ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОРНОМИКИ»

*Московский институт электроники и математики им. А.Н.Тихонова*

Отчет по курсовой работе по теме «Разработка нейронной сети Розенблатта для распознавания букв английского языка»

Автор: Марусев Егор

Группа: БИВ 197

МОСКВА 2020

**Оглавление**

[**I. Техническое задание** 3](#_Toc39509526)

[**II. Введение** 5](#_Toc39509527)

[**III. Общий алгоритм решения** 5](#_Toc39509528)

[**IV. Обзор аналогичных программ** 5](#_Toc39509529)

[**V. Обоснование выбора программных средств** 5](#_Toc39509530)

[**VI. Описание программы** 6](#_Toc39509531)

[**VII. Тестовые примеры** 6](#_Toc39509532)

[**VIII. Приложение 1** 8](#_Toc39509533)

# 

# **I. Техническое задание**

1. Формулировка задания

1.1 Общая формулировка

Выбрать среду для разработки программы для создания нейронной сети для распознавания букв. Разработать данную программу в выбранной среде разработки.

1.2 Требования к функциональным характеристикам

Программа позволяет пользователю нарисовать собственными руками букву латинского алфавита и дать программе распознать ее и указать точность предсказания.

1.3 Требования к надежности

Надежность соответствует способности программы распознавать все возможные буквы и соответствует значению, определенному на тесте на большой выборке данных. В данном случае надежность составляет 85%.

1.4 Требования к составу технических средств

Аналогичны требованиям к составу технических средств для используемой пользователем среды разработки, поддерживающей выбранный исполнителем язык программирования.

2. Спецификация входных и выходных данных

2.1 Перечень и требования к входным данным

- Нарисованный символ должен быть строчной или заглавной латинской буквой;

- Нарисованная буква может быть нарисована не идеально, но должна быть понятна пользователю.

-Нарисованная буква должна быть во все выделенное поле, для лучшего распознавания сетью.

2.2 Перечень и требования к выходным данным

Выходными данными является заглавная буква, которую определила программа.

-Определенная буква должна соответствовать нарисованной.

-Точность предсказания должна быть больше 50%.

3. Требования к программной документации

В документацию будут включены:

1)Описание алгоритма работы программы

2)Листинг программы

3)Тестовые примеры

4)Руководство пользователя

4. Этапы и сроки выполнения работ

Таблица 1.

Этапы и сроки выполнения работ.

|  |  |  |
| --- | --- | --- |
| 1 | Выбор темы курсовой работы | 15.11.2019 |
| 2 | Составление задания на курсовую работу | 20.12.2019 |
| 3 | Проект программы - первая версия ПО | 28.02.2020 |
| 4 | Проект документации - первая версия отчета | 30.03.2020 |
| 5 | Окончательная версия ПО | 15.05.2020 |
| 6 | Отчет и составление программной документации | 30.05.2020 |

**II. Введение**

Темой курсовой работы была выбрана «Разработка нейронной сети Розенблатта для распознавания букв английского языка», поскольку данное направление очень актуально сейчас. Данное программное обеспечение на данном этапе развития имеет очень узкое применение, но при будущем усовершенствовании может применяться для перевода рукописного текста в печатный формат.

**III. Общий алгоритм решения**

Алгоритм(используемый для пользователя):

* Пользователь рисует букву в специальном поле
* Поле переводится в изображение формата .jpg в черно-белом цвете
* Полученное изображение переводится в матрицу (28,28), где значение в каждой ячейке соответствует темноте цвета (0 -255), где эти значения переводится в диапазон от 0.01 до 0.99.
* Марица переводится в массив (784,), который подается на вход нейронной сети. Входные данные опрашиваются нейронной сетью и на основе весовых коэффициентов, определенными в тренировке, программа выдает маркер на определенную букву.

(подробнее о работе опроса, тренировки и теста нейронной сети в приложении 1)

# **IV. Обзор аналогичных программ**

Рассмотрим аналогичную программу моего одногруппника Кирилла Шайкина ( aka kiripace). Его тема: «Разработка нейронной сети Розенблатта для распознавания цифр». Так как нейронная сеть не отличается от моей по своему типу, работы могут отличаться только эффективностью работы и способом работы с введенным изображением.

-Работа с изображением надежна в обоих случаях и корректно переводит введенный пользователем символ. Принцип работы одинаковый.

-Эффективность работы у моей работы меньше(85% < 94%), так как больше распознаваемых символов (26 >10). Также буквы имеет более сложную форму, что может затруднить распознавание похожих символов.

# **V.** **Обоснование выбора программных средств**

Для разработки программного обеспечения была выбрана среда разработки «Jupyter Notebook» из дистрибутива «Anaconda Navigator», поскольку данная среда распространяется бесплатно официальным издателем и удобна в использовании, также туда входят многие библиотеки необходимые для работы.

Для написания программного обеспечения был выбран язык программирования «Python», поскольку данный язык очень прост в использовании и к нему написано множество библиотек, которые очень упрощают разработку нейронной сети.

# **VI.** **Описание программы**

Программа состоит из 4 частей:

1) Класс нейронной сети

2) Часть, отвечающая за тренировку сети

3) Часть, тестирующая работу сети

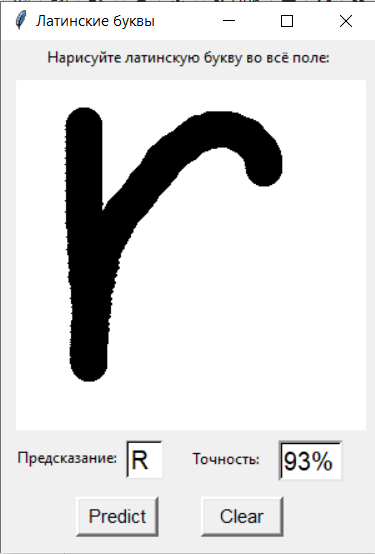
4) Часть, в которой мы работаем с нарисованной буквой пользователя

(пользователю нужна только 4 часть)

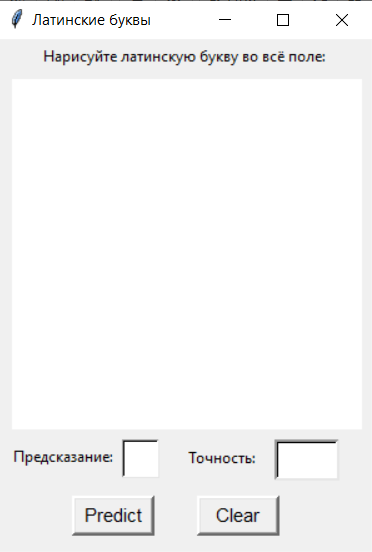
Подробнее о каждой части в Приложении 1.

**VII. Тестовые примеры**

1) Пользователь рисует букву и жмет кнопку «Predict». Он видит предсказание и его точность.



2)Пользователь может очистить поле, нажав кнопку «Clear» для последующего ввода.

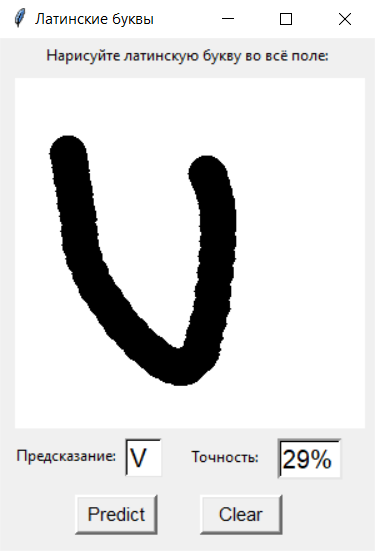


3)



4) Программа может ошибаться при не однозначном написании буквы.

(U или V)



# **VIII. Приложение 1**

Алгоритм

1.1 Нейронной сеть

Нейронная сеть состоит из 3 слоев узлов:

1 слой – входные данные. Так как мы обрабатываем изображения 28х28, то входной слой имеет 784 узла (28\*28). Значение в каждой ячейки – значение цвета(от 0 до 255), приведенное в диапазон от 0 до 1, не включая 0, чтобы избежать нулевых входных данных и не включая 1, так как значение сигмоиды никогда не достигает 1. ( о ней смотреть далее)

2 слой – сердце нейронной сети, где и происходит обучение (скрытый слой). Может иметь различное кол-во узлов, в зависимости от задачи, но как правило их больше, чем в 1 слое. Узлы во 2 слое принимают значения узлов из первого слоя, умноженные на весовые коэффициенты, определенные в тренировке и сглаженные сигмоидой. Сигмоида -функция, которая принимает значения от 0 до 1, асимптотически приближаясь к своим границам. Нужна, чтобы избежать выхода из выбранного нами диапазона.

3 слой – выходные данные или маркеры. Так как мы распознаем латинские буквы, то узлов в 3 слое будет 26. По номеру узла, который в итоге будет иметь наибольшее значение, можно понять какую букву распознала сеть. (Номер узда соответствует номеру буквы в алфавите)

1.2 Класс нейронной сети

- При создании новой нейронной сети автоматически с помощью метода \_\_init\_\_ определяются кол-во входных, скрытых и выходных узлов, а также коэффициент обучения сети. Выбираются случайным образом значение весов между 1-2 и 2-3 слоем ( от 0 до 1).

Также класс имеет два метода:  
-Train

В этом методе происходит тренировка сети. На вход подаются список входных данных и целевых значений (правильных ответов). Сперва производится проход через слои в одну сторону, посредством умножения значений узлов предыдущего слоя на весовые коэффициенты между этими слоями и сглаживается сигмоидой.

Дальше происходит корректировка весов между слоями в зависимости от близости значений на 3 слое и правильному ответа. Улучшение весовых коэффициентов производится путем уменьшения функции ошибки небольшими шагами. Каждый шаг совершается в направлении градиентного спуска.

-query

В этом методе мы просто опрашиваем сеть и узнаем какой результат она нам выдаст. На вход подаются только входные значения, а принцип работы такой же как и в первой части тренировки.

2. Тренировка сети

В этой части программы мы подаем большой набор данных на метод train.

Большой набор данных берется из csv файла(1 значение в строчке – правильный ответ, остальные значения в строчке – входные значения), переводится в матрицу. Входные значения переводятся в нужный диапазон, а целевой список создается посредством взятия массива размера 3 слоя и максимизации узел правильного ответа (0.99). Остальные – 0.01.

3. Тест сети

В этой части программы мы подаем большой набор данных на метод query.

Другой большой набор данных опрашивается сетью и фиксируется правильность ответов. Из правильных ответов считают процент эффективности.

4. Перевод введенного изображения.

Здесь создается поле для работы с пользователем с помощью возможностей библиотеки Tkinter. Создаются кнопки для работы с сетью располагаются на определенных местах поля. Метод работы с изображением описан ранее.