|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Фундаментальные науки»

КАФЕДРА «Вычислительная математика и математическая физика» (ФН-11)

**ОТЧЕТ ПО ПРАКТИКЕ**

Студент *Зарина Ольга Юрьевна*

Группа *ФН11-23Б*

Тип практики *Учебная (ознакомительная)*

Название предприятия *НОЦ «СИМПЛЕКС» МГТУ им. Н.Э. Баумана*

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_О.Ю.Зарина

*подпись, дата*

Руководитель практики \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.А. Прозоровский

*подпись, дата*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Москва*

*2020 г.*

РЕФЕРАТ

НЕЙРОННЫЕ СЕТИ, СЕТИ ПРЯМОГО РАСПРОСТРАНЕНИЯ, РЕККУРЕНТНЫЕ СЕТИ, ГРАДИЕНТНЫЙ СПУСК, МЕТОД ОБРАТНОГО РАСПРОСТРАНЕНИЯ, КЛАССИФИКАТОР, ОБУЧЕНИЕ, РЕАЛИЗАЦИЯ

Объектом изучения являются нейронные сети прямого распространения, способные решать задачи классификации.

Цель работы - создание классификатора на основе сети прямого распространения, способного распознавать рукописные цифры.

В процессе работы детально изучались математические идеи и принципы, необходимые для решения задачи, методы обучения сети, а также изучался язык программирования Python, на котором реализовывалась сеть.

В результате был создан классификатор, способный с точностью до 97,102% распознавать рукописные цифры.

**Оглавление**

**Введение**

1. **Общие сведения о нейронных сетях**
2. **Разработка классификатора**

**2.1 Детальная математическая основа разработки**

1. **Реализация на языке Python**

**3.1 Задача 1**

**3.2 Задача 2**

1. **Процесс обучения и результаты**

**Заключение**

**Список литературы**

**Приложение А**

**Введение**

Целями учебной (ознакомительной) практики являются:

- ознакомление с одной или несколькими научными областями;

- ознакомление с государственными требованиями к оформлению научно-технической документации.

Задачами учебной (ознакомительной) практики являются:

- формирование реферата по теме «Нейронные сети»;

- публичный доклад по выбранной теме;

- изучение ГОСТ 7.32-2017 «Отчёт о научно-исследовательской работе. Структура и правила оформления»;

- выполнение индивидуального задания, которое заключается в том, чтобы оформить реферат по выбранной научной теме в соответствии с требованиями ГОСТ 7.32-2017;

- выкладывание отчёта и презентации доклада в «Портфолио студента»;

- ознакомление со стандартами:

а) ГОСТ 2.105-95 «Единая система конструкторской документации. Общие требования к текстовым документам»;

б) ГОСТ 7.1-2003 «Библиографическая запись. Библиографическое описание. Общие требования и правила составления»;

в) ГОСТ Р 7.0.5-2008 «Библиографическая ссылка. Общие требования и правила составления»;

г) ГОСТ 7.82-2001 «Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления»; д) ГОСТ 7.12-93 «Библиографическая запись. Сокращение слов на русском языке. Общие требования и правила»;

е) ГОСТ 8.417-2002 «Единицы величин»;

ж) ГОСТ 7.54-88 «Представление численных данных о свойствах веществ и материалов в научно-технических документах. Общие требования»;

и) ГОСТ 7.9-95 (ИСО 214-76) «Реферат и аннотация. Общие требования»;

- ознакомление по желанию со стандартами:

а) ГОСТ 25123-82 (СТ СЭВ 1625-79) «Техническое задание. Порядок построения, изложения и оформления»;

б) ГОСТ 19.101-77 «Виды программ и программных документов»;

в) ГОСТ 2.305-2008 «Единая система конструкторской документации. Изображения, виды, разрезы, сечения»;

г) ГОСТ 2.316-68 «Единая система конструкторской документации. Правила нанесения на чертежах надписей, технических требований и таблиц»;

д) ГОСТ 2.111-68 «Единая система конструкторской документации. Нормоконтроль»;

е) ГОСТ 2.307-68 «Единая система конструкторской документации. Нанесение размеров и предельных отклонений».

Задачи современной науки становятся всё более сложными и ёмкими.

Многие из них невозможно или очень затратно решать старыми методами, потому что решение этих задач требует человеческого интеллекта. Такие задачи сегодня решаются с помощью специально обученных нейронных сетей (НС), способных классифицировать информацию, обнаруживать необходимые в ней данные, предсказывать течение сложных процессов.

Область науки, занимающаяся разработкой и реализацией нейронных сетей, в данный момент особенно бурно развивается, внедряя новое программное обеспечение в различные технологические отрасли, способствуя автоматизации и модернизации. Также учёные надеются с помощью нейронных сетей лучше изучить работу человеческого мозга, поэтому НС играют не последнюю роль в нейробиологии и медицине.

Данная тема является очень обширной и сложной, поэтому в отчёте будут затронуты только основные теоретические данные. Несколько более подробно будут описаны сведения, непосредственно касающиеся проведенной научной работы, изложены все полученные результаты в виде таблиц и графиков.

Целью работы является создание классификатора на основе НС прямого распространения, который должен распознавать цифры, написанные человеком на бумаге.

* 1. **Общие сведения о нейронных сетях**

**Искусственные нейронные сети (ИНС)** — упрощенные модели биологических нейросетей (см. рис. 1) [1].

На создание таких моделей учёных вдохновила простота устройства мозга пчёл и птиц, решающего сложные задачи управления полётом со нелинейной траекторией, адаптации организма к погодным условиям, принятия решения и распознания пищи.

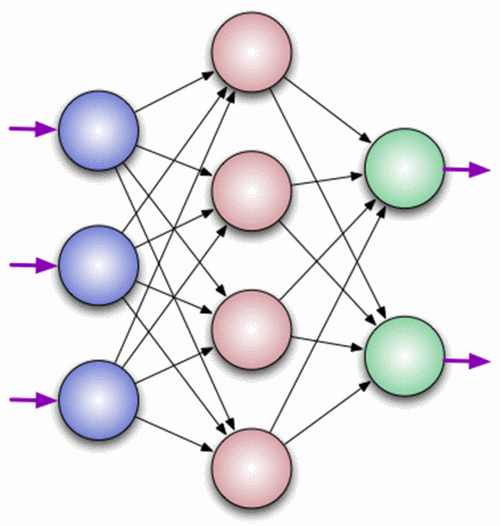


Рисунок 1 — модель ИНС

На основе ИНС строятся системы, называемые интеллектуальными системами или искусственным интеллектом.

**Искусственный интеллект (ИИ) —** наука о создании интеллектуальных систем (чаще всего специального программного обеспечения); свойство системы выполнять творческие функции, которые считаются прерогативой человека [1].

С понятием ИНС неразрывно связано машинное обучение, которое является чуть ли не самой важной частью получения умной системы.

**Машинное обучение** — подраздел области науки об искусственном интеллекте, изучающий различные способы построения обучающихся алгоритмов [1].

Если обобщить вышесказанное, то **искусственная нейронная сеть** — совокупность взаимодействующих между собой искусственных нейронов (узлов), способная посредством методов машинного обучения решать определенную задачу.

ИНС отличаются друг от друга количеством слоёв, направлением распространяемого сигнала.

По количеству слоёв выделяют:

1) однослойные сети (сигналы со входного слоя попадают сразу на выходной) (см. рис. 2);

2) многослойные сети (см. рис. 3);

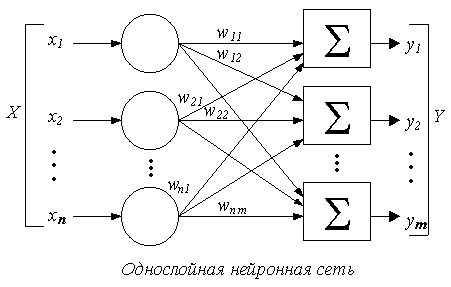


Рисунок 2 — однослойная сеть

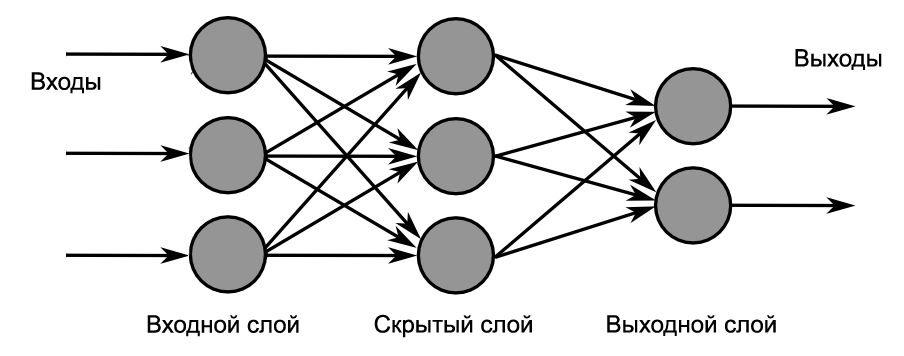


Рисунок 3 — модель трёхслойной сети

По направленности сигнала выделяют:

1) сети прямого распространения (сигнал передается строго от входного слоя к выходному);

2) сети обратного распространения (реккурентные сети) [2];

Теперь введём понятия, необходимые для понимания устройства и работы сети.

**Искусственный нейрон (ИН)** (см. рис. 4) представляет из себя сумматор, принимающий на вход сигналы от других нейронов и действующий на сумму сигналов специальной функцией активации, которая масштабирует сигнал и убирает шум [3]. Входной сигнал, поступающий в нейрон с

предыдущих нейронов выражается формулой

x1w1j+ x2w2j+…+ xnwnj=Σxiwij, (1)

где хi — сигнал, идущий с i-ого узла предыдущего слоя к j-ому узлу следующего слоя,

wij — вес связи между i-ым узлом предыдущего слоя и j-ым следующего,

i,j — индексы,

i=1..n.

ИН является упрощённой моделью настоящего нейрона.

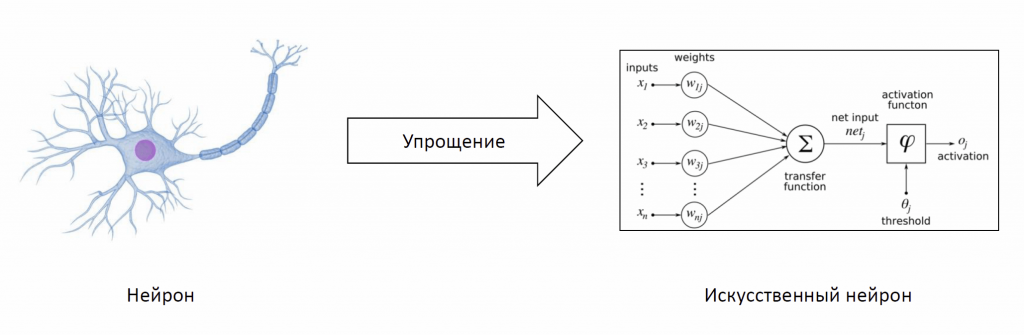


Рисунок 4 — искусственный нейрон

ИН бывают разных видов:

1) входной;

2) скрытый (находится в промежуточном слое);

3) выходной;

4) нейрон смещения (смещает сигнал в другую область значений);

5) контекстный нейрон [4];

У нейрона есть 2 параметра: **значение** **входных** и **выходных данных**. Нейроны взаимодействуют с друг другом с помощью **аксонов**, у каждого из которых есть один параметр — **вес**. Именно благодаря этому настраиваемому параметру каждого аксона сеть получает возможность обучаться.

Так как ИНС работает с сигналами только в определённом диапазоне, то сигнал нужно масштабировать. Также необходимо отличать настоящий сигнал от шума. Все эти условия обеспечивает **функция активации**, значение которой является выходным значением узла

out=ϕ(net), (2)

где out — выходное значение узла,

net — входной сигнал в узел, который вычисляется по формуле (1).

Основные функции активации:

1) **сигмоид** (самая популярная); диапазон значений: от 0 до 1;

2) **гиперболический тангенс**; диапазон значений: от -1 до 1 [4];

Сигмоид выражается формулой

Тренировка сети происходит с помощью **правильных тренировочных данных**, которая невозможна без ещё одной важной информации — **ошибки**, которая получается путём сравнения правильного ответа с ответом сети. Значение ошибки помогает корректировать вес каждой связи между нейронами. Целью обучения является максимальное снижение ошибки после каждой тренировки. Ошибку вычисляют несколькими способами.

Три основных:

1) среднеквадратичная ошибка (Mean Squared Error)(самый популярный способ)

2)квадратный корень от среднеквадратичной ошибки (Root Mean Squared Error)

3) арктангенс от значения ошибки (Arctan) [4].

Каждый из трёх способов задает функцию ошибки, которая является функцией от множества переменных. Переменными же являются веса.

Чтобы обновить вес, нужно выяснить, какой вклад в ошибку он сделал. Для этого используют следующие методы:

1) метод обратного распространения ошибки (более подробно в следующем разделе);

2) метод упругого распространения;

3) генетический алгоритм;

Чтобы выяснить, насколько нужно обновить вес, используют **метод градиентного спуска** (подробнее в следующем разделе).

Непосредственное обновление весов происходит с помощью методов:

1) стохастического (подробнее — далее);

2) пакетного;

3) мини-пакетного [4];

На способность обучения сети влияют **гиперпараметры**.

**Гиперпараметрами** называют такие изменяемые характеристики сети, которые подбираются опытным путём.

Гиперпараметрами являются:

1) коофициент скорости обучения;

2) количество скрытых слоев;

3) количество узлов в скрытых слоях [1].

Теперь стоит сказать о самом важном — об основных задачах, которые выполняют сети:

1) **Классификация** — распределение данных по параметрам;

2) **Предсказание** — предсказание развития процесса;

3) **Распознавание** (лиц, номеров, каких-либо объектов) [3];

Итак, в первом разделе была приведена основная информация, касающаяся ИНС, были предложены различные методы для каждого этапа создания сети.

* 1. **Разработка классификатора**

В данном разделе приступаем к подробному описанию работы.

**Цель работы**: получение обученной нейронной сети, которая должна максимально качественно классифицировать рукописные цифры по значению;

**Этапы разработки**:

1) подбор модели сети, подходящей под решение данной задачи;

2) подбор методов обучения сети, уточнение всех деталей;

3) выбор языка программирования для реализации;

4) выбор гиперпараметров;

5) написание и отладка;

6) обучение;

7) тестирование;

8) сбор статистических данных для настройки гиперпараметров;

9) настройка параметров;

10) завершение обучения и тестирование.

**Архитектура сети**: трехслойная сеть прямого распространения с настраиваемым количеством узлов;

**Язык реализации**: Python.

Python выбран как один из наиболее подходящих и функциональных языков для машинного обучения.

**Материалы для обучения и тестирования:** коллекция изображений рукописных цифр MNIST в формате csv.

**Основной тренировочный сет:** 60000 изображений.

**Основной сет для тестирования:** 10000 изображений.

**2.1. Математическая основа разработки**

Сеть состоит из трёх слоев, данного количества должно быть достаточно для классификации 10 объектов; первый слой только принимает входные данные, не действуя на них функцией активации, и передает их дальше. Сигнал с последнего слоя является финальным.

Из первого раздела известно, что сигнал, пришедший к узлу, является суммой произведений выходного сигнала каждого соответствующего узла с предыдущего слоя и веса соответствующей связи, то есть он вычисляется по формуле (1). Между двумя любыми узлами соседних слоев обязательно есть связь, имеющая вес. Понятно, что связей очень много, поэтому, чтобы упростить вычисления, веса между соседними слоями, а также входные, промежуточные, выходные сигналы записываются в матрицы, и все действия выполняются в матричной форме. Так входное значение для каждого узла во втором слое можно рассчитать по формуле

Где wij — матрица весов связей между первым и вторым слоями, причем в столбцах записаны веса всех связей, идущие от j-ого узла,

inputj — матрица-столбец входных значений,

xj— матрица-столбец значений, входящих в каждый j-ый узел второго слоя [3].

Вычисления для последующих слоев аналогичны, поэтому формула для них та же. Теперь после того, как слой получил свои входные значения в каждый узел, на них нужно подействовать функцией активации. Её выбор зависит от того, в каком диапазоне мы ожидаем выходные значения. Для моей задачи подойдет сигмоид с диапазоном от 0 до 1, потому что в отрицательных значениях нет необходимости. Таким образом, на значения полученной матрицы входных данных в любой из скрытых слоев, а также в выходной слой, необходимо подействовать функцией сигмоид, чтобы получить выходные данные из слоя. Все эти вычисления типичны вплоть до получения финальных значений.

После получения финальных значений, нужно понять, насколько правильно ответила сеть. Так начинается обучение. Для этого находится разность между правильным значением и выходным значением для каждого выходного узла (ошибка). Далее необходимо понять, какой вклад в ошибку сделал вес каждой связи: используем метод обратного распространения ошибки. Его принцип: пусть мы рассматриваем какой-либо узел выходного слоя и уже посчитали для него ошибку как разность между правильным значением, которое должны были получить на этом узле, и полученным. Известно, что к этому узлу идет по одной связи от узлов предыдущего слоя, тогда ошибку для каждой такой связи (веса) вычислим так

Вычисления аналогичны и для других узлов. Таким образом вычисляем ошибки для каждой связи между выходным и скрытым слоями. В матричном виде:

Знаменатели на деле можно отбросить [3]. Тогда первая матрица окажется транспонированной матрицей весов между рассматриваемыми слоями, тогда формула будет такой

Все вычисления одинаковы для связей между любыми другими соседними слоями, только вместо разности теперь берется уже высчитанная доля ошибки для соответствующей связи.

Теперь переходим к самому главному. Метод градиентного спуска позволяет определить, как нужно изменить вес каждой связи, чтобы ошибка ответа сети становилась минимальной. Для этого необходимо сначала определить функцию ошибки для каждого узла, она будет зависеть от функции выходного сигнала (2), то есть от функции многих переменных. Запишем её ещё раз в другом виде

Ок =сигмоид(ΣwikOi) ,

где Ок — выходной сигнал на рассматриваемом k-ом узле,

wik — вес i-ого узла предыдущего слоя, ведущего к k-ому узлу,

Oi — выходной сигнал из i-ого узла предыдущего слоя.

i=1..n , где n — количество узлов в предыдущем слое.

Мы уже определили ошибку как разность между правильным значением и полученным значением, тогда определим функцию ошибки для каждого узла как квадрат этой ошибки [3], а именно

Fe=(tk-Ok)2,

Где tk=const, правильное значение,

Ok — сложная функция выходного сигнала от n переменных.

С помощью вектора градиент нужно спуститься в локальный минимум этой функции. Если производная больше нуля, то надо двигаться против направления градиента, если производная меньше нуля — по направлению.

Чтобы найти градиент, мы должны вычислить все частные производные функции ошибки. Выкладки расчета частной производной от этой функции длинные, поэтому запишем конечную формулу

///= -(tk-Ok)\*сигмоид(ΣwikOi)\*(1- сигмоид(ΣwikOi))\* Oi ,

где tk-Ok — значение ошибки для каждой связи, которое высчитали методом обратного распространения ошибки, и эти значения находятся в матрице ошибок из формулы,

Oi — выходной сигнал из i-ого узла предыдущего слоя,

сигмоид(ΣwikOi) — функция выходного сигнала i-ого узла.

Теперь нужно записать функцию обновления, она будет выглядеть:

ΔW=α\*///,

Где α — коофициент обучения,

ΔW — это поправка для веса рассматриваемой связи.

Все наши рассуждения обобщенные, а формулы подходят для расчета поправок весов между любыми соседними слоями. Мы имеем матрицы весов, матрицы ошибок, входных и выходных значений, поэтому нет смысла считать поправки для каждой связи в отдельности — запишем их в матричной форме, охватывая одним выражением сразу все веса между любыми двумя соседними слоями///

Для непосредственного обновления матрицы весов используем стохастический метод, то есть просто прибавим поправки к существующим весам, а для этого сложим матрицу весов с соответствующей матрицей поправок.

На этом основные математические идеи заканчиваются. Осталось только сказать о рекомендациях диапазона входных, целевых значений и диапазона начальных весов. Сигмоид асимптотически стремится к 0 и 1, поэтому нет смысла в качестве начальных значений брать 0 и 1; лучше ограничиться диапазоном от 0,01 до 0,99; это же касается и целевых значений. Начальные веса определяются случайным образом, их рекомендуют брать из нормального распределения с центром в нуле [3].

* 1. **Реализация на Python**

**Задача 1**: написать класс с инициализацией трехслойной сети и основными методами: тренировкой и опросом. Также нужны два сеттера для изменения коофициента обучения и количества узлов в скрытом слое.

**Задача 2**: написать программу с использованием объекта класса непосредственно для решения главной задачи, в которую будут входить полные функции тренировки и опроса, функции эпохальной тренировки, сбора статистических данных и визуализации этих данных.

В Python есть модуль для удобной работы с массивами и матрицами — NumPy. Из SciPy взят сигмоид. Matplotlib.pyplot используется для визуализации данных [5]. Код написан в Visual Studio 2019.

**3.1 Задача 1**

В **методе инициализации** **(\_init\_)** сети 4 параметра:

1) количество узлов входного слоя;

2) количество узлов скрытого слоя;

3) количество узлов выходного слоя;

4) коофициент обучения

Матрицы весов заполняются случайными числами из стандартного распределения с центром в нуле. Так делаем только в начале обучения: уже после первой тренировки матрицы весов загружаются из файла.

В **методе тренировки (train)** 2 параметра:

1) список входных значений;

2) список выходных значений;

В данный метод заложена математическая основа, разобранная в 2.1 подразделе, позволяющая сети обучаться. Метод ничего не возвращает.

В **методе опроса** **(query)** 1 параметр: список входных значений. Метод возвращает ответ сети по каждому запросу [3].

В двух **сеттерах** по одному параметру, содержащие значения, на которые нужно изменить коофициент обучения и количество узлов в скрытом слое.

**3.2 Задача 2**

Сперва стоит сказать о файлах MNIST в формате csv, которые используются в качестве данных для обучения и опроса. Они могут быть открыты в текстовом редакторе. Их компонентами являются строки. Самый первый элемент строки — маркер, то есть число, изображение которого определяют последующие числа в строке.

В строке после маркера через запятую записаны числа в диапазоне от 0 до 255. Их ровно 784 в каждой строке. Это дискретное представление изображения с рукописной цифрой, указанной маркером.

Сразу ясно, что в качестве входных данных на каждой итерации нужно использовать значения после маркера, приведенные к диапазону от 0,01 до 0,99. А так как этих значений 784, то таким должно быть количество узлов во входном слое.

Также необходимо решить, в каком виде мы хотим получить выходные данные. Самой лучшей идеей будет использовать 10 выходных узлов, условно поставить на каждом из них маркеры от 0 до 9. Тогда наибольшее выходное значение на одном из десяти узлов можно считать фактическим ответом сети, определяя, какую цифру она выбрала. Отсюда сразу вытекает то, в каком виде должны быть целевые значения. Это должна быть матрица-столбец на 10 элементов, в которой индекс максимального значения 0,99 будет определяться маркером минус один, а все остальные элементы должны быть минимальными, то есть 0,01.

Количество узлов скрытого слоя — это гиперпараметр. Возьмем 100 узлов, этого должно быть оптимальным вариантом: не слишком мало, не слишком много. К сожалению, установить лучшее значение можно только опытным путём.

Как уже говорилось, в тренировочном сете 60000 примеров цифр, а в тестовом — 10000.

Переходим к основной части задачи 2. В основной программе подключаем все необходимые модули, в том числе и тот, что создан в Задаче 1.

Создаем необходимые подпрограммы.

**Процедура Training** не содержит параметров. Онасначала загружает данные из тренировочного файла, затем, в итерационном цикле, количество повторений в котором равно количеству компонентов в файле, каждая численная интерпретация цифры последовательно по одной преобразуется в список, также создается список целевых значений с помощью маркера. Полученные списки и передаются при вызове метода train. Ясно, что итераций во время тренировки должно быть 60000. После каждой тренировки матрицы обновленных весов должны записываться в бинарный файл в сжатом виде, чтобы не терялись результаты, и при каждом следующем запуске программы матрицы весов инициализировались матрицами из файла.

**Функция Query** должна опрашивать сеть, чтобы понять, как хорошо она обучилась. Она содержит один логический параметр. Сначала загружаются данные из тестового файла. Входные данные подготавливаются таким же образом, что и в предыдущей процедуре. Целевые значения для метода query не требуются, но маркер будет использован в качестве верного ответа для сравнения с ответом сети и занесения результатов в контрольный список: если ответ сети правильный — пишется 1, если нет — 0. В конце опроса вычисляется процент правильных ответов. Если переданный в функцию параметр равен “stat” (это значит, что Query вызвана из функции сбора статистических данных), то просто возвращается процент выполнения; если параметр равен другому значению, то результат опроса выводится на экран.

Примечательно, что примеры в файлах не повторяются и не содержат примеры друг друга. Это значит, что опрос покажет действительные навыки сети, а не заученные данные.

**Процедура Statistic** (без параметров)нужна для определения наилучшей конфигурации сети, то есть для настройки гиперпараметров. Для этого понадобится обучать сеть с нуля (снова со случайными весами без перезаписи их в файл) с измененными параметрами, которые фиксируются в специальных списках. Параметры сети (коофициент обучения и количество узлов во внутреннем слое) изменяются с помощью специальных методов. На каждой итерации после вызова метода тренировки сеть нужно опросить и занести все результаты в третий список. Для опроса вызывается функция Query со значением параметра “stat”. В конце работы процедуры все три списка преобразуются в массивы и пишутся в отдельный файл.

**Функция Statistic\_View** (без параметров) загружает массивы со статистикой и строит трёхмерный точечный график.

**Функция** **Epoch\_Train** обучает за некоторое количество эпох (повторений тренировочного цикла), которое определяется единственным параметром функции. Суть такого обучения заключается в отточке навыков сети на одних и тех же данных. Нюанс: эпох не должно быть много, потому что есть риск перенасытить сеть. **Epoch\_Train** делает всё тоже самое, что и Training, только несколько раз. После каждой итерации вызывается Query для опроса. Процент правильных ответов должен расти до некоторого предела. Если переборщить с количеством эпох — он начнёт падать.

Управление всеми подпрограммами происходит из основного цикла, выйти из которого можно, два раза нажав Enter. Чтобы вызвать любую функцию, нужно ввести её название. Если название некорректное, программа уведомит вас об этом.

* 1. **Процесс обучения и результаты**

После отладки всей программы и проверки работоспособности сети на более маленьких количествах примеров можно начинать полное обучение.

После успешно завершенной первой тренировки и вызова функции опроса результат составил 94,66%. Это неплохой результат, но можно добиться и лучшего с помощью настройки гиперпараметров.

Была вызвана процедураStatistic, затем Statistic\_View, которая вывела данные в виде трехмерного графика (см. рис.5 и 6).

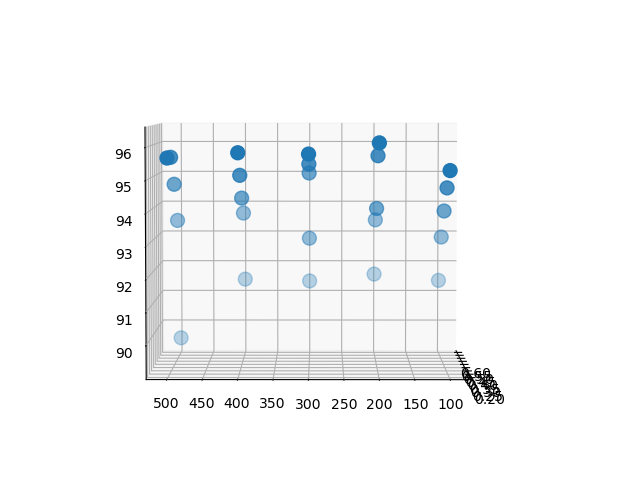


Рисунок 5 — вид графика спереди.

По нижней оси — количество узлов скрытого слоя.

По вертикальной оси — процент правильных ответов.

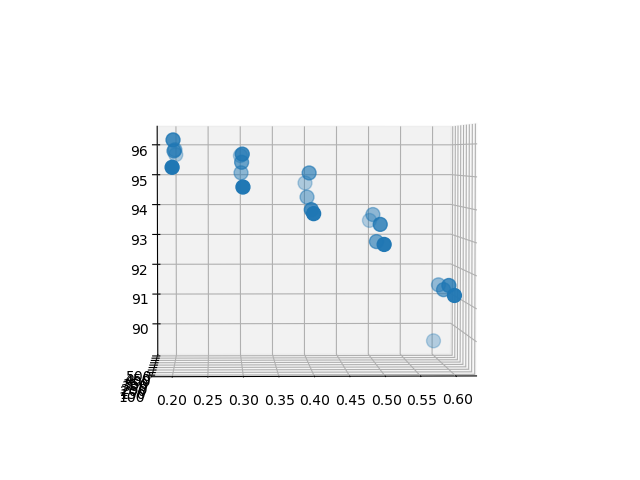


Рисунок 6 — вид графика сбоку.

По нижней оси — коофициент обучения.

По вертикальной оси — процент правильных ответов.

Представим данные также в виде таблицы (см. таблицу 1).

Из таблицы видно, что наилучшей комбинацией параметров будут коофициент обучения, равный 0,2, и 200 узлов скрытого слоя.

После замены нужных параметров сети и последующим обучением сети с перезаписью весов в файле количество правильных ответов достигло 96,1%.

Теперь стоит провести эпохальную тренировку. Трёх эпох должно быть достаточно.

Данные эпохальной тренировки представлены в таблице 2.

Больше на имеющихся данных сеть тренировать нельзя. Нужно использовать другую базу примеров или создавать свою, что требует отдельного исследования. Сейчас сеть натренирована достаточно, чтобы корректно работать. Мы пришли к конечной цели.

**Заключение**

В результате прохождения учебной (ознакомительной) практики было проведено:

- ознакомление с такой научной областью, как нейронные сети,

- ознакомление с государственными требованиями к оформлению научно-технической документации.

Выполнены следующие задачи учебной (ознакомительной) практики:

- сформирован реферат по выбранной научной теме «Нейронные сети»,

- сделан публичный доклад по выбранной теме, - изучен ГОСТ 7.32-2017 «Отчёт о научно-исследовательской работе. Структура и правила оформления»,

- выполнено индивидуальное задание, которое заключается в том, чтобы оформить реферат по выбранной научной теме в соответствии с требованиями ГОСТ 7.32-2017,

- осуществлено ознакомление со стандартами:

а) ГОСТ 2.105-95 «Единая система конструкторской документации. Общие требования к текстовым документам»,

б) ГОСТ 7.1-2003 «Библиографическая запись. Библиографическое описание. Общие требования и правила составления»,

в) ГОСТ Р 7.0.5-2008 «Библиографическая ссылка. Общие требования и правила составления»,

г) ГОСТ 7.82-2001 «Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления»,

д) ГОСТ 7.12-93 «Библиографическая запись. Сокращение слов на русском языке. Общие требования и правила»,

е) ГОСТ 8.417-2002 «Единицы величин»,

ж) ГОСТ 7.54-88 «Представление численных данных о свойствах веществ и материалов в научно-технических документах. Общие требования»,

и) ГОСТ 7.9-95 (ИСО 214-76) «Реферат и аннотация. Общие требования».

**Список литературы**

1. Петр Радько — электронный учебник “Нейронные сети” <https://neuralnet.info/book/>
2. Наталия Ефремова — расшифровка доклада [*Highload++*](http://www.highload.ru/) по теме “Нейронные сети: практическое применение” https://habr.com/ru/post/322392/
3. Статья “Нейронные сети для начинающих” <https://habr.com/ru/post/313216/>
4. Тарик Рашид “Создаем нейронную сеть” ООО “Альфа-книга”, 195027, Санкт-Петербург 2017. — 272 с.
5. Пол Берри “Изучаем программирование на Питон” Москва : издательство «Э», 2017. — 624 с.

**Приложение А**

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Московский государственный технический университет имени Н.Э. Баумана

(национальный исследовательский университет)»

(МГТУ им. Н.Э. Баумана)

Кафедра «Вычислительная математика и математическая физика» (ФН-11)

**ЗАДАНИЕ**

**на прохождение учебной практики**

на предприятии НОЦ «СИМПЛЕКС» МГТУ им. Н.Э. Баумана

Студент группы ФН11-23Б \_Зарина Ольга Юрьевна

Во время прохождения учебной (ознакомительной) практики студент должен:

1. изучить основные направления деятельности предприятия,

2. изучить государственные стандарты в области оформления научно-технической документации,

3. электронную версию готового отчёта по практике (формат Word) выслать в электронный архив кафедры (на адрес электронной почты [archive-fn@mail.ru](mailto:archive-fn@mail.ru) ) и разместить в «Портфолио студента».

Дата выдачи задания «29 » июня 2020 г.

Руководитель практики от кафедры\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / А.А. Прозоровский /

(подпись, дата)

**Студент**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /** О.Ю.Зарина **/**

(подпись, дата)