МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«кубанский государственный университет»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра информационных технологий**

**курсовая работа**

**РЕАЛИЗАЦИЯ И АНАЛИЗ РАБОТЫ НЕЙРОННЫХ СЕТЕЙ**

Работу выполнил А.Ю. Герштенцвейг

(подпись)

Направление подготовки 01.03.02 Прикладная математика и информатика

Направленность (профиль) «Системное программирование и компьютерные технологии» (Математическое и программное обеспечение вычислительных машин)

Научный руководитель

канд. пед. наук, доц. Н.Ю. Добровольская

(подпись)

Нормоконтролер

ст. преп. А.В. Харченко

(подпись)

Краснодар

2018

**РЕФЕРАТ**

Курсовая работа содержит 25 страниц, 5 изображений. Ключевые слова: нейронная сеть, обучение, связь, вес связи, нейрон, точность результатов.

Цель курсовой работы – изучение искусственных нейронных сетей с последующей программной реализацией и анализом отдельных параметров.

В ходе курсовой работы была изучена формальная модель искусственной нейронной сети и методы ее реализации, были рассмотрены некоторые способы представления и сохранения основных функциональных частей данной модели.

В работе реализована программа, позволяющая взаимодействовать с искусственной нейронной сетью. В приложении прописан метод обучения обратным распространением ошибки, функции оценки результатов работы нейронной сети, а также простой интерфейс для взаимодействия пользователя с программой. Приложение целиком реализовано на языке программирования C++.

**СОДЕРЖАНИЕ**

Введение 4

1 Математическая модель нейронной сети 5

1.1 Основные понятия теории нейронных сетей 5

1.2 Модель искусственного нейрона 5

1.2.1 Общая информация о модели нейрона 5

1.2.2 Способы представления искусственного нейрона 6

1.2.3 Функция активации нейрона 7

1.3 Синапсы в нейронных сетях 9

1.3.1 Понятие «веса» 9

1.3.2 Способы сохранения весов связей нейронов 9

1.4 Нейрон смещения 10

2 Основные принципы машинного обучения 14

2.1 Обучение с учителем 14 2.2 Обучение без учителя 15

2.3 Смешанное обучение 16

3 Реализация нейронной сети в приложении “Neya” 17

3.1 Технические особенности реализации 17

3.2 Проектирование приложения “Neya” 17

3.3 Обоснование выбора языка программирования и инструментария 18

3.4 Результаты работы 18

4 Анализ производительности и точности результатов работы искусственной нейронной сети 22

Заключение 24

Список использованных источников 25

**ВВЕДЕНИЕ**

В любой предметной области в явном или неявном виде применяется анализ и обработка данных, с их возможной классификацией. Существует несколько методов и механизмов математического решения этой задачи, которые легко автоматизировать и представить на языках программирования. Одно из самых перспективных направлений развития в вопросе обработки данных это искусственные нейронные сети. Искусственные нейронные сети представляют собой воплощение биологических нейронных сетей - связей нервных клеток. С помощью нейронной сети всевозможные обработки данных могут происходить без прямого участия человека.

В курсовой работе представлена программная реализация нейронной сети и метода ее обучения, а также результаты анализа скорости работы и обучения, и погрешности вычислительных операций нейронной сети.

**1 Математическая модель нейронной сети**

**1.1 Основные понятия теории нейронных cетей**

Искусственная нейронная сеть (ИСН) – это математическая модель, подразумевающая организацию сетей нервных клеток, а также ее программное или аппаратное воплощение [5]. Данное понятие формализовали Уоррен Мак-Каллок и Уолтер Питтс еще в 1943 году, предлагая вакуумные лампы для аппаратной реализации эквивалента нейронных сетей.

Искусственный нейрон – структурная единица сети, которая обрабатывает и передает информацию [9]. Нейроны соединяются друг с другом, образуя упорядоченную слоистую систему. Для нейронов существует активационная функция, преобразующая данные, поступившие в нейрон

Синапс – соединение двух нейронов, подразумевающее под собой путь передачи информации от одного нейрона к другому и ее значимость – «вес».

Активационная функция – функция, предназначенная для сглаживания значения нейрона и ограничения его целостности [9]. Данные функции всюду определены, но не каждая их них является непрерывной и дифференцируемой.

Смещение (сдвиг) – особый нейрон, не имеющий входного синапса (связи). Его входное значение всегда равно либо 1, либо -1. Нейроны смещения нужны для того, чтобы получать необходимый в обучении результат через смещение графика активационной функции нейрона.

**1.2 Модель искусственного нейрона**

**1.2.1 Общая информация о модели нейрона**

По своему описанию искусственный нейрон является моделью биологического нейрона и выполняет соответствующие эквивалентные функции. Человеческий мозг имеет приблизительно 1010 нейронов, каждый из которых может быть связан с 20 тыс. других нейронов. Для сравнения, у большинства насекомых количество нейронов не превышает 106. Чтобы создать нейронную сеть, подобную человеческой, необходимо огромное количество памяти и вычислительной способности процессора, для обработки данных и обучения.

Поскольку модель искусственной нейронной сети имеет слоистую систему, нейроны обычно классифицируют по их положению в топологии сети. Соответственно выделяют входные, скрытые (промежуточные) и выходные нейроны. Входные нейроны принимают вектор закодированных входных данных. Эти нейроны, обычно, не выполняют вычислительных функций, а просто передают взвешенный сигнал в следующие слои. Скрытые (промежуточные) нейроны выполняют основные вычислительные операции, как правило, суммирование и произведение. Выходные нейроны представляют из себя конечный результат работы нейросети. Результат при необходимости может дополнительно пройти через интерпретатор для представления в понятном для человека варианте.

**1.2.2 Способы представления искусственного нейрона**

Поскольку основные операции, выполняемые искусственным нейроном это сложение, умножение и хранение информации, то представлять его удобнее всего средствами, для которых предусмотрены данные функции. Стоит заметить, что при аппаратном строении нейросети эти операции могут быть реализованы через установку интегральных транзисторов, выполняющих основные функции алгебры логики.

В программной реализации сети нейроны можно представлять по-разному, например, двух- или трехмерным массивом одного из основных типов данных, позволяющих работать с вещественными числами. Аналогично подойдут контейнеры векторов или множеств.

Еще один способ представления нейрона – функциональный. Достаточно иметь два массива для хранения данных (не учитывая веса связей) и реализовать функцию, возвращающую последовательную сумму произведений. Таким образом из одного массива в другой можно получить результат прохода одного слоя связей нейросети, затем повторить эту операцию для всех слоев. Этот способ представления один из самых малотребовательных по памяти и достаточно просто реализуем на языках программирования, однако сильно забивает стек вызова.

**1.2.3 Функция активации нейрона**

Активационные функции иначе называют передаточными. Данные функции предназначены для сглаживания значения, хранящегося в нейроне. Это применяется для ограничения целостности значений и недопущении их выхода за допустимые пределы. Выделяют четыре основных передаточных функций: линейная, пороговая, сигмоидальная, гиперболический тангенс [4 с.53-54].

(1)

В формуле (1) представлена линейная активационная функция нейрона. Данную функцию принято применять к нейронам входного слоя с возможным сдвигом по обеим осям. Недостаток данной функции заключается в ее грубом переходе, не позволяющем дифференцировать функцию в некоторых методах обучения, которые требуют участия самой функции и ее производной в настройке весов.

(2)

Пороговая передаточная функция (2) иначе называется функцией Хэвисайда. Представляет собой резкий перепад. Сигнал в нейроне будет считаться нулевым пока взвешенный сигнал не достигнет некоторого уровня T. Данная функция не является дифференцируемой на всей оси абсцисс и ее не рекомендуется использовать в сетях, обучаемых по алгоритмам дифференцирования, например, сопряженные градиенты и обратное распространение ошибки.

(3)

Существует целый ряд функций, относящихся к сигмоидальным. Отличаются они, в основном, скоростью сходимости к своим предельным значениям. В формуле (3) представлена логистическая сигмоидальная функция, ограниченная по значениям в интервале (0;1). Преимущество сигмоидальных функций заключается в том, что они гладкие и непрерывные на всей оси абсцисс, следовательно, они легко дифференцируемы и применимы в методах обучения, требующих использование производной. Также сигмоидальные активационные функции позволяют избежать насыщения нейрона только от больших сигналов.

(4)

Передаточная функция гиперболического тангенса (4) во многом похожа на сигмоидальные, например, в гладкости и непрерывности, в дифференцируемости и т.д. Основное отличие с сигмоидальными функциями заключается в области значений. Гиперболический тангенс может иметь значения (-1;1), однако, график, по сравнению с сигмоидальными, меняет только масштаб. Данная активационная функция тоже позволяет избежать перенасыщения и насыщения от одного сильного сигнала.

**1.3 Синапсы в нейронных сетях**

**1.3.1 Понятие «веса»**

Под синапсом будем понимать связь двух искусственных нейронов. Основная задача связей между нейронами – это регулирование значимости сигнала, поступающего из одного нейрона в другой. Для формализации значимости используют понятие «вес» связи. Под весом понимается некоторое числовое значение, которое умножается на значение сигнала нейрона и, как результат, либо увеличивает значение, получаемое другим нейроном, либо, напротив, уменьшает его.

(5)

Формула (5) демонстрирует схему получения значения нейрона, путем суммирования произведений весов всех связей wi на значения нейронов xi, передающих сигнал, а также нейрона смещения x0. Затем результат суммы произведений проходит через активационную функцию f(), значение которой y и является полным сигналом, поступившим в данный нейрон.

**1.3.2 Способы сохранения весов связей нейронов**

Одним из самых важных моментов в теории нейронных сетей является представление и сохранение весов и самих связей между нейронами сети. Пусть в нейронной сети есть n слоев нейронов с m нейронами на каждом слое, в итоге всего выходит n \* m нейронов в данной модели. Учитывая факт строения сети и то, что в любых соседних слоях каждый нейрон из одного слоя должен быть связан с каждым нейроном из соседнего слоя. Следовательно, между каждыми двумя слоями будет m2 связей между нейронами, а всего связей в данной сети будет (n – 1) \* m2. Очевидно, что количество связей превышает количество нейронов в сети в (m – m / n) раз. Чем больше сеть, тем, соответственно, больше эта разница.

Необходимо также учесть тот факт, что веса связей должны сохраняться каким-либо способом для неоднократного использования, поскольку обучать нейронную сеть заново при каждом использовании крайне нерационально. Исходя из этого, возникает необходимость использования файловой системы или базы данных для сохранения и перезаписи весов связей. Для небольших нейронных сетей вполне достаточно использовать пару файлов, с возможностью чтения из них данных и их перезаписи. Однако, если размеры сети требуют сохранения больших данных, доступ к которым станет неудобным из файловой системы, рекомендуется использовать базу данных, способную хранить веса, а также индексацию связей.

Из способов сохранения весов следуют несколько вариантов представления связей в программном варианте, например, массивы данных для хранения полного набора всех весов связей. Существует специальный способ взаимодействия со статичными данными – это прямое чтение из файла или базы каждого значения в момент совпадающей индексации и последующая работа с ним. Данный способ представления может применяться если нет необходимости просмотра и изменения всех весов связей. К сожалению этот способ представления все равно вынуждает использовать массивы или контейнеры в процессе обучения, поскольку с постоянным обращением к файлам или базе данных значительно упадет скорость обучения практически для всех методов. Выбирая модель представления связей, придется пожертвовать либо свободной памятью, либо быстродействием и скоростью обучения.

**1.4 Нейрон смещения**

Помимо основных типов нейронов, существует еще один – нейрон смещения. Особенности этого типа заключаются в том, что они никогда не имеют входных синапсов, т.е. в них не передается значение из других нейронов и у нейронов смещения значение всегда равно константной -1 либо 1. Любая нейронная сеть может, либо иметь на каждом слое, кроме выходного, нейрон смещения, либо не иметь их вообще.

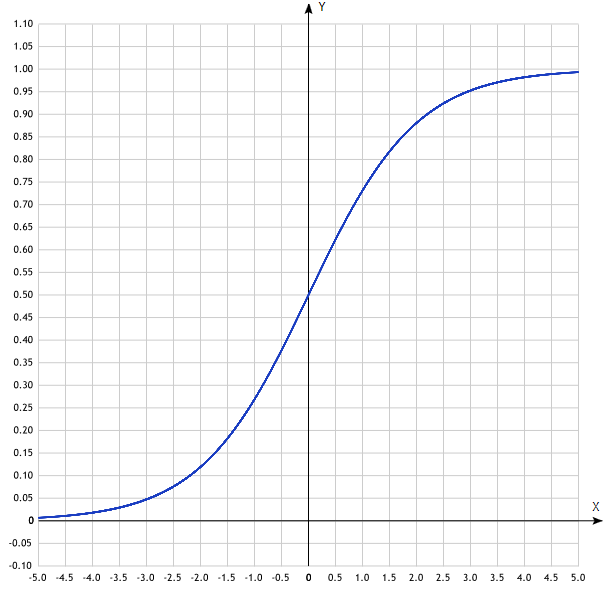


Рисунок 1 – График логической функции без смещения

Как показано на рис.1 график простой логической активационной функции не позволит нейрону принять иное значение в точках, удаленных от нуля. Иными словами, в процессе обучения нейронной сети изменяется только угол наклона графика активационной функции, и если точка, отвечающая за решение, будет находиться левее или правее передаточной функции, то едва ли нейронная сеть будет способна решить поставленную задачу без нейронов смещения.

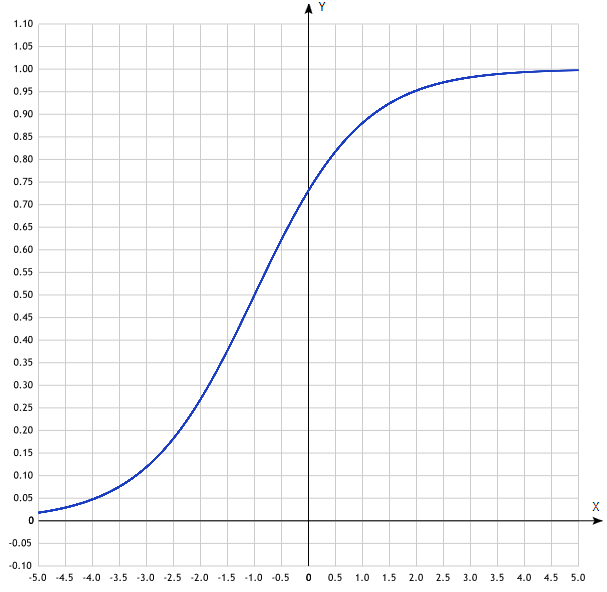


Рисунок 2 – График логической функции со смещения

На рис.2 показана возможность сдвига графика активационной функции используя смещение. Также нейрон смещения позволяет на наборе входных данных нулей получить не ожидаемые 0.5, а любое другое значение, необходимое для обучения.

Обычно в небольших нейронных сетях, предназначенных для узкого круга задач, не используют смещение, или же используют его исключительно для обучения. Каждый разработчик в постановке модели сети должен определить необходимость использования нейронов смещения.

В случае реализации в программе нейронов смещения надо учесть изменения методики обучения. При настройке весов связей нейронов на слоях возникает также потребность изменения весов на связях нейронов смещения с другими нейронами. Во многих методах обучения с учителем данный момент учитывается, однако если метод обучения не рассматривает использование нейронов смещения, рекомендуется отказаться от их реализации или рассмотреть другой метод обучения.

Представлять нейроны смещения достаточно просто. Хватает одного массива вне слоев сети, для хранения константной -1 или 1 любого нейрона смещения. Например, возможно представить нейрон смещения массивом булевского типа, где false означает -1, а true – 1. И перед передачей, пропустить значения через условный оператор для перевода в численный тип данных.

**2 Основные принципы машинного обучения**

**2.1 Обучение с учителем**

Само обучение представляет собой динамическое редактирование весов связей нейронной сети для получения ожидаемых результатов. Обучение нейронной сети с учителем подразумевает наличие определенного количества наборов выборок входных данных и соответствующих им результатам. Процесс настройки весов происходит следующим образом:

а) всем весам связей присваивают либо случайные маленькие значения в диапазоне [0;1], либо 0.5;

б) каждый пример из набора обучающих выборок прогоняется через нейронную сеть;

в) для результата прогонки данных рассчитывается ошибка через процентную точность относительно ожидаемых результатов или через максимальное (среднее) отклонение от ожидаемых результатов;

г) если ошибка достаточно мала обучение считается законченным, если нет происходит подстройка весов связей, в соответствии с методом обучения и повторяется шаг б).

Один из самых распространенных методов обучения нейронной сети с учителем является метод обратного распространения ошибки. Данный способ обучения подразумевает градиентный спуск при настройке весов и, соответственно, использование производной активационной функции. Исходя из этого, получим, что обучать данным методом можно только нейронные сети, передаточная функция в которых непрерывно дифференцируема, как например, логическая сигмоидальная функция.

Сама настройка весов связей происходит следующим образом:

- для каждого нейрона выходного слоя рассчитывается ошибка результата и умножает ее на производную активационной функции, параметром которой будет значение в нейроне, это значение называют дельта;

- для каждой связи выходного слоя с предыдущим происходит изменение ее веса путем вычитания дельты принимающего нейрона, умноженной на скорость обучения и на значение отправляющего нейрона;

- последовательно для каждого слоя связей повторяются предыдущие действия до тех пор, пока не будут просмотрены все связи нейронов или пока вес хотя бы одной из связей не примет недопустимое значение.

**2.2 Обучение без учителя**

Основным преимуществом обучения без учителя является то, что в данном методе не требуется использование размеченных данных, т.е. нет необходимости вручную подбирать выборки ожидаемых результатов. Из этого следуют две особенности – во-первых, появляется возможность использовать несопоставимо большие объемы данных, поскольку результат обучения не будет зависеть от количества и качества выборок ожидаемых результатов, во-вторых, неопределенность измерения точности результатов работы нейронной сети, т.к. в большинстве вопросов отсутствуют прямые или интуитивные метрики.

Часто встречающейся задачей данного типа обучения является задача кластеризации по конечному числу признаков. С этой задачей может справиться нейронная сеть, обученная с учителем, но достаточно часто бывает так, что нет возможности получить конечные результаты обучающих выборок. Задача нейронных сетей, обученных без учителя, состоит в предсказании соответствия объектов их классам и сформировать тем самым кластеры на основе всех имеющихся данных.

Наиболее популярным алгоритмом обучения для кластеризации данных является метод k-средних. Данный метод итеративный, он основан на минимизации суммарных квадратичных отклонений точек классов от их центров. Данный метод чаще всего применяется для фильтрования сигнала.

Этот алгоритм не подразумевает настройки весов связей в нейронной сети. Он используется для получения возможных ожидаемых результатов для выборок, а также для создания фильтров, используемых при обработке и анализе изображений. Зачастую данный алгоритм могут использовать для быстрого создания возможного ожидаемого результата, с целью подстройки весов через расчет ошибки.

**2.3 Смешанное обучение**

Если в случае полного отсутствия ожидаемых результатов приходится пользоваться алгоритмами обучения без учителя, то, когда имеется хотя бы несколько выборок с наличием конечного ожидаемого результата, используется метод обучения с подкреплением. За основу берут часть выборок с размеченными результатами и проходят алгоритм обучения без учителя, периодически дополняя основу другими выборками, используя тем самым алгоритм обучения с учителем. Поэтому данный способ обучения также называют смешанным обучением.

Этот тип обучения отлично подходит для той же задачи кластеризации, когда мы на основе имеющихся ожидаемых результатов можем представить примерное число возможных кластеров и их центры.

**3 Реализац­ия нейронной сети в приложении “Neya”**

**3.1 Технические особенности реализации**

С учетом особенностей модели нейронной сети, а также трудоемкости методов обучения выделяются некоторые трудности реализации. Самая главная проблема нейронных сетей – это процесс обучения. В зависимости от размеров нейронной сети время, затрачиваемое на обучение, изменяется, причем, чем больше сеть, тем дольше проходит одна эпоха обучения. К тому же, некоторые методы обучения могут привести к тупику обучения и процесс придется начинать заново. Второй, но не менее важной проблемой, стали затраты памяти на хранение данных нейросети, а именно динамические массивы и контейнеры для функционирования полносвязной сети, стек вызова функций и процедур, а также файлы статических данных и весов связей.

**3.2 Проектирование приложения “Neya”**

Главной задачей разработки является создание программы, реализующей модель искусственной нейронной сети. В дальнейшем будет модифицирована в полноценное приложение для работы с данными разного типа посредством нейросети. Также требуется учесть возможную разницу в размерности сети для выполнения разного рода задач.

Учитывая сказанное в пункте 3.1, возникает необходимость минимизировать затраты времени и памяти при создании приложения, реализующего нейронную сеть.

Таким образом выбрана модель нейронной сети, в которой:

- для нейронов и связей будут использоваться массивы контейнеров, в дальнейшем возможна реализация других методов;

- при хранении данных весов связей будет использоваться файловая система нерегулярного обращения;

- в модели, на первых этапах разработке не будут использоваться нейроны смещения;

- обучение сети будет осуществляться методом обратного распространения ошибки с использованием сигмоидальной логической функции и ее производной.

**3.3 Обоснование выбора языка программирования и инструментария**

Языком программирования, для разработки приложения был выбран C++. Несмотря на все преимущества Python, Java и C#, было принято решение начать разработку на более знакомом и очень функциональном языке программирования. Для первой версии программы достаточно было основных функций и процедур, доступных в C++, но с добавлением все новых и новых деталей программы, появлялась необходимость подключать дополнительно библиотеки.

На первых этапах разработки были подключены следующие библиотеки:

а) <iostream> - реализации потокового ввода/вывода;

б) <fstream> - работа с файловой системой;

в) <vector> - представление массивов данных в удобной для модификации форме;

г) <conio.h> - использование функции \_getch для реализации скачкового управления и переходов в меню.

**3.4 Результаты работы**

На текущий момент реализованы следующие элементы разработки:

- модель нейронной сети, представленная в виде двух классов;

- процедура обучения с учителем;

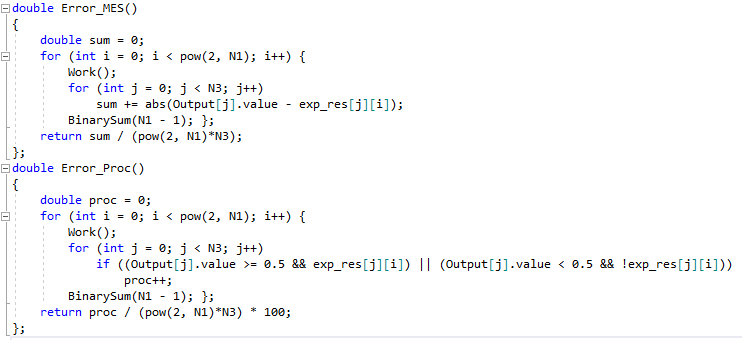
- процедура работы нейросети с интерпретаторами;

- простой консольный интерфейс для удобства взаимодействия пользователя с программой;

- небольшая файловая система для сохранения все необходимых для работы данных нейронной сети.

В текущей версии программы описано три класса, один из которых отвечает за взаимодействие с файлами, посредством прямого обращения, чтения и записи данных, а два других реализуют нейрон и связь между нейронами как объекты. Для представления массивов нейронов и связей используются контейнеры векторов типа объектов соответствующего класса.

Для расчета погрешности результатов работы, а также оценки точности нейронной сети прописано две метрики, представленные функциями типа double**.** Одна из функций возвращает значение, равное средней погрешности на наборе доступных выборок. Другая возвращает процент, с какой точностью сеть дала ответы, близкие к необходимым. Данные функции представлены на рис.3.

Рисунок 3 – Процедуры расчета точности работы сети

Обучение нейронной сети пока что осуществляется только одним методом – обратного распространения ошибки. На данный момент доступно только два способа обучения. Первый дает пользователю право самостоятельно указать скорость обучения и количество эпох прохождения обучения. Второй запускает автоматическое обучение по текущему установленному набору ожидаемых результатов с постепенным изменением скорости обучения. Этот способ будет проводить процедуру обучения эпохально до тех пор, пока не достигнет точности меньшей чем 0.01 по средней погрешности результатов или пока пользователь не прервет обучение вручную нажатием клавиши Enter.

Интерфейс взаимодействия с пользователем построен на основе скачущего выборочного меню, в котором пользователю доступны реализованные моменты программы. Меню выводит заранее заданные константные пункты, посредством перемещения по меню и через операторов switch/case вызывает ту или иную процедуру, функцию или же другой пункт меню.

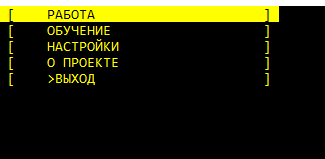


Рисунок 4 – Главное меню программы

На рис.4 продемонстрировано основное меню текущей версии программы. Из представленных пунктов только один не имеет никакого функционала – пункт «О ПРОЕКТЕ», поскольку приложение не доведено до этапа описания его полного функционала для первой версии релиза. Все остальные пункты дают пользователю доступ к всем уже реализованным элементом программы.

На данном этапе разработки программа способна:

- принять у пользователя набор входных данных и прогнав их через сеть и текущий интерпретатор вывести результат на экран;

- продемонстрировать результаты прохода сети на всех возможных наборах входных данных (при допущении на вход только нулей и единиц);

- запустить обучение нейронной сети доступным способом;

- вывести на экран текущие оценки точности и погрешности;

- позволить пользователю сбросить текущие веса связей нейронов до случайно заданных;

- позволить пользователю задать полный набор случайных ожидаемых результатов с целью случайного обучения сети.

В версии 0.1.02 реализована узконаправленная цель – анализ преступности деяния. На 7 Всероссийской научно-практической конференции “Информационное общество и право” докладывались результаты работы приложения, обученного на заранее подготовленных обучающих выборках. Основываясь на трех основных параметрах, каждый из которых содержит четыре взаимоисключающих варианта, представляемых шестью входными нейронами, на выходной нейрон подается результат прохода сети. Результат интерпретирует деяние, охарактеризованное входным сигналом, как преступное или же наоборот.

**4 Анализ производительности и точности результатов работы искусственной нейронной сети**

На основе созданной программы, реализующей нейронную сеть, проведен статистический эксперимент с обучением обратным распространением ошибки. С целью определить наиболее оптимальный параметр скорости обучения для данного алгоритма нейронная сеть многократно обучена, сброшена и перестроена под другую размерность. Обучение автоматически завершалось на каждом статистическом варианте при достижении точности в среднем отклонении от ожидаемого результата меньшей чем 0.1, причем количество входных нейронов не превышает десяти, выходных – трех. Для каждого эксперимента случайным образом генерируется полная выборка наборов с их ожидаемыми результатами. Это значение не превышает 210 = 1024 набора размеченных данных.

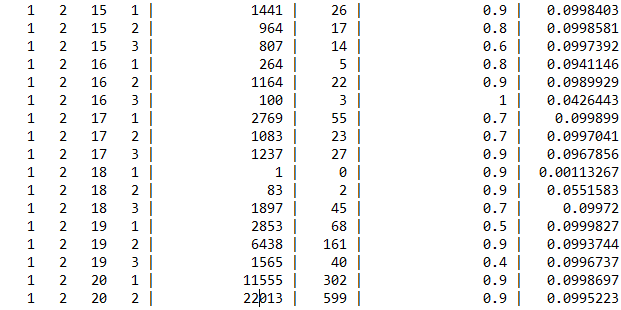
Вся статистика собрана через функции библиотеки <ctime>, а также с помощью средств диагностики в visual studio. Процентная точность результатов обучения не замерялась, поскольку в обучении участвовали случайные ожидаемые результаты и никакой смысловой ценности и значимости замеры не несли. Результатами анализа является следующая статистика:

- среднее время обучения нейронной сети методом обратного распространения ошибки размерностью до 102 нейронов составляет примерно 5 минут 51 секунду, это время значительно снижается при удалении излишних скрытых слоев;

- самым оптимальным параметром скорости обучения оказалось значение 0.7, при котором результат достигает заданной точности достаточно быстро и практически без скачков погрешности в процессе обучения. Можно сделать вывод, что параметр скорости обучения, лежащий в некоторой окрестности числа 0.7 будет самым оптимальным для сетей небольшой размерности;

- время прогонки входного сигнала через все слои нейронной сети, размерности до 102 не занимает более 309 миллисекунд, учитывая, что способ представления нейронов достаточно оптимальный по времени обработки.

На рис.5 приведена часть статистики из файла перебора. В первых четырех столбцах записана размерность сети, далее идет количество эпох и время обучения соответственно, а последние два столбца это параметр скорости обучения и достигнутая на последней эпохе обучения погрешность относительно ожидаемых результатов.

Рисунок 5 – Вырезка из статистического файла

Однако данная статистика проводилась исключительно на обработке формальной случайной логики исходя из данных нулей и единиц. При обработке изображения или текстовых данных придется использовать при обработке интерпретаторы и сжимающие процедуры, что займет гораздо больше времени для простой прогонки, а также для обучения.

**ЗАКЛЮЧЕНИЕ**

Цель курсовой работы – изучение искусственных нейронных сетей с последующей программной реализацией и анализом отдельных параметров – достигнута.

При подготовке курсовой работы изучены формальная модель искусственной нейронной сети, основы машинного обучения нейросетей, а также способы программной реализации.

В оригинальной части работы спроектировано и программно реализовано приложение “Neya”. На данном этапе разработки приложение способно принимать у пользователя входные данные для прогонки их через нейронную сеть, запускать алгоритм обучения обратным распространением ошибки для сети текущей размерности и заготовленных выборок. Приложение полностью реализовано на языке C++. Для сохранения весов связей нейронов, а также данных о строении и размерности сети использована файловая система.

Дальнейшими целями разработки являются дополнение, развитие и оптимизация программы. Будут реализованы другие алгоритмы обучения и способы представления основных компонент нейронной сети. Рассматривается перенос программы на API приложение для создания нового интерфейса взаимодействия с пользователем, а также переход от файловой системы к базе данных. Планируется добавление интерпретаторов, фильтров и карт активации для обработки как отдельных, так и целого массива изображений, с последующим выводом результатов в указанном пользователем варианте.

**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1 Уоссермен Ф. Нейросетевая техника: теория и практика / Пер. с английского Ю.А.Зуев. – М.:Мир, 1992. – 184 с. (23.07.2018)

2 Каллан Р. Основные концепции нейронных сетей / Пер. с англ. — М.: Издательский дом "Вильямc", 2001. – 287 с. (12.10.2018)

3 Осовский С. Нейронные сети для обработки информации / Пер. с польского И.Д. Рудинского. – М.: Фнансы и статистика, 2002. – 344 с. (11.08.2018)

4 Галушкин А.И. Нейронные сети: основы теории / М.: Горячая линия-Телеком, 2010. – 496 с. (05.12.2018)

5 Искусственная нейронная сеть — Википедия: [Электронный ресурс] // ru.wikipedia.org. URL: <https://ru.wikipedia.org/wiki/Искусственная_нейронная_сеть>. (16.12.2018)

6 Горбань А.Н. Обучение нейронных сетей / М.: ПараГраф, 1990. – 160 с. (09.10.2018)

7 Боровиков В.П. Нейронные сети. Statistica Neural Networks. Методология и технологии современного анализа данных / 2-е изд., перераб. и доп. – М.: Горячая линия - Телеком, 2008. – 392 с. (12.09.2018)

8 Каллан Р. Основные концепции нейронных сетей / Пер. с англ. — М.: Издательский дом "Вильямc", 2001. – 287 с. (01.09.2018)

9 Искусственный нейрон — Википедия: [Электронный ресурс] // ru.wikipedia.org. URL: <https://ru.wikipedia.org/wiki/Искусственный_нейрон>. (16.12.2018)

10 Добронравов О.Е., Овчинников В.В. Проектирование схем и узлов ЭВМ на пороговых элементах. – М.: Энергия, 1976. – 107 с. (10.08.2018)