшенной нейронной сети будет добавлена новая строка в информационную таблицу. Кроме этого, в нижней части окна отображается время работы и процент выполненного задания. Если на протяжении длительного времени не произошло никаких улучшений, то можно нажать кнопку **Готово** в диалоге **Идет обучение**, чтобы завершить процесс поиска сетей.

8. После окончания поиска будет отображен диалог **Результаты**, который содержит информацию о найденных сетях и предоставляет функции для дальнейшего анализа.

9. На вкладке **Быстрый** в диалоге **Результаты** нужно выбрать опцию **Все** в группе **Выборки для вывода результатов**, затем нажать кнопку **Предсказанные**. Программа создает таблицу с предсказаниями для множества данных. В эту таблицу включаются наблюдаемые значения из мно-

жества данных. В таблице отображается наблюдаемое значение (КЛАСС) в первом столбце, затем соответствующие предсказания нейронной сети (КЛАСС.n, где n – идентификатор выбранной сети – например, КЛАСС.4 обозначает нейронную сеть с идентификатором 4). Если выбрать несколько моделей, то для наблюдаемого значения будет один столбец, а для различных предсказаний будет выделено несколько столбцов.

На основе стандартного выбора в таблице отображаются предсказания, сделанные нейронной сетью – 1 или 2. Нейронная сеть работает с числовой информацией, а предсказания строятся на основе встроенных постпроцессируемых функций, которые соответствуют каждой сети. Подобное предсказание осуществляется за счет сравнения уровня активации выходного нейрона с доверительным уровнем. После этого проводится сопоставление соответствующего класса. В этом процессе также учитываются некоторые доверительные пороги (устанавливаемые автоматически **Мастером решений**).

10. Выбрать вкладку **Предсказанные** в диалоге **Результаты**. В группе **Прогноз** отменить выбор опции **Предсказанные значения** и выбрать опцию **Уровни доверия**. Нажать кнопку **Предсказанные**. После этого на экране появятся доверительные уровни для каждого наблюдения (все значения будут находиться в интервале между 0 и 1), где нижний график построен для перегрева вентилятора **1**, а верхний – для перегрева **2**. Оказывается, что нейронная сеть очень хорошо различает большинство типов перегревов.

11. Нейронная сеть предсказывает класс для каждой комбинации температуры и скорости вращения. Можно отобразить это предсказание в виде поверхности отклика в трехмерном пространстве (рис. 3.8, 3.9), где высота поверхности обозначает предсказанный класс. Выбрать вкладку **Дополнительно** в диалоге **Результаты** и нажать кнопку **Поверхность отклика**, чтобы отобразить диалог **Поверхность отклика**. Проверить, что в полях **Независимая (Ось X)** и **Независимая (Ось Y)** выбраны «температура» и «скорость» соответственно. Нажать кнопку **Поверхность отклика** на вкладке **Быстрый**, чтобы отобразить поверхность отклика. На поверхности отклика для КЛАССА 2 видно плато в средних температурах, соответствующее предсказанию перегрева КЛАССА 2. Нижние области соответствуют предсказанию перегрева КЛАССА 1.

Поверхность отклика для класса 1 является зеркальным отображением для класса 2.



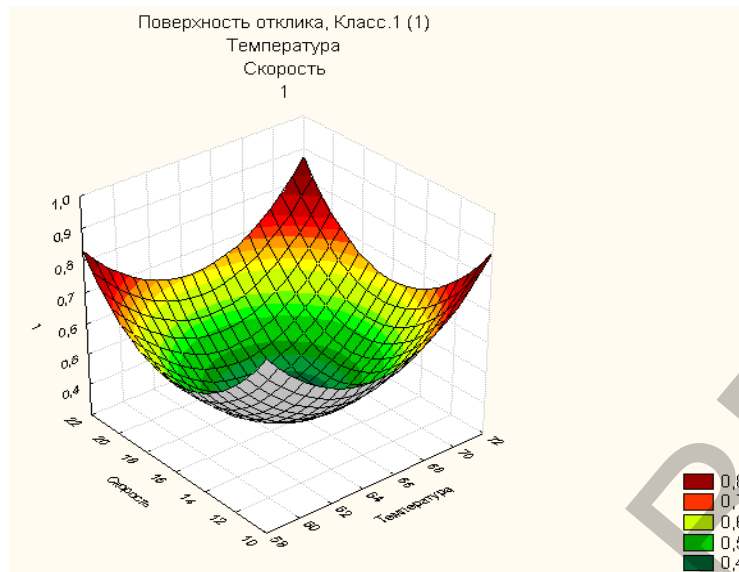


Рисунок 3.8 - Поверхность отклика – Класс 1

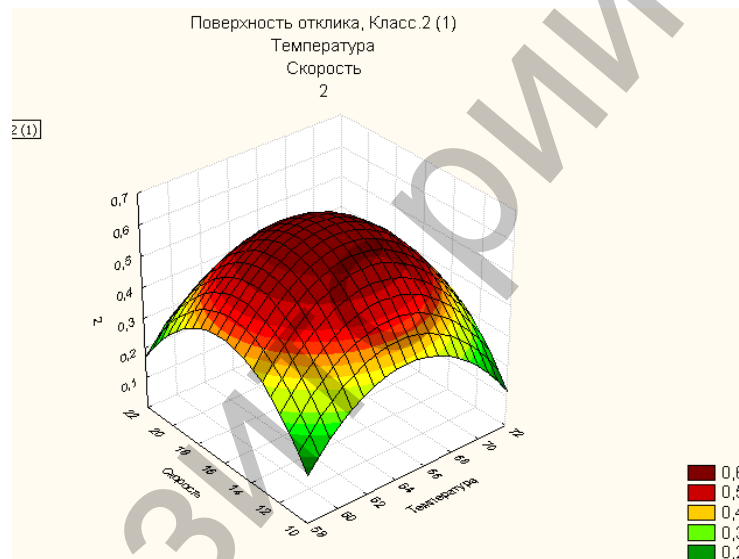


Рисунок 3.9 - Поверхность отклика – Класс 2

12. Обучив нейронную сеть, можно использовать ее для выполнения предсказаний на основе новых данных. Для этого нажать кнопку **Наблюдение пользователя** на вкладке **Дополнительно** в диалоге **Результаты** для отображения диалога **Прогноз наблюдения пользователя**. На вкладке **Быстрый** в этом диалоге содержится таблица с текущим входом. Чтобы запустить существующий вход, нажать кнопку **Прогнать текущее**. Чтобы изменить это значение, нажать кнопку **Значение пользователя**. После этого будет отображена таблица. Нужно ввести температуру и скорость в соответствующие ячейки таблицы и нажать кнопку **ОК**. После этого будут отображены новые пользовательские значения на вкладке **Быстрый**. Чтобы использовать эти значения, нажать кнопку **Прогнать текущее**. Пустая ячейка свидетельствует о пропущенном значении, и программа автоматически заменит подходящее значение при запуске сети.

Для проверки нужно ввести температуру и скорость и нажать кнопку **Выполнить**. К итоговым данным будет добавлена одна строка, содержащая предсказания. Для завершения анализа надо закрыть диалог **График отклика** и нажать кнопку **ОК** в диалоге **Результаты**.

### Задания

1. Построить и обучить нейронную сеть с помощью **Конструктора сетей**, автоматизирующую определение типа транспортного средства на основе представленных данных для четырех типов автомашин (Легковой а\м, Грузовой а\м, Джип, Микроавтобус). В исходных данных содержатся измерения четырех переменных (Клиренс, Масса, Радиус колес, Количество колес).

2. Ввести исходные данные с количеством наблюдений 30 в соответствии со следующими условиями, указанными в табл. 3.2, конкретные значения переменных получить у преподавателя.

Тип автомашины	Клиренс, градус	Масса, кг	Радиус дисков, дюйм	Количество колес, шт.
Легковой а\м	5–10	300–1500	10–13	4
Минивен	8–13	900–3000	12–15	4–6
Джип	10–20	2000–3500	12–15	4–6
Грузовой а\м	15–35	3500–6000	14–18	4–8

Таблица 3.2 - Основные данные для определения типа транспортного средства

3. Построить нейронную сеть в соответствии с настройками:

- тип задачи: классификация;
- инструмент: мастер решений;
- количество сетей: 25;
- сохранять сетей: 10.

4. Вывести таблицу с предсказаниями для множества данных. Интерпретировать полученные результаты.

5. Отобразить **Поверхность отклика**, проанализировать предсказанный класс.

6. Изменить наблюдения пользователя и выявить связь между применяемыми данными и полученным результатом.

7. Сделать анализ построения нейронной сети и отразить в отчете.

## Лабораторная работа № 4 НЕЙРОННЫЕ СЕТИ С РАДИАЛЬНОЙ БАЗИСНОЙ ФУНКЦИЕЙ

**Цель работы** – научиться определять количество скрытых слоев и элементов для построения эффективной нейронной сети с максимальной производительностью.

### Порядок работы

1. Откройте файл **Вуз.sta** (табл.П.2).
2. Выбрать команду **Нейронные сети**.
3. Выбрать опцию **Классификация** в группе **Тип задачи**, а затем выбрать опцию **Конструктор сетей**.
4. Нажать кнопку **Переменные** и выбрать переменную **ИТОГ** в поле **Категориальные выходные**, а переменные **Рус.яз\_ЕГЭ**, **Матем\_ЕГЭ**, **Химия** и **Ин.яз** в группе **Непрерывные входные**.
5. В диалоге **Конструктор сетей** выбрать **Радиальная базисная функция (RBF-сети)**, а затем **обучение (ОК)**.
6. На вкладке **Быстрый** выбрать опцию **К-средних** в группе **Задание центров К-Средних** (рис.3.10). Таким образом, центры кластеров будут присвоены радиальным элементам.

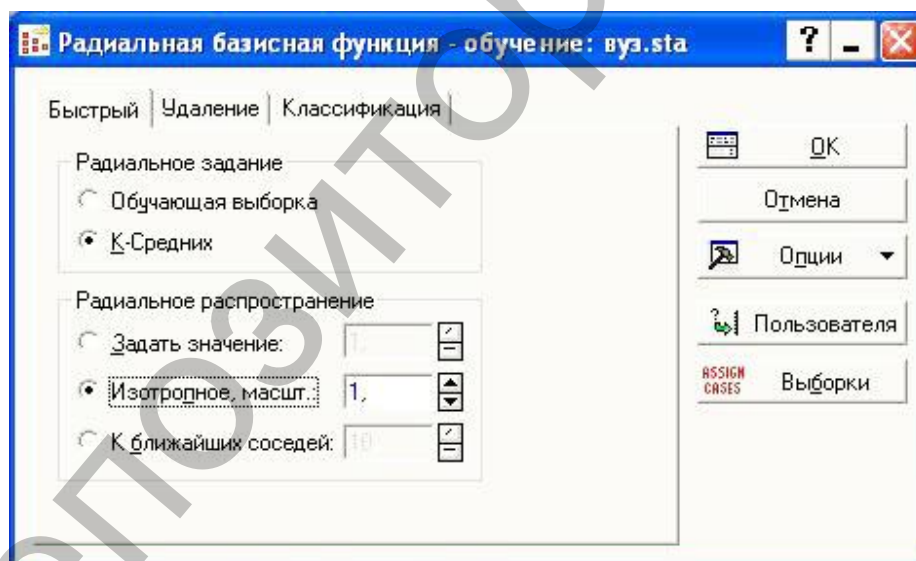


Рисунок 3.10 - Настройка для обучения сети РБФ

7. Выбрать опцию **Изотропное, масштаб**, чтобы присвоить соответствующие отклонения, основанные на нескольких элементах в обучающей выборке.
8. Нажать кнопку **ОК** для обучения сети.
9. На вкладке **Дополнительно** (предварительно завершить анализ – кнопка **ОК**) выбрать опцию **Создать ансамбль** (рис. 3.11) из сохраненных сетей, где тип – **Выход**.

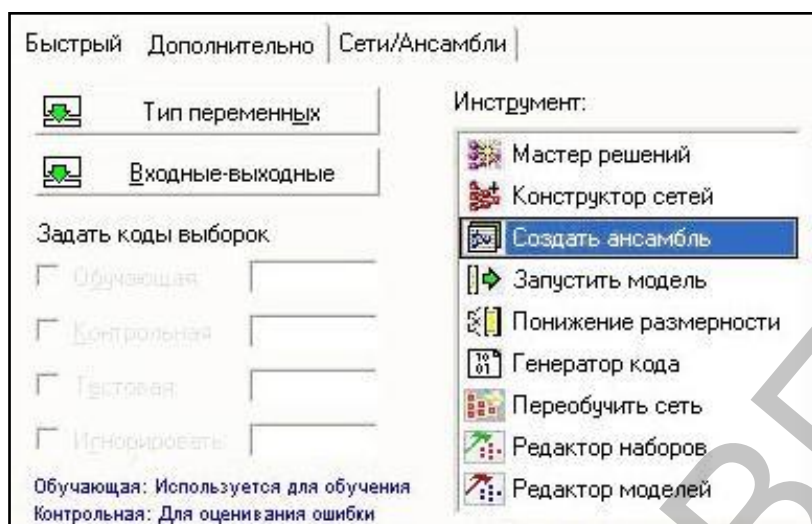


Рисунок 3.11 - Выбор инструмента «Создать ансамбль»

10. Для получения ансамбля нажать **Выбрать Сети**, затем **Выбрать все** и **ОК**.

11. В итоговом списке находится простая модель с **Архитектурой** вида, например, **Выход 4:[5]:1** в диалоге **Результаты**, т.е. итоговый ансамбль с четырьмя входами, одним выходов и пятью нейронными сетями (рис. 3.12).

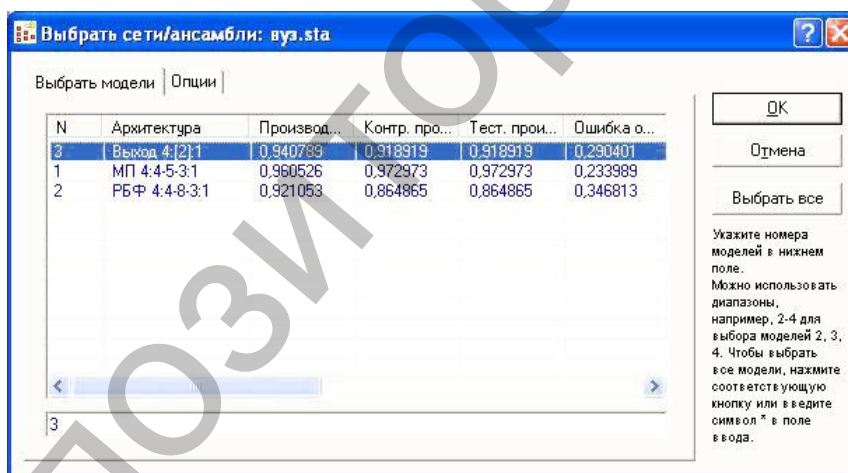


Рисунок 3.12 - Архитектура ансамбля

12. Для применения сети отдельно от ансамбля в диалоге **Результаты** нужно нажать кнопку **Выборить модели**.

13. В диалоге **Выборить сети/ансамбли** на вкладке **Опции** выбрать **Выборить ансамбль и сети** в группе **При выделении ансамбля** и нажать кнопку **ОК**.

14. В списке в диалоге **Результаты** будет отображен ансамбль, а также его составляющие сети.

15. Нажать кнопку **Предсказанные**. В итоговой таблице будут отображены прогнозы для всех сетей и общего ансамбля (рис.3.13).

16. Для возврата на стартовую панель **Нейронные сети** нужно нажать кнопку **ОК**.

	Итого	Итого.3
12	ПЕРЕВЕЛСЯ	
30	ПЕРЕВЕЛСЯ	ПЕРЕВЕЛСЯ
38	ПЕРЕВЕЛСЯ	ПЕРЕВЕЛСЯ
66	ПЕРЕВЕЛСЯ	ПЕРЕВЕЛСЯ
129	ПЕРЕВЕЛСЯ	ПЕРЕВЕЛСЯ
131	ПОСТУПИЛ	ПОСТУПИЛ
138	НЕ ПОСТУПИЛ	НЕ ПОСТУПИЛ
140	ПЕРЕВЕЛСЯ	ПЕРЕВЕЛСЯ
143	НЕ ПОСТУПИЛ	НЕ ПОСТУПИЛ

Рисунок 3.13 - Таблица предсказаний

### Задания

1. Построить и обучить нейронную сеть с помощью конструктора, автоматизирующую выбор банка для выдачи кредита заемщику на основе представленных данных для шести видов банков (Сбербанк, Омск-Банк, ОТП-банк, ИТ-банк, ВТБ, КМБ-банк). В исходных данных содержатся четыре переменные (срок кредита, процентные ставки, сумма кредита, заработная плата заемщика).

2. Введите исходные данные с количеством наблюдений – 50 (конкретные значения переменных получить у преподавателя) в соответствии со следующими условиями (см.табл. 3.2).

Наименование банка	Срок кредита	Процентная ставка	Сумма кредита, руб.	Зарботная плата заемщика
Сбербанк	1–10	10–25	40000 – 600000	5000 – 33000
КМБ-банк	10–30	10–25	500000 – 3000000	15000 – 35000
ОТП-банк	10–30	10–25	100000 – 3500000	8000 – 40000
ВТБ	20–30	10–25	900000 – 4000000	18000 – 60000
Омск-банк	15–25	10–25	700000 – 2500000	15000 – 50000

Таблица 3.2 - Основные данные для определения типа банка

3. Построить нейронную сеть в соответствии с настройками:
- тип задачи: классификация;
  - инструмент: конструктор сетей;

- тип сети: многослойный персептрон;
  - число скрытых слоев: 1;
  - число элементов в скрытом слое: 5.
4. Провести обучение сети.
  5. Проверить производительность сети.
  6. Сравнить производительность полученной сети с производительностями предыдущих сетей.
  7. Переобучить сеть при необходимости.
  8. Предотвратить зашумление графика.
  9. Определить производительность нейронной сети.
  10. Интерпретировать результаты после обучения.
  11. Завершить анализ.
  12. Изменить тип сети: радиальная базисная функция,
  13. Радиальное задание: К-Среднее, Радиальное распространение: Изотропное, Масштаб: 1.
  14. Создать ансамбль.
  15. Вывести Архитектуру сети и интерпретировать ее.
  16. Осуществить выборку модели из ансамбля с наилучшими показателями.
  17. Сохранить нейронные сети на диске.
  18. Оформить отчет о выполненной работе.

## ПРИЛОЖЕНИЕ

Таблица П.1 - Данные файла Вентилятор.stw

Температура, гр.	Частота вращения, мин <sup>-1</sup>	Класс
59,0	1350	1
59,5	2900	1
60,0	3350	2
60,5	1395	1
61,0	3650	2
61,0	3810	2
61,5	2940	1
61,5	2850	1
62,0	3650	2
63,0	3650	2
63,5	2940	1
64,0	3810	2
64,5	3810	2
65,0	3950	2
65,0	3950	2

65,0	1810	1
65,5	3950	2
65,5	2040	1
65,5	1450	1
66,0	4540	2
66,0	4540	2
66,0	1625	1
66,5	1625	1
66,5	1810	1
66,5	1810	1
67,0	4625	2
67,5	1810	1
68,0	4625	2
68,5	1000	1
69,0	4810	2
69,0	4810	2
69,5	4810	2
69,5	1000	1
70,0	4950	2
70,5	4950	2
71,0	4950	2
71,5	1060	1

Таблица П.2 - Данные файла Вуз.sta для определения поступления в вуз абитуриента

Рус. язык	Математика	Химия	Ин. язык	Итог
5	3,3	1,4	0,2	Не поступил
6,4	2,8	5,6	2,2	Поступил
6,5	2,8	4,6	1,5	Перевелся
6,7	3,1	5,6	2,4	Поступил
6,3	2,8	5,1	1,5	Поступил
4,6	3,4	1,4	0,3	Не поступил
6,9	3,1	5,1	2,3	Поступил
6,2	2,2	4,5	1,5	Перевелся
5,9	3,2	4,8	1,8	Перевелся
4,6	3,6	1	0,2	Не поступил
6,1	3	4,6	1,4	Перевелся
6	2,7	5,1	1,6	Перевелся
6,5	3	5,2	2	Поступил
5,6	2,5	3,9	1,1	Перевелся
6,5	3	5,5	1,8	Поступил
5,8	2,7	5,1	1,9	Поступил

6,8	3,2	5,9	2,3	Поступил
5,1	3,3	1,7	0,5	Не поступил
5,7	2,8	4,5	1,3	Перевелся
6,2	3,4	5,4	2,3	Поступил
7,7	3,8	6,7	2,2	Поступил
6,3	3,3	4,7	1,6	Перевелся
6,7	3,3	5,7	2,5	Поступил
7,6	3	6,6	2,1	Поступил
4,9	2,5	4,5	1,7	Поступил
5,5	3,5	1,3	0,2	Не поступил
6,7	3	5,2	2,3	Поступил
7	3,2	4,7	1,4	Перевелся
6,4	3,2	4,5	1,5	Перевелся
6,1	2,8	4	1,3	Перевелся
4,8	3,1	1,6	0,2	Не поступил
5,9	3	5,1	1,8	Поступил
5,5	2,4	3,8	1,1	Перевелся
6,3	2,5	5	1,9	Поступил
6,4	3,2	5,3	2,3	Поступил
5,2	3,4	1,4	0,2	Не поступил
4,9	3,6	1,4	0,1	Не поступил
5,4	3	4,5	1,5	Перевелся
7,9	3,8	6,4	2	Поступил
<b>Рус. язык</b>	<b>Математика</b>	<b>Химия</b>	<b>Ин. язык</b>	<b>Итог</b>
4,4	3,2	1,3	0,2	Не поступил
6,7	3,3	5,7	2,1	Поступил
5	3,5	1,6	0,6	Не поступил
5,8	2,6	4	1,2	Перевелся
4,4	3	1,3	0,2	Не поступил
7,7	2,8	6,7	2	Поступил
6,3	2,7	4,9	1,8	Поступил
4,7	3,2	1,6	0,2	Не поступил
5,5	2,6	4,4	1,2	Перевелся
7,2	3,2	6	1,8	Поступил
4,8	3	1,4	0,3	Не поступил
5,1	3,8	1,6	0,2	Не поступил
6,1	3	4,9	1,8	Поступил
4,8	3,4	1,9	0,2	Не поступил
5	3	1,6	0,2	Не поступил
5	3,2	1,2	0,2	Не поступил
6,1	2,6	5,6	1,4	Поступил
6,4	2,8	5,6	2,1	Поступил



4,3	3	1,1	0,1	Не поступил
5,8	4	1,2	0,2	Не поступил
5,1	3,8	1,9	0,4	Не поступил
6,7	3,1	4,4	1,4	Перевелся
6,2	2,8	4,8	1,8	Поступил
4,9	3	1,4	0,2	Не поступил
5,1	3,5	1,4	0,2	Не поступил
5,6	3	4,5	1,5	Перевелся
5,8	2,7	4,1	1	Перевелся
5	3,4	1,6	0,4	Не поступил
4,6	3,2	1,4	0,2	Не поступил
6	2,9	4,5	1,5	Перевелся
5,7	2,6	3,5	1	Перевелся
5,7	4,4	1,5	0,4	Не поступил
5	3,6	1,4	0,2	Не поступил
7,7	3	6,1	2,3	Поступил
6,3	3,4	5,6	2,4	Поступил
5,8	2,7	5,1	1,9	Поступил
5,7	2,9	4,2	1,3	Перевелся
7,2	3	5,8	1,6	Поступил
5,4	3,4	1,5	0,4	Не поступил
5,2	4,1	1,5	0,1	Не поступил
7,1	3	5,9	2,1	Поступил
<b>Рус. язык</b>	<b>Математика</b>	<b>Химия</b>	<b>Ин. язык</b>	<b>Итог</b>
6,4	3,1	5,5	1,8	Поступил
6	3	4,8	1,8	Поступил
6,3	2,9	5,6	1,8	Поступил
4,9	2,4	3,3	1	Перевелся
5,6	2,7	4,2	1,3	Перевелся
5,7	3	4,2	1,2	Перевелся
5,5	4,2	1,4	0,2	Не поступил
4,9	3,1	1,5	0,2	Не поступил
7,7	2,6	6,9	2,3	Поступил
6	2,2	5	1,5	Поступил
5,4	3,9	1,7	0,4	Не поступил
6,6	2,9	4,6	1,3	Перевелся
5,2	2,7	3,9	1,4	Перевелся
6	3,4	4,5	1,6	Перевелся
5	3,4	1,5	0,2	Не поступил
4,4	2,9	1,4	0,2	Не поступил
5	2	3,5	1	Перевелся
5,5	2,4	3,7	1	Перевелся

5,8	2,7	3,9	1,2	Перевелся
4,7	3,2	1,3	0,2	Не поступил
4,6	3,1	1,5	0,2	Не поступил
6,9	3,2	5,7	2,3	Поступил
6,2	2,9	4,3	1,3	Перевелся
7,4	2,8	6,1	1,9	Поступил
5,9	3	4,2	1,5	Перевелся
5,1	3,4	1,5	0,2	Не поступил
5	3,5	1,3	0,3	Не поступил
5,6	2,8	4,9	2	Поступил
6	2,2	4	1	Перевелся
7,3	2,9	6,3	1,8	Поступил
6,7	2,5	5,8	1,8	Поступил
4,9	3,1	1,5	0,1	Не поступил
6,7	3,1	4,7	1,5	Перевелся
6,3	2,3	4,4	1,3	Перевелся
5,4	3,7	1,5	0,2	Не поступил
5,6	3	4,1	1,3	Перевелся
6,3	2,5	4,9	1,5	Перевелся
6,1	2,8	4,7	1,2	Перевелся
6,4	2,9	4,3	1,3	Перевелся
5,1	2,5	3	1,1	Перевелся
5,7	2,8	4,1	1,3	Перевелся
<b>Рус. язык</b>	<b>Математика</b>	<b>Химия</b>	<b>Ин. язык</b>	<b>Итог</b>
6,5	3	5,8	2,2	Поступил
6,9	3,1	5,4	2,1	Поступил
5,4	3,9	1,3	0,4	Не поступил
5,1	3,5	1,4	0,3	Не поступил
7,2	3,6	6,1	2,5	Поступил
6,5	3,2	5,1	2	Поступил
6,1	2,9	4,7	1,4	Перевелся
5,6	2,9	3,6	1,3	Перевелся
6,9	3,1	4,9	1,5	Перевелся
6,4	2,7	5,3	1,9	Поступил
6,8	3	5,5	2,1	Поступил
5,5	2,5	4	1,3	Перевелся
4,8	3,4	1,6	0,2	Не поступил
4,8	3	1,4	0,1	Не поступил
4,5	2,3	1,3	0,3	Не поступил
5,7	2,5	5	2	Поступил
5,7	3,8	1,7	0,3	Не поступил
5,1	3,8	1,5	0,3	Не поступил

5,5	2,3	4	1,3	Перевелся
6,6	3	4,4	1,4	Перевелся
6,8	2,8	4,8	1,4	Перевелся
5,4	3,4	1,7	0,2	Не поступил
5,1	3,7	1,5	0,4	Не поступил
5,2	3,5	1,5	0,2	Не поступил
5,8	2,8	5,1	2,4	Поступил
6,7	3	5	1,7	Перевелся
6,3	3,3	6	2,5	Поступил
5,3	3,7	1,5	0,2	Не поступил
5	2,3	3,3	1	Перевелся

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Самое главное о нейронных сетях. Лекция в Яндексе. [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/company/yandex/blog/307260/> – Дата доступа: 8.09.2017.
2. Л.М. Местецкий. Математические методы распознавания образов / МГУ-ВМиК, 2002–2004.
3. С.Хайкин. Нейронные сети: полный курс. 2-е изд. : Пер. с англ. – М.: ООО “И.Д.Вильямс”, 2016. – 1104 с.: ил – Парал.тит.англ.
4. Hubel D.H., Wiesel T.N. // Journal of Physiology London., 1962. Vol. 15. P 106–154.
5. Красильников Н. Н. Цифровая обработка 2D- и 3D-изображений: учеб. пособие / Н.Н.Красильников. – М.: ООО «БХВ-Петербург», 2011. – 602 с.
6. Что такое сверточная нейронная сеть [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/309508/> – Дата доступа: 08.09.2017.
7. В.В. Жильцов, В.В. Чувилова. Практикум по нейросетевым технологиям: учебно-методическое пособие. – Омск: СибАДИ, 2010. – 60 с.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1. ЗАДАЧА РАСПОЗНАВАНИЯ ОБРАЗОВ .....	5
1.1 Постановка задачи .....	5
1.2 Сравнительная характеристика методов распознавания изображений .....	5
2. АРХИТЕКТУРА СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ .....	9
2.1 Понятие сверточной нейронной сети .....	9
2.2 Слой свертки .....	10
2.3 Суть слоя свертки .....	12
2.4 Слой подвыборки .....	13
2.5 Обучение и тестирование .....	14
3. ЛАБОРАТОРНЫЕ РАБОТЫ .....	16
Лабораторная работа № 1 .....	16
Лабораторная работа № 2 .....	18
Лабораторная работа № 3 .....	22
Лабораторная работа № 4 .....	27
ПРИЛОЖЕНИЕ .....	30
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	35