ГУАП

КАФЕДРА № 33

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | Н.С. Красников |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2 |
| РАЗРАБОТКА ОДНОСЛОЙНОЙ НЕЙРОННОЙ СЕТИ ПРЯМОГО РАСПРОСТРАНЕНИЯ СИГНАЛА |
| по курсу: |
| ОСНОВЫ МАШИННОГО ОБУЧЕНИЯ |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ гр. № | 3131 |  |  |  | А.П. Чирков |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2024

Цель работы

Изучение свойств нейрона (персептрона Розенблатта). Приобретение навыков разработки однослойных нейронных сетей в объектно-ориентированных средах программирования.

Задание к лабораторной

1. Изучить свойства персептрона Розенблатта и нейронной сети прямого распространения сигнала.
2. Создать входные данные (файл с весовыми коэффициентами нейрона) для обучения нейрона со случайными значениями.
3. В программной среде MS Visual Studio (или иной объектноориентированной IDE) создать нейрон, реализующий функцию распознавания заданного объекта датасета в соответствии со своим вариантом из лабораторной работы №1 (изображения объектов загружаются из файлов, созданных самостоятельно).
4. Обучить нейрон, используя правило Хебба. Выполнить обучение нейрона для классификации векторов (изображения объекта датасета) на две категории (верно/не верно).
5. Создать слой нейронов и выполнить имитацию работы однослойной нейронной сети для всех классов датасета, созданного в лабораторной работе №1.
6. График изменения loss-функции в ходе обучения сети следует выводить на отдельной форме приложения и, при необходимости, иметь возможность его со хранить.
7. Рассчитать метрики качества классификации, созданной однослойной сети как на обучающей, таки и на тестовых выборках.
8. Изучить возможности однослойных нейронных сетей решать линейно несепарабельные задачи.
9. Реализовать (при необходимости) многослойный персептрон, обеспечивающий после обучения значение метрики Accuracy не ниже 0,95 для каждого класса.
10. Оформить отчет по лабораторной работе.

Ход работы

1. Подготовка данных для обучения

Подготовленный датасет «annotations.cs» из лабораторной работы №1 был разбит на две выборки: обучающую и валидационную в соотношении 80/20. Разбиение было сделано так, что из 1082 изображений 865 присваивались в тренировочную выборку, а 217 в валидационную.

1. Описание созданной программы

На языке Python 3.10.12 была создана программа с графическим интерфейсом для обучения нейросети на основе подготовленных данных.

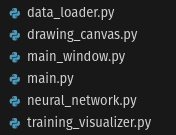


Рисунок 1 -Архитектура программы

Описание модулей программы:

* 1. Main.py:

Является точкой входа в приложение. Инициализирует PyQt для работы с графическим интерфейсом и создает основные объекты (загрузчик данных, нейронную сеть, главное окно графического интерфейса).

* 1. Neural\_network.py:

Реализует многослойный перцептрон, методы прямого и обратного распространения, функции активации (Relu, Softmax), сохранение и загрузку весов, а также методы для обучения и предсказания.

* 1. Data\_loader.py:

Загружает данные из csv файла. Выполняет предобработку данных и разделяет их на обучающую и валидационную выборки.

* 1. main\_window.py:

Является главным окном графического интерфейса, управляет параметрами обучения, содержит кнопки управления и интегрирует все компоненты GUI.

* 1. drawing\_canvas.py:

Представляет собой область для рисования цифр, преобразует рисунки в массив данных.

* 1. training\_visualizer.py:

Содержит графики процесса обучения (loss, accuracy, precision, recall) и обновляет их в реальном времени.

Взаимодействие компонентов:

* + GUI позволяет настраивать параметры и управлять процессом обучения;
  + Обучение выполняется в отдельном потоке (Training Thread);
  + Результаты визуализируются в реальном времени;
  + Пользователь может рисовать цифры и получать предсказания от обученной модели.

Архитектура нейронной сети представляет собой:

* 1. Входной слой:

Размер равен количеству пикселей входного изображения — 784 нейрона (28x28 пикселей).

* 1. Скрытые слои:
     + Первый скрытый слой: 512 нейронов;
     + Второй скрытый слой: 256 нейронов;
     + Третий скрытый слой: 128 нейронов.
  2. Выходной слои:

Размер равен количеству классов — 10 нейронов (цифры 0 — 9)

Для скрытых слоев используется функция активации ReLU, а для выходного слоя Softmax.

Графический интерфейс приложения представлен на рисунке 2:

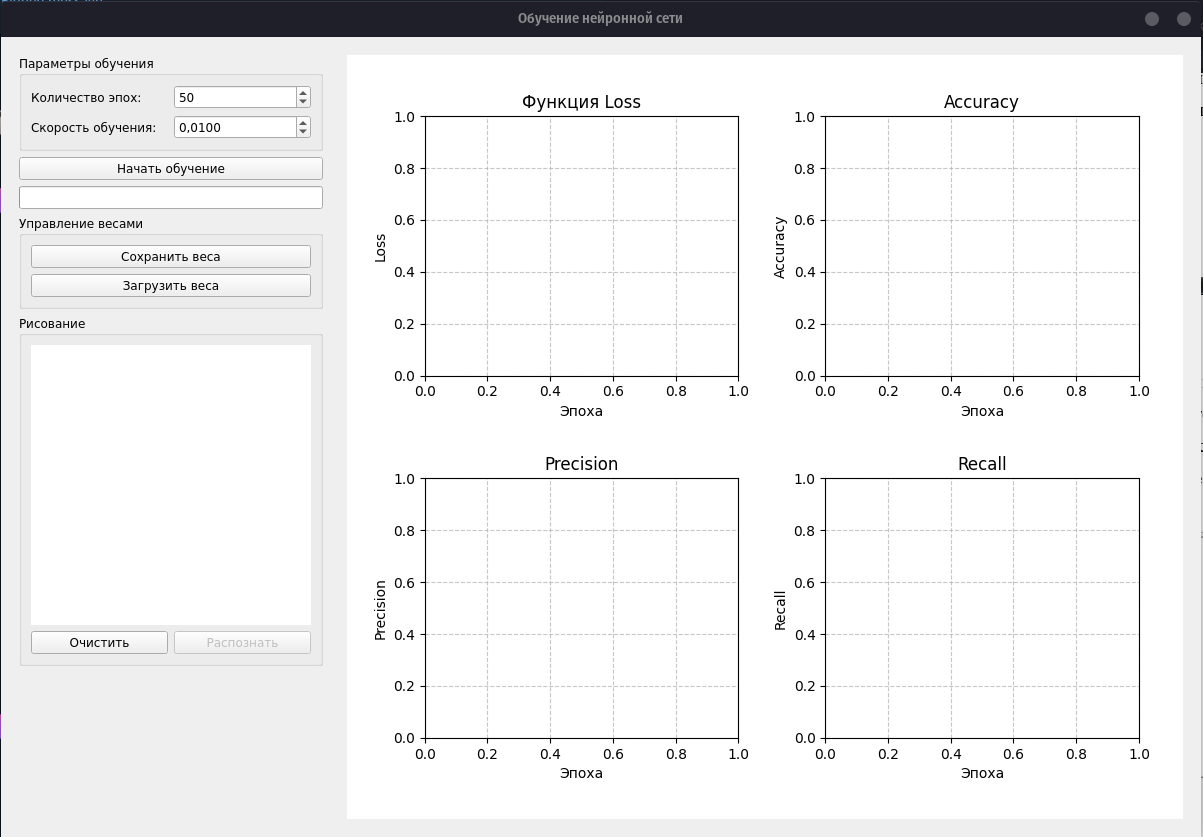


Рисунок 2 — Интерфейс приложения

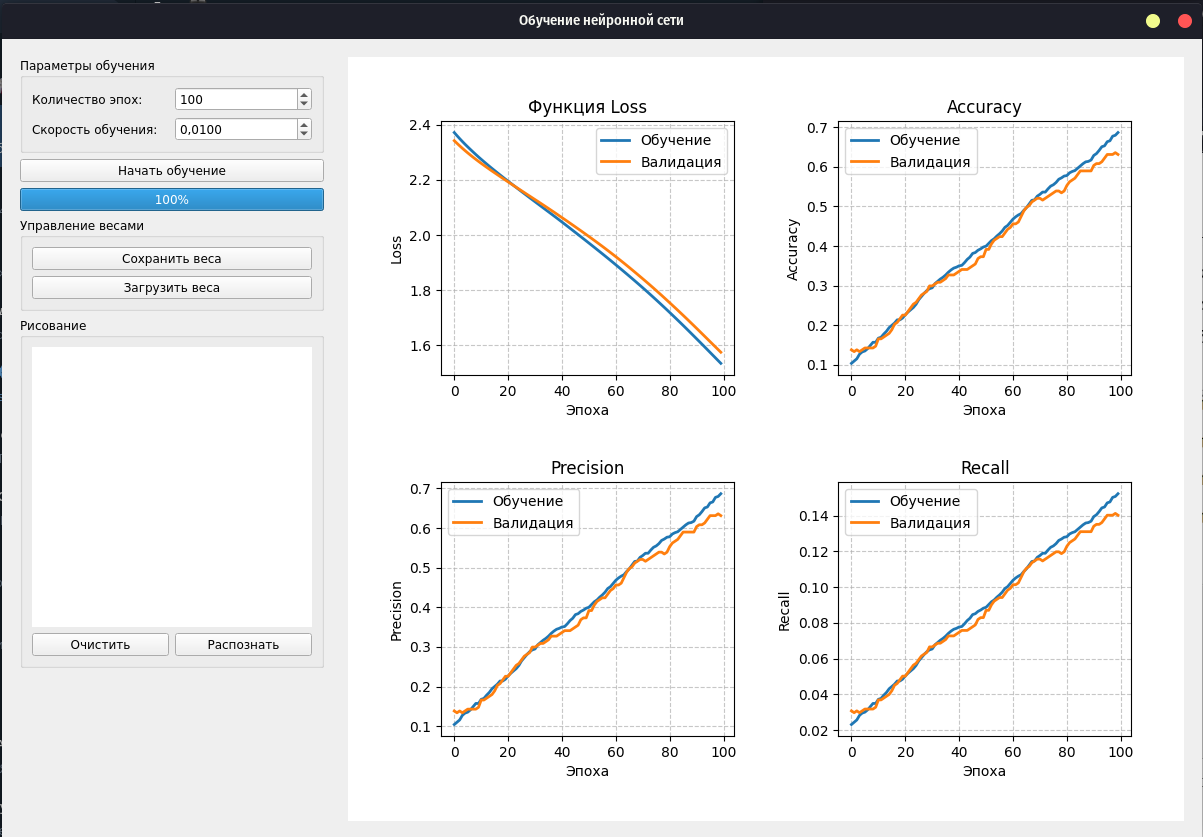
Пользователь может настроить количество эпох обучения и скорость обучения. Также можно загрузить уже существующие веса, либо сохранить новые после обучения нейросети.

1. Проверка работы программы

Запустим обучения со следующими параметрами:

* + Количество эпох = 100;
  + Скорость обучения = 0,01.

Результаты обучения представлены на рисунке 3:

Рисунок 3 — Результаты обучения

После завершения обучения пользователь может посмотреть на графики метрик. Сами значения метрик выводятся в консоль приложения.

Сеть обучалась 100 эпох, начальные значения метрик были низкие:

* + Тренировочные точность и loss: 0,1040 (10,4%) и 2,3720;
  + Валидационные точность и loss: 0,1382 (13,8%) и 2,3425.

Конечные значения значительно улучшились:

* + Тренировочные точность и loss: 0,6867 (68,7%) и 1,5352;
  + Валидационные точность и loss: 0,6313 (63,1%) и 1,5755.

Loss постоянно уменьшается, что говорит об успешном обучении, но можно было бы и выбрать большее количество эпох для повышения качества обучения.

К концу обучения появляется небольшой разрыв между тренировочными и валидационными метриками, однако, разрыв не критичный, что говорит об отсутствии сильного переобучения.

Несмотря на низкий показатель точности, нейросеть может определить некоторые нарисованные от руки цифры (Рис. 4-5).

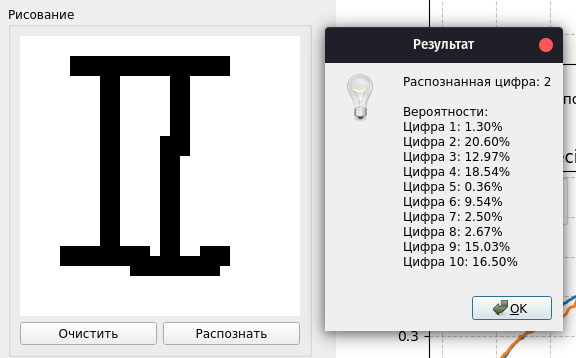


Рисунок 4 — Определение цифры 2

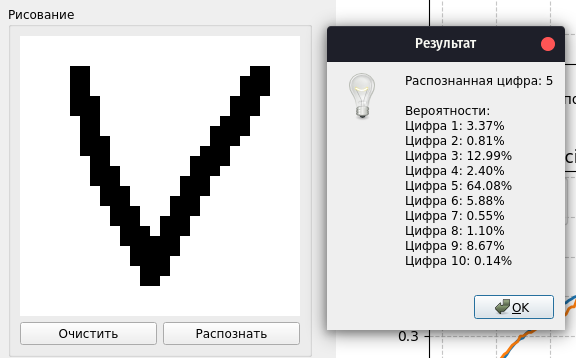
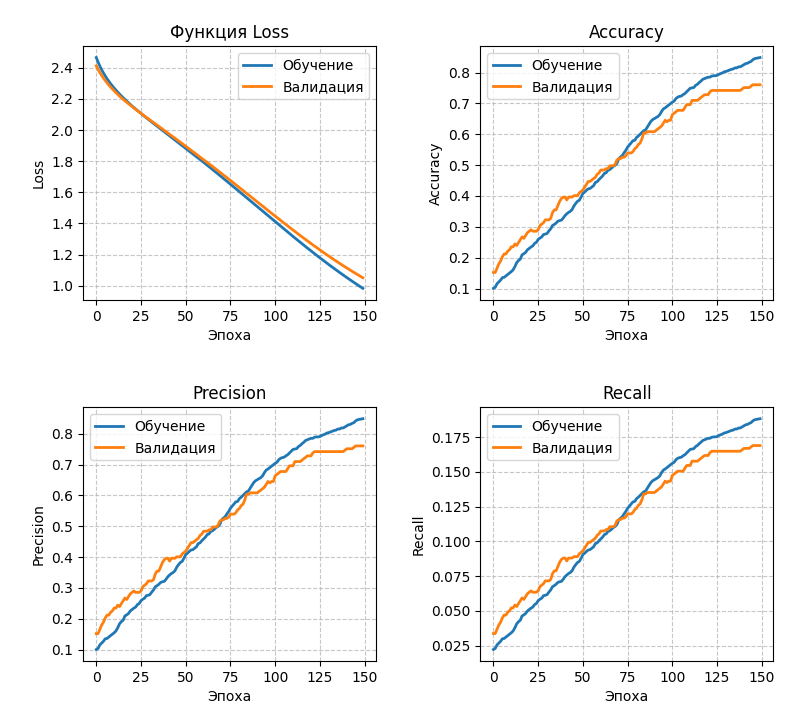


Рисунок 5 — Определение цифры 5

Попробуем задать другие параметры для получения более высокой точности, увеличив количество эпох до 150.

Получили следующие графики:

Рисунок 6 — Графики для 150 эпох

Показатели метрик в конце обучения составили 85% тренировочной и 76% валидационной точности. Результаты распознавания представлены на рисунках 7-16.

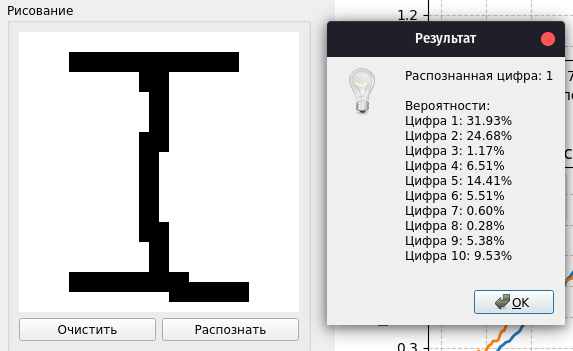


Рисунок 7 —Распознавание цифры 1

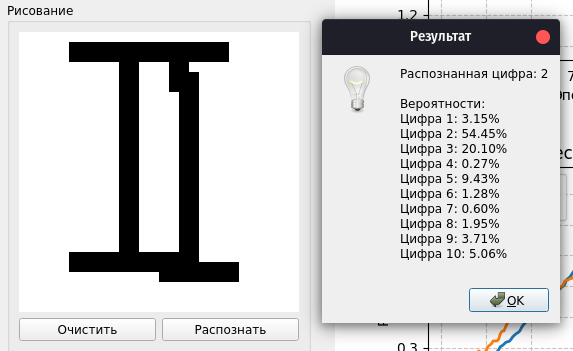


Рисунок 8 — Распознавание цифры 2

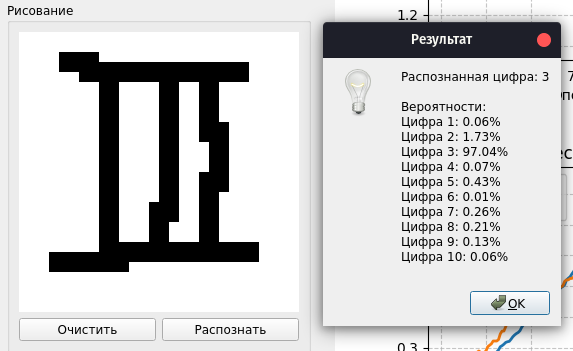


Рисунок 9 — Распознавание цифры 3

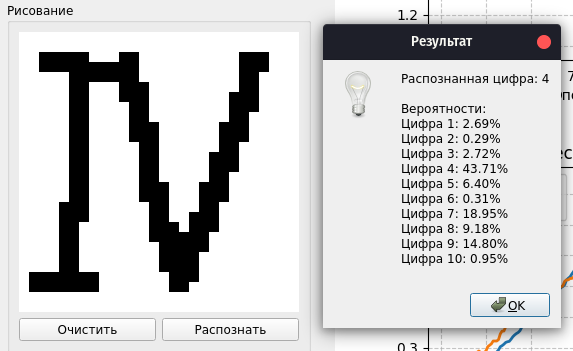


Рисунок 10 — Распознавание цифры 4

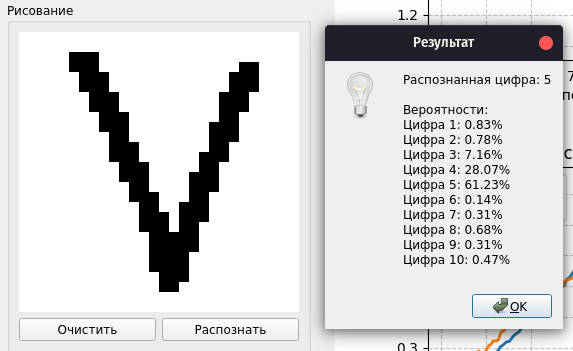


Рисунок 11 — Распознавание цифры 5

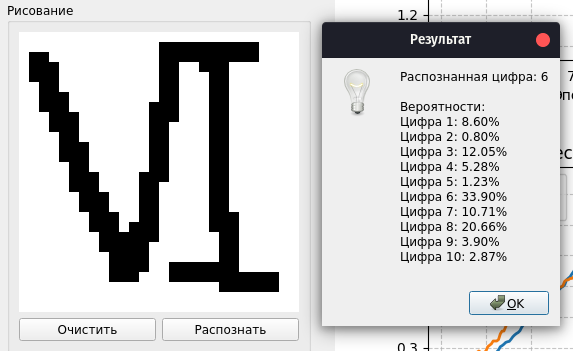


Рисунок 12 — Распознавание цифры 6

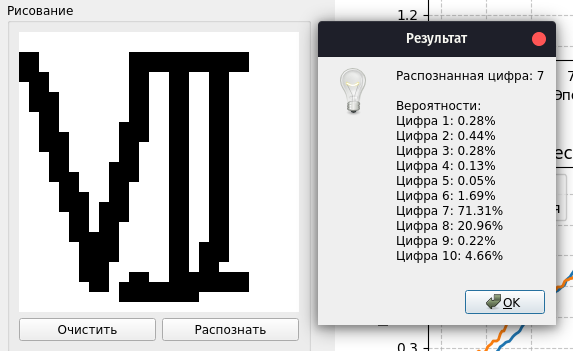


Рисунок 13 — Распознавание цифры 7

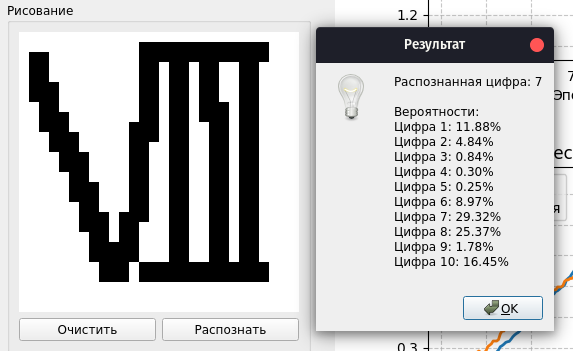


Рисунок 14 — Распознавание цифры 8

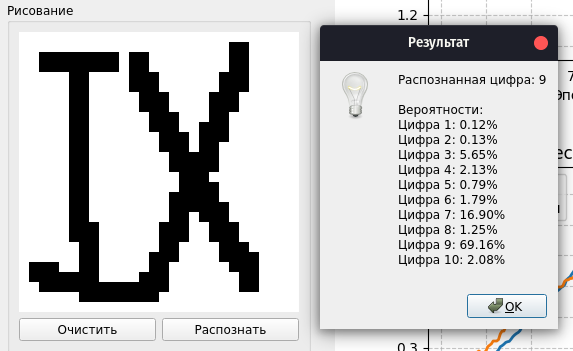


Рисунок 15 — Распознавание цифры 9

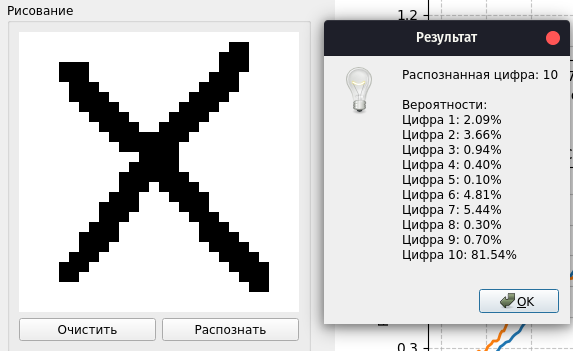
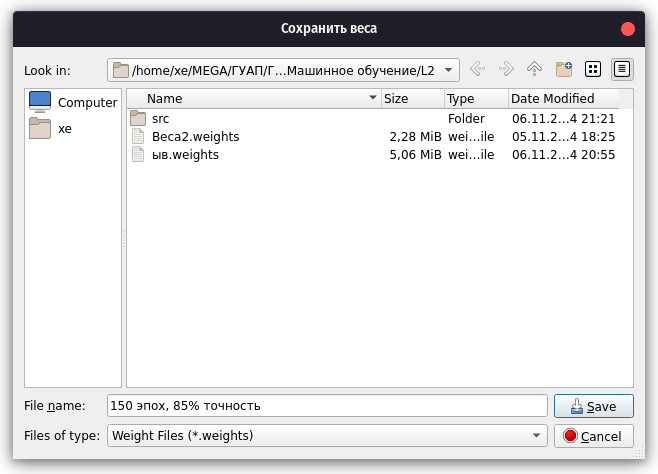


Рисунок 16 — Распознавание цифры 10

Как видно из результатов, наблюдаются небольшие проблемы с распознаванием цифры 8, но, судя по процентам вероятности распознавания, программа почти определила, что эта цифра 8.

Сохраним веса для использования их в дальнейшем

Рисунок 17 — Сохранение весов

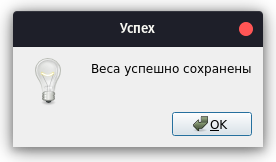


Рисунок 18 — Веса успешно сохранены

Вывод