МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ «ВИТЕБСКИЙ ГОСУДАРСТВЕННЫЙ

УНИВЕРСИТЕТ ИМЕНИ П.М. МАШЕРОВА»

Факультет\_\_\_\_\_\_\_\_Информационных технологий\_\_\_\_\_\_\_\_\_\_

(название факультета)

Кафедра\_\_\_\_\_\_\_Информатики и информационных технологий\_\_\_\_\_\_\_

(название кафедры, за которой закреплена дисциплина)

КУРСОВОЙ ПРОЕКТ

по дисциплине Операционные системы и системы программирования

(название дисциплины)

ПРЕДСКАЗАНИЕ ПРОГНОЗА ПОГОДЫ СПОМОЩЬЮ НЕЙРОННОЙ СЕТИ

(название темы работы заглавными буквами)

\_\_\_\_\_\_\_Овчаренко Валерий Иванович\_\_\_\_\_\_\_\_

(фамилия, имя, отчество студента (слушателя)

\_\_\_\_\_\_\_\_\_\_\_3 курс, 31з группа\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(курс, группа)

\_\_\_\_\_\_\_\_Корчевская Елена Алексеевна\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_доцент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(должность, учёная степень, звание)

Витебск,2019 **Содержание**

[Введение 3](#_Toc536755346)

[1. НЕЙРОННЫЕ СЕТИ 5](#_Toc536755347)

[1.1. Параллели из биологии 5](#_Toc536755348)

[1.2. Искусственная нейронная сеть 6](#_Toc536755349)

[1.3. Функция активации 7](#_Toc536755350)

[1.4. Архитектура нейроной сети 8](#_Toc536755351)

[2. Обучение нейроной сети 10](#_Toc536755352)

[2.1.Способы обучения нейронной сети 10](#_Toc536755353)

[2.2.Алгоритм обратного распространения ошибки 11](#_Toc536755354)

[3. Разработка программного продукта 14](#_Toc536755355)

[3.1.Выбор инструментальных средств разработки 14](#_Toc536755356)

[3.2.Алгоритмическое обеспечение проекта 14](#_Toc536755357)

[4. Пользовательский интерфейс программного продукта «коллекционер» 16](#_Toc536755358)

[4.1.Главное окно программы 16](#_Toc536755359)

[Заключение 18](#_Toc536755360)

[Список использованных источников 19](#_Toc536755361)

Введение

Искусственные нейронные сеети (ИНС) — математические модели, а также их программные или аппаратные реализации, построенные по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма. Это понятие возникло при изучении процессов, протекающих в мозге, и при попытке смоделировать эти процессы. Первой такой попыткой были нейронные сети Маккалока и Питтса.

Впоследствии, после разработки алгоритмов обучения, получаемые модели стали использовать в практических целях: в задачах прогнозирования, для распознавания образов, в задачах управления и др.

ИНС представляют собой систему соединённых и взаимодействующих между собой простых процессоров (искусственных нейронов). Такие процессоры обычно довольно просты, особенно в сравнении с процессорами, используемыми в персональных компьютерах. Каждый процессор подобной сети имеет дело только с сигналами, которые он периодически получает, и сигналами, которые он периодически посылает другим процессорам. И тем не менее, будучи соединёнными в достаточно большую сеть с управляемым взаимодействием, такие локально простые процессоры вместе способны выполнять довольно сложные задачи.

С точки зрения машинного обучения, нейронная сеть представляет собой частный случай методов распознавания образов, дискриминантного анализа, методов кластеризации и т. п. С математической точки зрения, обучение нейронных сетей — это многопараметрическая задача нелинейной оптимизации.

С точки зрения кибернетики, нейронная сеть используется в задачах адаптивного управления и как алгоритмы для робототехники. С точки зрения развития вычислительной техники и программирования, нейронная сеть — способ решения проблемы эффективного параллелизма. А с точки зрения искусственного интеллекта, ИНС является основой философского течения коннективизма и основным направлением в структурном подходе по изучению возможности построения (моделирования) естественного интеллекта с помощью компьютерных алгоритмов.

Нейронные сети не программируются в привычном смысле этого слова, они обучаются. Возможность обучения — одно из главных преимуществ нейронных сетей перед традиционными алгоритмами. Технически обучение заключается в нахождении коэффициентов связей между нейронами. В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными данными и выходными, а также выполнять обобщение. Это значит, что, в случае успешного обучения, сеть сможет вернуть верный результат на основании данных, которые отсутствовали в обучающей выборке, а также неполных и/или «зашумленных», частично искаженных данных.

**Цель курсовой работы**: создать программный продукт, который может предсказать погоду с помощью нейронных сетей.

**Задача курсовой работы:** изучить особенности работы взаимодействий с базой данных.

**Объекты исследования:** нейронные сети.

[1, 4, 7, 11]

©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©

©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©

©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©

©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©

©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©

©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©

©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©

©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©

©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©© lskcvnskjdhfksjcvn8798 sdkjghkjh234 jdf87 ksjdth34hs87 kdmgvnhsjehfd7 lksdghasdlkfj4 ghjk;t7 sldkvnsj5 slcmvnskdjg234234 sldkvhkjcvlskfjgl;kj98987 lsmdhglskdhlksjd34234 sldkfhslkgjlsd7897 sdkjghkjh234 jdf87 ksjdth34hs87 kdmgvnhsjehfd7 lksdghasdlkfj4 ghjk;©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©

©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©

©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©

©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©

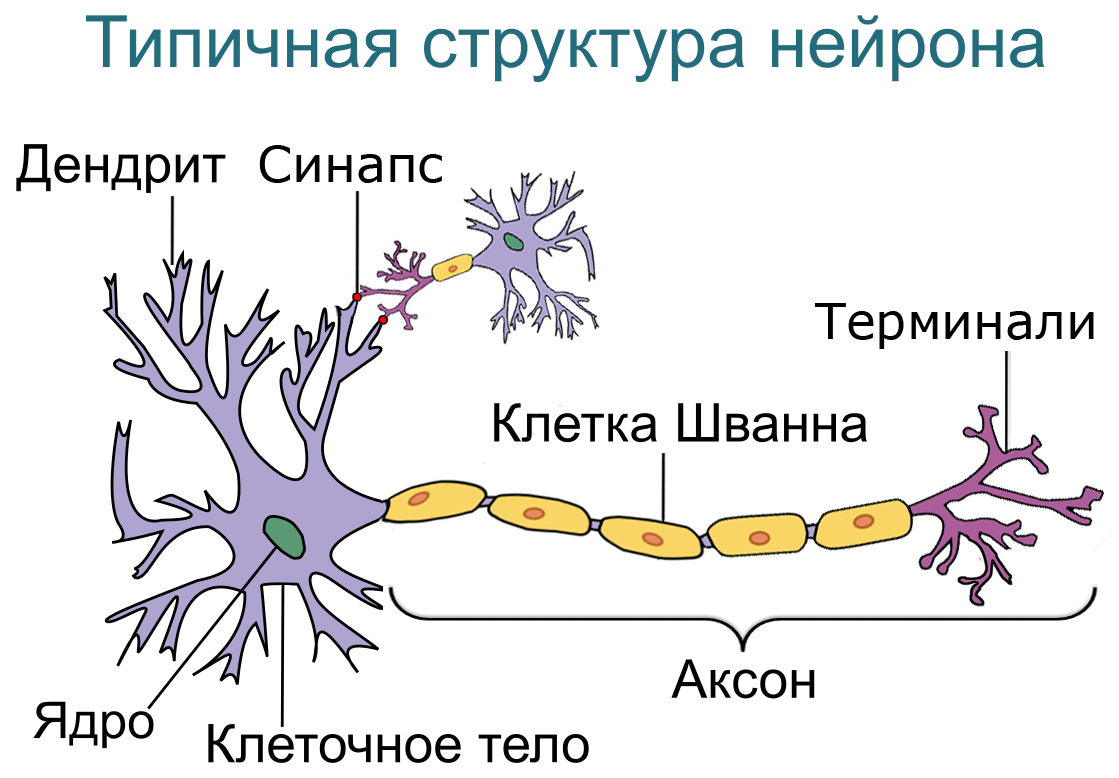
©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©

# НЕЙРОННЫЕ СЕТИ

## Параллели из биологии

Нейронные сети возникли из исследований в области искусственного интеллекта, а именно, из попыток воспроизвести способность биологических нервных систем обучаться и исправлять ошибки, моделируя низкоуровневую структуру мозга (Patterson, 1996).

Мозг состоит из очень большого числа (приблизительно 10,000,000,000) нейронов, соединенных многочисленными связями (в среднем несколько тысяч связей на один нейрон, однако это число может сильно колебаться).



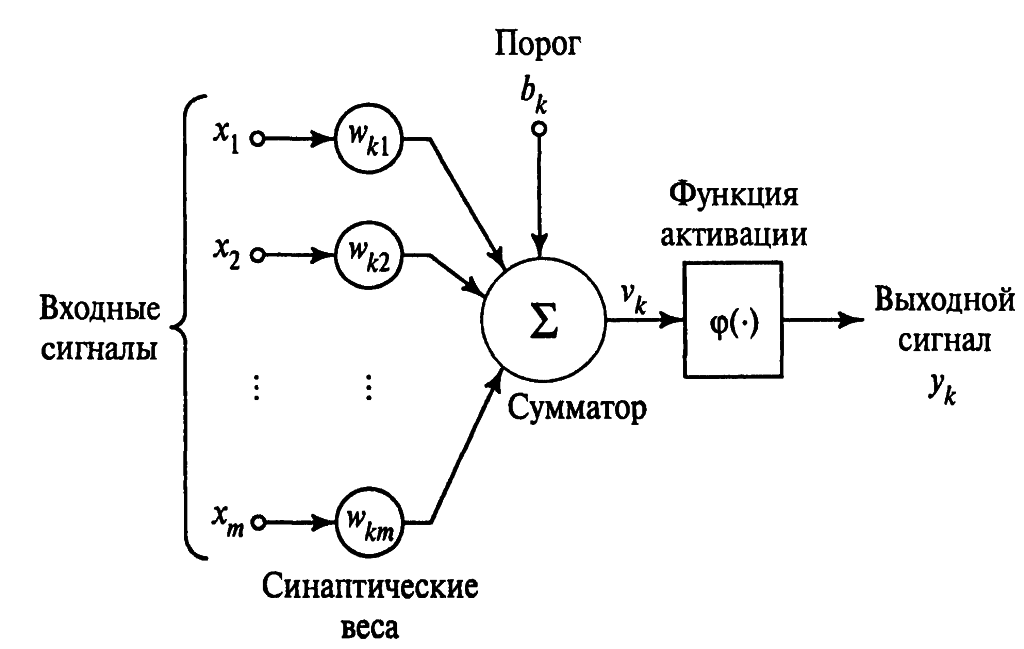
**Рисунок 1.1.1 – Нейрон**

Нейроны – это клетки, способные распространять электрохимические сигналы. Нейрон имеет разветвленную структуру ввода информации (дендриты), ядро и разветвляющийся выход (аксон). Аксоны клетки соединяются с дендритами других клеток с помощью синапсов. При активации нейрон посылает электрохимический сигнал по своему аксону. Через синапсы этот сигнал достигает других нейронов, которые могут в свою очередь активироваться. Нейрон активируется тогда, когда суммарный уровень сигналов, пришедших в его ядро из дендритов, превысит определенный уровень (порог активации).

## Искусственная нейронная сеть

Искусственная нейронная сеть (ИНС, нейронная сеть) - это набор нейронов, соединенных между собой. Некоторые входы нейронов помечены как внешние входы нейронной сети, а некоторые выходы - как внешние выходы нейронной сети. Подавая любые числа на входы нейронной сети, мы получаем какой-то набор чисел на выходах нейронной сети.

Основу каждой искусственной нейронной сети составляют относительно простые, в большинстве случаев - однотипные, элементы (ячейки), имитирующие работу нейронов мозга (далее под нейроном мы будем подразумевать искусственный нейрон, ячейку искусственной нейронной сети).



**Рисунок 1.2.1 – Нейрон**

Нейрон обладает группой синапсов - однонаправленных входных связей, соединенных с выходами других нейронов. Каждый синапс характеризуется величиной синоптической связи или ее весом wi.

Каждый нейрон имеет текущее состояние, которое обычно определяется, как взвешенная сумма его входов:



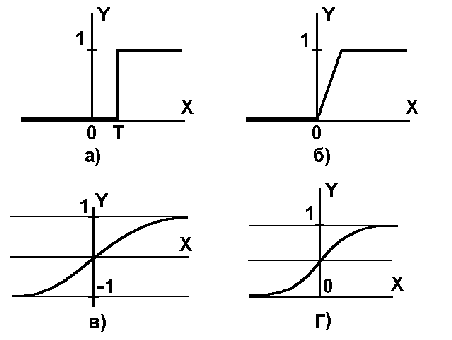
Нейрон имеет аксон - выходную связь данного нейрона, с которой сигнал (возбуждения или торможения) поступает на синапсы следующих нейронов. Выход нейрона есть функция его состояния:

y = f(s) – функция активации, которая и выполняет преобразование внутреннего сигнала.

## Функция активации

Функция активации может иметь разный вид :

* пороговый ( рис. 3.a),
* кусочно-линейный ( рис. 3.б),
* сигмоид( рис. 3.в, 3.г ).



**Рисунок 1.3.1 – Функции активации**

Множество всех нейронов искусственной нейронной сети можно разделить на подмножества – на слои. Взаимодействие нейронов происходит послойно.

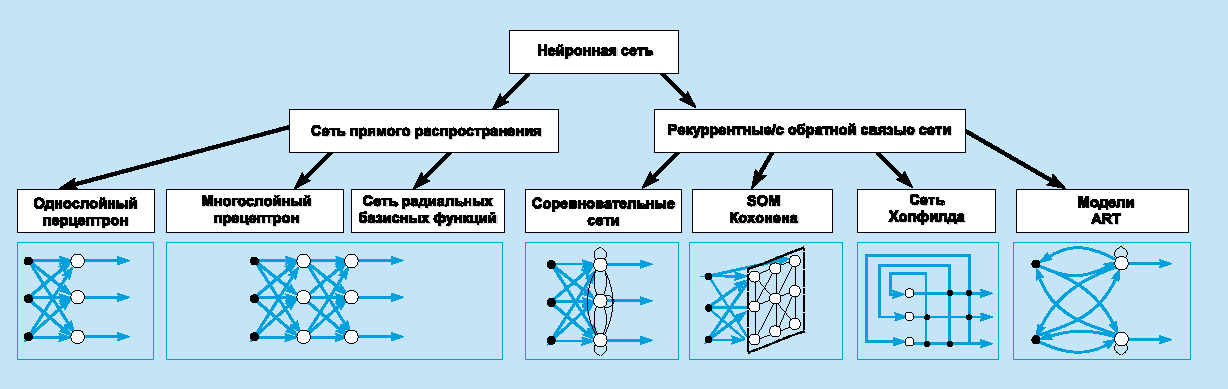
Слой искусственной нейронной сети - это множество нейронов на которые в каждый такт времени параллельно поступают сигналы от других нейронов данной сети.

Выбор архитектуры искусственной нейронной сети определяется задачей. Для некоторых классов задач уже существуют оптимальные конфигурации. Если же задача не может быть сведена ни к одному из известных классов, разработчику приходится решать задачу синтеза новой конфигурации.

Искусственные нейронные сети могут быть программного и аппаратного исполнения. Реализация аппаратная обычно представляет собой параллельный вычислитель, состоящий из множества простых процессоров.

## Архитектура нейроной сети

Искусственная нейронная сеть может рассматриваться как направленный граф со взвешенными связями, в котором искусственные нейроны являются узлами. По архитектуре связей ИНС могут быть сгруппированы в два класса (рис. 4): сети прямого распространения, в которых графы не имеют петель, и рекуррентные сети, или сети с обратными связями.

****

**Рисунок 1.4.1 – Архитектуры нейронных сетей**

В наиболее распространенном семействе сетей первого класса, называемых многослойным перцептроном, нейроны расположены слоями и имеют однонаправленные связи между слоями. На рис. 4 представлены типовые сети каждого класса. Сети прямого распространения являются статическими в том смысле, что на заданный вход они вырабатывают одну совокупность выходных значений, не зависящих от предыдущего состояния сети. Рекуррентные сети являются динамическими, так как в силу обратных связей в них модифицируются входы нейронов, что приводит к изменению состояния сети.

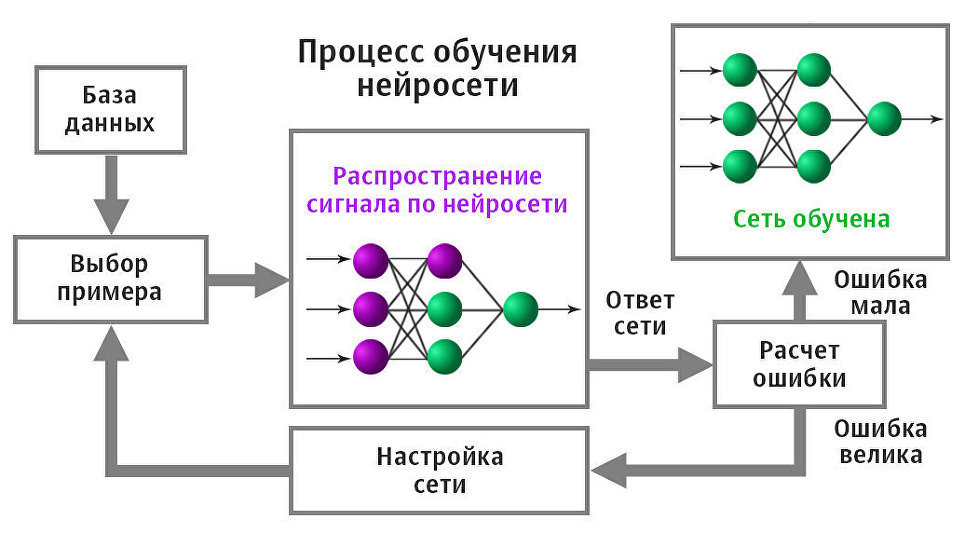
[1,3,13,14]

Sdfskj23 scvhsiudfhksdjf8 sldmvnsknv3 skdhvkjchv7 lkdshglkajsdhf457 lskcvnskjdhfksjcvn8798 sdkjghkjh234 jdf87 ksjdth34hs87 kdmgvnhsjehfd7 lksdghasdlkfj4 ghjk;t7 sldkvnsj5 slcmvnskdjg234234 sldkvhkjcvlskfjgl;kj98987 lsmdhglskdhlksjd34234 sldkfhslkgjlsd7897 slkdfjjhs;ldfkgj;lkdj345298798 lskdhg;akjgerjlksjdfoigjdlkjgs;j9384759234 jgldk3

# Обучение нейроной сети

## 2.1.Способы обучения нейронной сети

Способность к обучению является фундаментальным свойством мозга. В контексте ИНС процесс обучения может рассматриваться как настройка архитектуры сети и весов связей для эффективного выполнения специальной задачи. Обычно нейронная сеть должна настроить веса связей по имеющейся обучающей выборке. Функционирование сети улучшается по мере итеративной настройки весовых коэффициентов. Свойство сети обучаться на примерах делает их более привлекательными по сравнению с системами, которые следуют определенной системе правил функционирования, сформулированной экспертами.

****

**Рисунок 2.1.1 – Процесс обучения нейронной сети**

Для конструирования процесса обучения, прежде всего, необходимо иметь модель внешней среды, в которой функционирует нейронная сеть - знать доступную для сети информацию. Эта модель определяет парадигму обучения. Во-вторых, необходимо понять, как модифицировать весовые параметры сети - какие правила обучения управляют процессом настройки. Алгоритм обучения означает процедуру, в которой используются правила обучения для настройки весов.

Существуют три парадигмы обучения: "с учителем", "без учителя" (самообучение) и смешанная.

В первом случае нейронная сеть располагает правильными ответами (выходами сети) на каждый входной пример. Веса настраиваются так, чтобы сеть производила ответы как можно более близкие к известным правильным ответам. Усиленный вариант обучения с учителем предполагает, что известна только критическая оценка правильности выхода нейронной сети, но не сами правильные значения выхода.

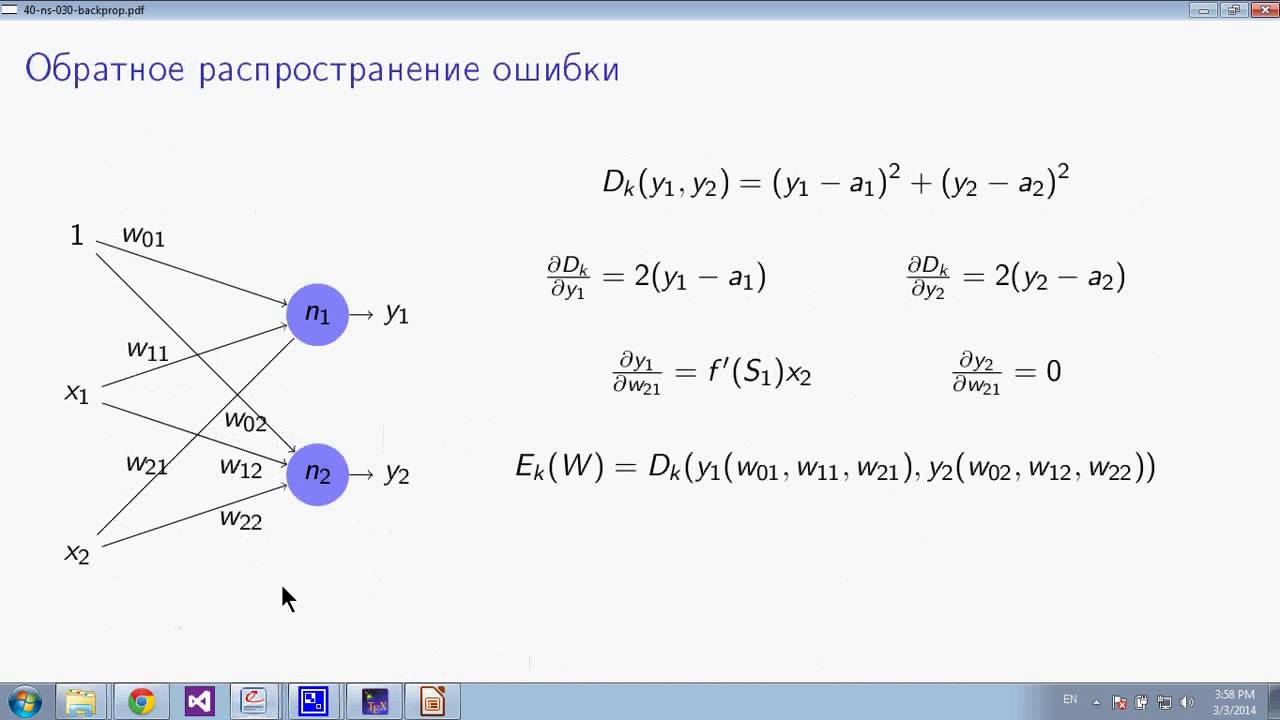
Обучение без учителя не требует знания правильных ответов на каждый пример обучающей выборки. В этом случае раскрывается внутренняя структура данных или Корреляции между образцами в системе данных, что позволяет распределить образцы по категориям.

При смешанном обучении часть весов определяется посредством обучения с учителем, в то время как остальная получается с помощью самообучения.

## 2.2.Алгоритм обратного распространения ошибки

Самый известный вариант алгоритма обучения нейронной сети - так называемый алгоритм обратного распространения (back propagation). Алгоритм обратного распространения наиболее прост для понимания, а в некоторых случаях он имеет определенные преимущества. Разработаны также эвристические модификации этого алгоритма, хорошо работающие для определенных классов задач, - быстрое распространение (Fahlman, 1988) и Дельта-дельта с чертой (Jacobs, 1988).

В алгоритме обратного распространения вычисляется вектор градиента поверхности ошибок. Этот вектор указывает направление кратчайшего спуска по поверхности из данной точки, поэтому если мы "немного" продвинемся по нему, ошибка уменьшится. Последовательность таких шагов (замедляющаяся по мере приближения к дну) в конце концов приведет к минимуму того или иного типа. Определенную трудность здесь представляет вопрос о том, какую нужно брать длину шагов.

****

**Рисунок 2.2.1 – Процесс обучения нейронной сети**

При большой длине шага сходимость будет более быстрой, но имеется опасность перепрыгнуть через решение или (если поверхность ошибок имеет особо вычурную форму) уйти в неправильном направлении. Классическим примером такого явления при обучении нейронной сети является ситуация, когда алгоритм очень медленно продвигается по узкому оврагу с крутыми склонами, прыгая с одной его стороны на другую. Напротив, при маленьком шаге, вероятно, будет схвачено верное направление, однако при этом потребуется очень много итераций.

На практике величина шага берется пропорциональной крутизне склона (так что алгоритм замедляет ход вблизи минимума) с некоторой константой, которая называется скоростью обучения. Правильный выбор скорости обучения зависит от конкретной задачи и обычно осуществляется опытным путем; эта константа может также зависеть от времени, уменьшаясь по мере продвижения алгоритма.

Обычно этот алгоритм видоизменяется таким образом, чтобы включать слагаемое импульса (или инерции). Этот член способствует продвижению в фиксированном направлении, поэтому если было сделано несколько шагов в одном и том же направлении, то алгоритм "увеличивает скорость", что (иногда) позволяет избежать локального минимума, а также быстрее проходить плоские участки.

Одна из наиболее серьезных трудностей изложенного подхода заключается в том, что таким образом мы минимизируем не ту ошибку, которую на самом деле нужно минимизировать ошибку, которую можно ожидать от сети, когда ей будут подаваться совершенно новые наблюдения. Иначе говоря, мы хотели бы, чтобы нейронная сеть обладала способностью обобщать результат на новые наблюдения. В действительности сеть обучается минимизировать ошибку на обучающем множестве, и в отсутствие идеального и бесконечно большого обучающего множества это совсем не то же самое, что минимизировать "настоящую" ошибку на поверхности ошибок в заранее неизвестной модели явления (Bishop, 1995).

[2, 5, 8, 10]

©ывало дывло мsdgkj lkscnv

Xfwiey hkjf ckj hskj dvnkjns©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©© lskcvnskjdhfksjcvn8798 sdkjghkjh234 jdf87 ksjdth34hs87 kdmgvnhsjehfd7 lksdghasdlkfj4 ghjk;©©©©©©©©©©©©©©©©©©©©©©©©©sldkhlskhdf skdjflk lskdjf lkasjdf lkasjdf iuwhejshcv iwhefkh jhsdioias3jdkghuhv jjlskdjf lkcjvbuhwie3uriohg lskdjoisvhj3nwngvnsjchf lskdjflkjdf oiwe6uvnkxjnclkvjslkdhfoiwhef asdlkfjlkjlcnvkhslkdjfj woeihfsdh2vmcnvskdjfhgisdk wieuryiwuyteptojjc xmcnb,xnkknvksjd iuywefhsjjlkcnv x nvnxklkcnvshdfoinv xmncv mxcnvjhsu9erysjdhmncv.,mkjdlfjowgmnx0cv scvhlkfgnnxkc3hfuwyfo0jlkzn7vlksjfwflksncvlk kchvoiejgk 8vshvbjlkjf kiv kjflsjd kksjdfldk vk;jslfjkgnnvs3oicj9vdj ksnkvnsv

slcmvnskdjg234234 sldkvh ©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©

# Разработка программного продукта

## 3.1.Выбор инструментальных средств разработки

Пользовательское программное обеспечение «WeatherNetwork» будет разработано в среде программирования Microsoft Visual Studio 2017 на объектно-ориентированном языке высокого уровня Си Шарп, с использованием разработанной специально для этого проекта библиотекой NNL (Neural Network Library).

«Microsoft Visual Studio Ultimate 2017». Визуальное программирование позволяет быстро создавать интерфейс программы и делать его более качественным за счёт наилучшего расположения информации окна на экране монитора, избежать многих ошибок уже на этапе проектирования.

Интегрированная среда разработки Visual Studio – полнофункциональная интегрированная среда разработки для Windows и веб-приложений и облачных приложений.

Visual Studio позволяет быстро и эффективно писать код, не теряя из виду контекст текущего файла. Можно легко углубиться в подробности, такие как структура вызова, связанные функции, возвраты и состояние тестирования. Также доступны рефакторинг кода, нахождение и устранение ошибок в коде.

## 3.2.Алгоритмическое обеспечение проекта

В ходе проектирования программного продукта проведена следующая работа:

− исследована предметная область, собрана необходимая информация (текстовая, графическая и др.);

− определены цели и задачи проектирования приложения;

− выбран язык программирования.

При разработке структуры проекта необходимо:

− спланировать структуру приложения (разделы, навигация, программные функции);

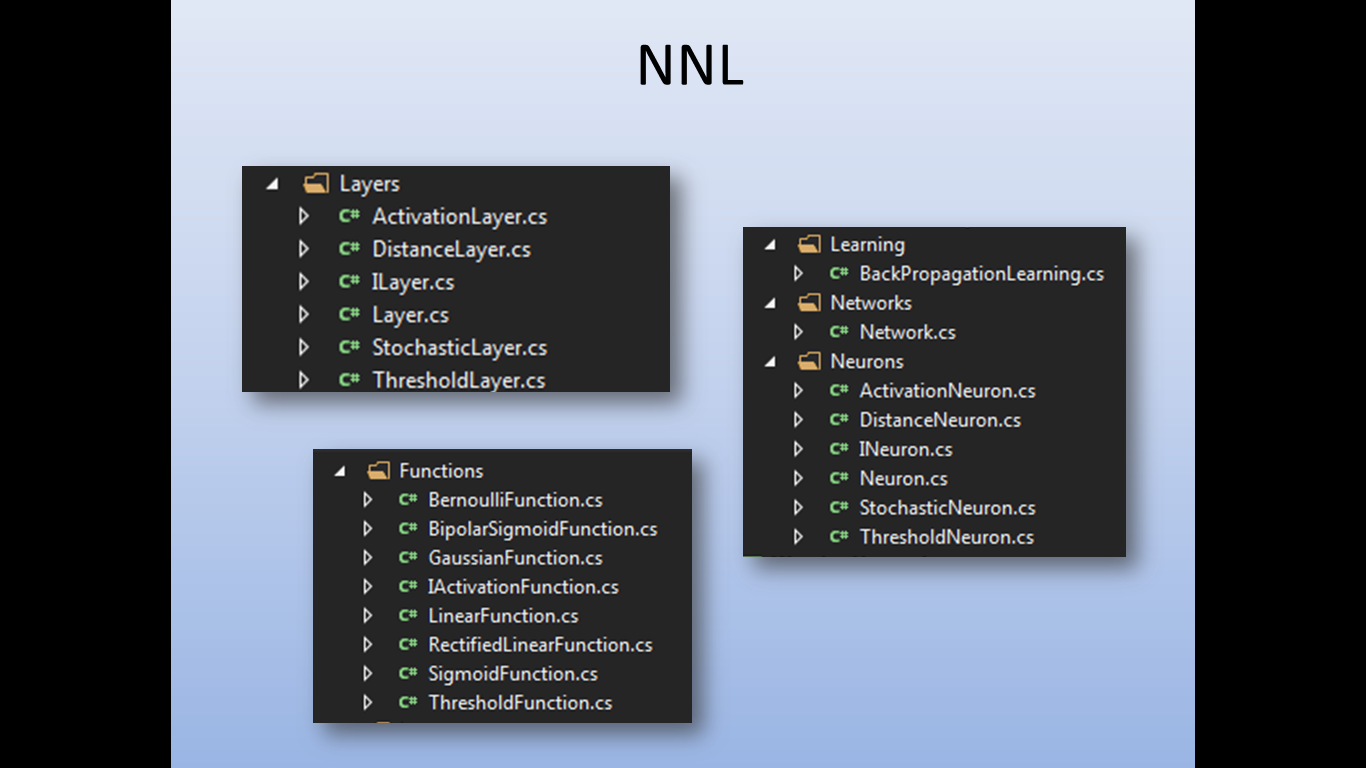
− разработать дизайн приложения;

Интерфейс программы должен быть простым и удобным для пользователя. При этом программный продукт должен хранить необходимую информацию и выполнять заявленные функции. Необходимо также продумать удобную навигацию по приложению.

Для реализации программного продукта выбран язык программирования С#.

## 3.3. Библиотека Neural Network Library

Neural Network Library (NNL) – высокоуровневая объектноориентированная библиотека, разработанная специально для данного курсового проекта, и содержит функционал для организации работы с нейронной сетью.



**Рисунок 3.3.1 – Структура библиотки NNL**

В состав библиотеки входят классы для создания нейронов:

* ActivationNeuron – нейрос с активационной функцией;
* DistanceNeuron – дистанционный нейрон;
* TresholdNeuron – нейрон с дистанционной функцией;
* StohasticNeuron – стохастический нейрон;

Так же в состав этой библиотеки входят шесть классов функций активации.

Помимо функций активации есть пять классов слоёв для организации нейронной сети и класс, представляющий собой нейронную сеть – Network.

Для обучения нейронной сети был реализован класс BackPropagationLearning – реализующий обучение на основе алгоритма обратного распространения ошибки.

## 3.4. Структура разрабатываемой нейронной сети

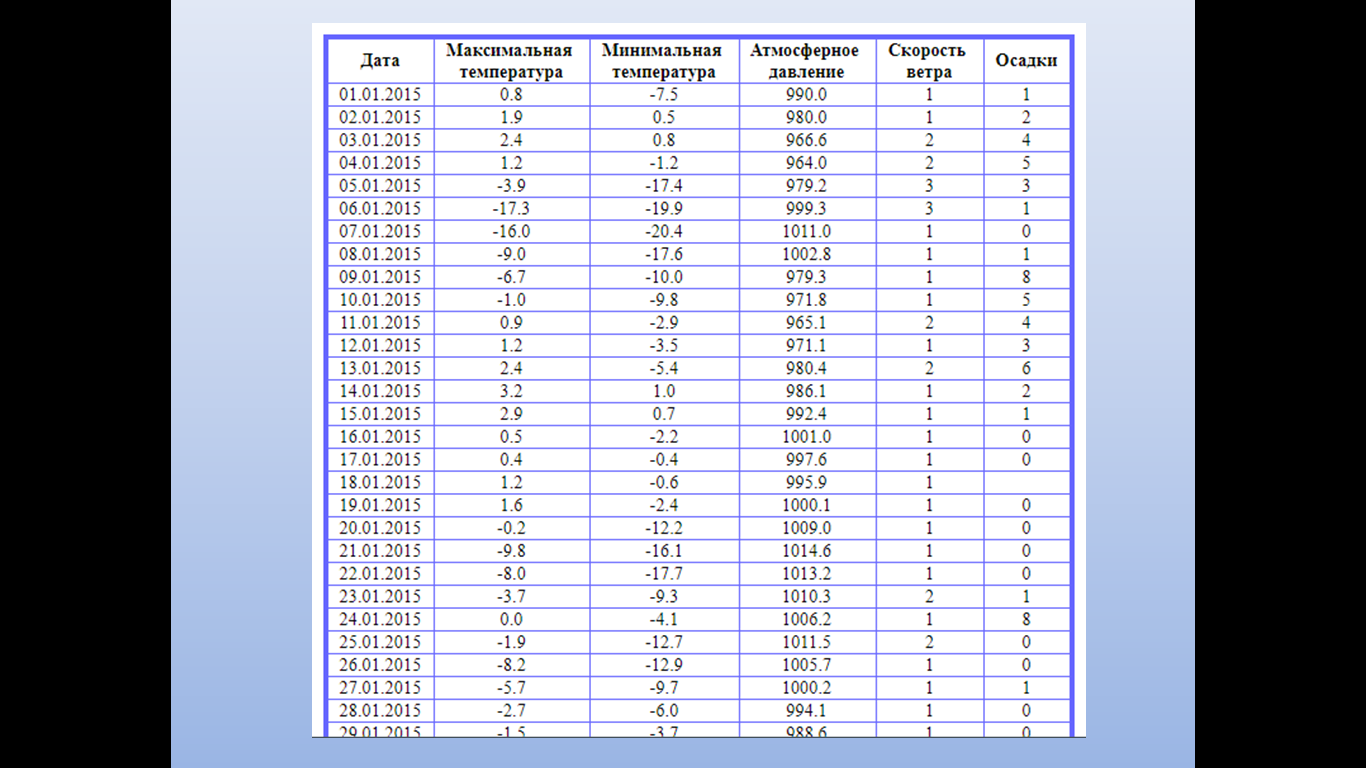
Разрабаывамая нейронная сеть будет представлять собой трёхслойный перцептрон с пятью нейронами на каждом слое. У нейронов первого слоя будет четыре входа, а у остальных – пять, по количество нейронов на предыдущем слое.



**Рисунок 3.4.1 – Разрабатываемая нейронная сеть**

Данная нейронная сеть разработана с учётом структуры входной информации, и реализована следующими объектами:

* TresholdLayer – это слой который содержит массив объектов TresholdNeuron – нейрон с дистанционной функцией;
* ActivationLayer – это слой который содержит массив объектов ActivationNeuron – нейрон с функцией активацией;
* Network – объект который представляет собой нейронную сеть и содержит необходитмый функционал для работы с ней;
* BackPropagationLearning – объект, который организует обучение нейронной сети методом обратного распространения ошибки.



**Рисунок 3.4.2 – Таблица исходных данных**

[7,8, 9, 11]

©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©l89k8797 we234

# Пользовательский интерфейс программного продукта «коллекционер»

## 4.1.Главное окно программы

Главное окно программы представляет собой основное рабочее пространство для пользователя, программно представлено объектом MainForm, наследованным от стандартного класса Form. Остальные окна программы представляют собой аналогичные объекты.

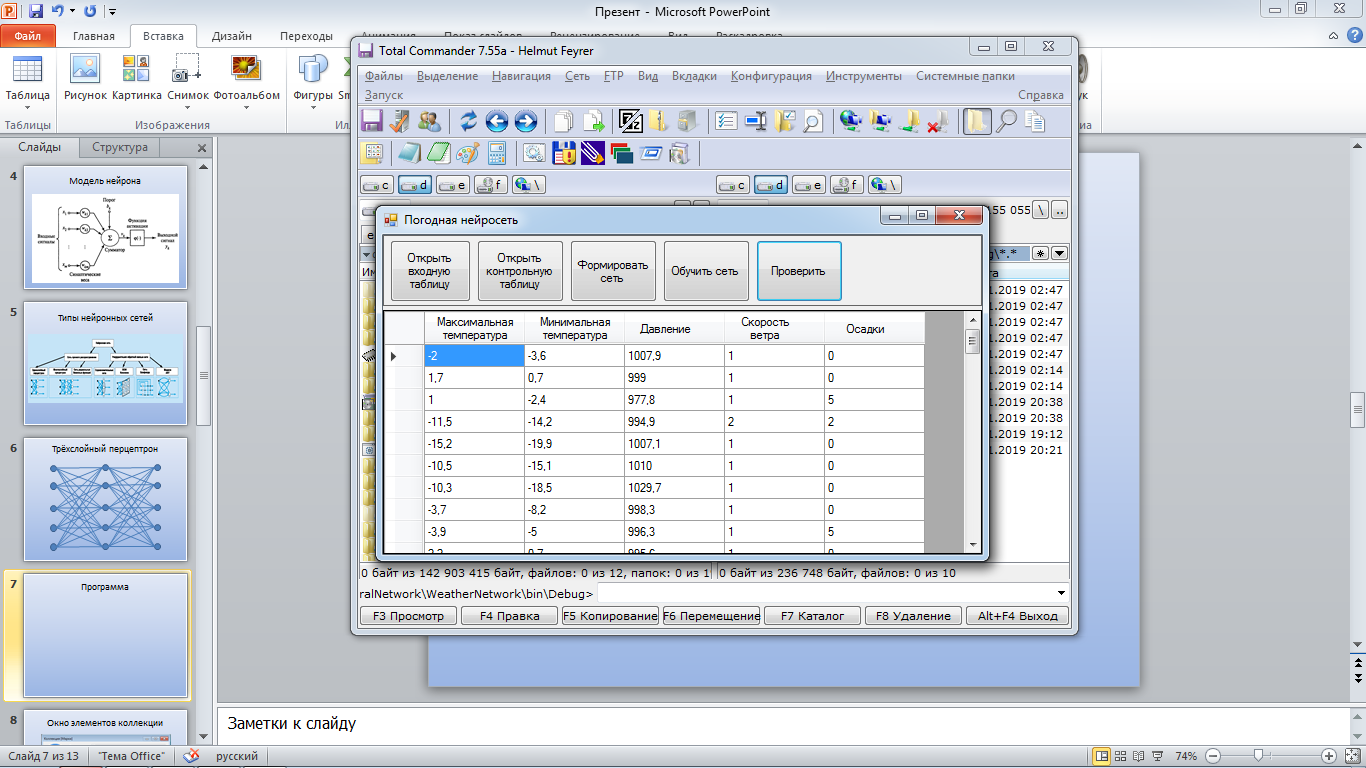


Рисунок 6.4.1 – Главное окно программы

На главной форме размещён компонент splitContainer, который предназначен для разделения формы на две динамических панели, которые нужны для упорядочивания других элементов управления.

Верхняя панель – Panel1 компонента splitContainer содержит компонент toolStrip, внутри которого размещены кнопки и вертикальные разделители:

* buttonOpenInputTable – кнопка для вызова диалога открытия файла, чтобы выбрать файл исходных табличных данных;
* buttonOpenControlTable – кнопка для вызова диалога открытия файла, чтобы выбрать файл контрольных табличных данных;
* buttonFormNetwork – кнопка для формирования объекта, представляющего нейронную сеть;
* buttonLearningNetwork – кнопка для обучения уже созданной нейронной сети;
* buttonValidate – кнопка для вывода полученных данных.

[4,6,8]

# Заключение

На данном этапе развития информационных технологий данная программа может упростить пользователям работу с помощью графического интерфейса, который не требуют большего изменения.

При выполнении курсового проекта были пройдены все этапы разработки специализированного прикладного программного обеспечения:

* изучение предметной области;
* создание объектноориентированной модели нейронной сети;
* создание библиотеки NNL;
* проектирование пользовательского интерфейса.

В данном курсовом проекте было разработано визуальное приложение, которое позволяет:

* На основе имеющихся данных с использованием алгоритма обратного распространения ошибок обучить нейронную сеть;
* Просматривать полученную информацию, полученную после обработки.

Для программы был разработан интерфейс, позволяющий не очень опытному пользователю работать с программой.

В ходе курсового проекта был более детально изучен язык высокого уровня С#, а также использование платформы .NET и ADO.

Разработанный в ходе выполнения курсового проекта программный продукт обладает следующими плюсами:

* Программный продукт разработан с учётом актуальных на сегодняшний день технологий проектирования программного обеспечения и баз данных;
* Программный продукт выполнен с учётом поставленной задачи курсового проектирования;
* Программный продукт разработанной версии содержит необходимый объём функционала, позволяющего пользователю использовать его потенциал.

[10,11,12]

Список использованных источников

1. Андреев А.М. Среда и хранилище: ООБД. – М.: «Мир ПК», 1997.

2. Бен Ватсон. C# 4.0 на примерах. Санкт-Петербург. : БХВ-Петербург, 2011, 590 с.

3. Буч Г. Объектно-ориентированный анализ и проектирование. 2-ое издание. - М.: «Бином», 1997.

4. Грабалов П.К. Компьютерная графика и основные графические редакторы. - Калининград,2003

5. Гуриков С. Р. Введение в программирование на языке Visual C#; Форум, Инфра-М, 2013. - 448 с.

6. Дейтел, Х. М.; Дейтел, П. Дж. Как программировать на С#; М.: Бином; Издание 4-е, 2005. - 390 с.

7. Джейсон, Прайс; Майк, Гандэрлой Visual C# .NET. Полное руководство; КОРОНА принт, 2004. - 960 c.

8. Лабор В.В. – С# Создание приложений для Windows (2003)

9. Мартин Р. С., Мартин М. Принципы, паттерны и методики гибкой разработки на языке С#; Символ-Плюс, 2011. - 768 с.

10. Нейгел, К. С# 2011 для профессионалов; Вильямс, 2012. - 763 с.

11. Пугачев С., Шериев А., Кичинский К. Разработка приложений для Windows 8 на языке C#; БХВ-Петербург, 2013. - 416 с.

12. Рихтер, Джефри CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке C#; Питер, 2007. - 656 c.

13. Робинсон, С.; Корнес, О.; Глинн, Д. и др. C# для профессионалов; М.: Лори, 2005. - 396 c.

14. Фленов Михаил Библия С#; БХВ-Петербург, 2009. - 560 с.