

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №4**  
**по дисциплине «Искусственные нейронные сети»**  
**Тема: Распознавание рукописных символов**

Студент гр. 8383

\_\_\_\_\_

Сахаров В.М.

Преподаватель

\_\_\_\_\_

Жангиров Т.Р.

Санкт-Петербург

2021

### **Цель работы.**

Реализовать классификацию черно-белых изображений рукописных цифр (28x28) по 10 категориям (от 0 до 9).

### **Задачи.**

1. Ознакомиться с представлением графических данных
2. Ознакомиться с простейшим способом передачи графических данных нейронной сети
3. Создать модель
4. Настроить параметры обучения
5. Написать функцию, позволяющая загружать изображение пользователя и классифицировать его

### **Ход работы.**

Из tensorflow был загружен датасет mnist (изображения рукописных цифр). Тестовые данные были нормализованы для корректной обработки.

Была выполнена модель из методических материалов, её точность составила 98.5%

Были исследованы все доступные оптимизаторы, далее приведены точности при использовании их с разными параметрами:

- SGD (стохастический градиентный спуск, хорош для небольших сетей. Быстрее работает с моментами Нестерова, так как получает ускорение от результата предыдущих эпох), по умолчанию = 91.2%
- RMSprop (усовершенствованный Adagrad, более стабильное движение на значения среднего квадрата градиентов, делает движение плавным), по умолчанию = 97.7%

- Adam (RMSprop с моментами, получаемыми из предыдущих эпох, самый популярный), по умолчанию = 97.6%
- Adadelatа (усовершенствованный Adagrad, решает проблему замедления скорости обучения у алгоритма Adagrad), по умолчанию = 58.7%
- Adagrad (базируется на параметре частоты обновлений, хорошо работает на редких, данных с большими разбросами, однако, скорость обучения сильно замедляется), по умолчанию = 87.5%
- Adamax (борется с шумами на кривой обучения лучше, чем Adam), по умолчанию = 96.6%
- Nadam (Adam с моментами Нестерова вместо простых моментов, те же преимущества, что и в SGD), по умолчанию = 97.5%
- Ftrl, по умолчанию = 20.9%

Исходя из полученных данных, были более подробно протестированы алгоритмы RMSprop, Adam и Nadam:

- RMSprop(learning\_rate=0.001, rho=0.9, centered=True) = 98.5%
- RMSprop(learning\_rate=0.0001, rho=0.9, centered=True) = 94.6%
- RMSprop(learning\_rate=0.01, rho=0.9, centered=True) = 98.1%
- RMSprop(learning\_rate=0.001, rho=0.99, centered=True) = 98.6
- RMSprop(learning\_rate=0.001, rho=0.999, centered=True) = 98.7%
- Adam(learning\_rate=0.01, beta\_1=0.9, beta\_2=0.999) = 98.0%
- Adam(learning\_rate=0.0001, beta\_1=0.9, beta\_2=0.999) = 94.0%
- Adam(learning\_rate=0.01, beta\_1=0.99, beta\_2=0.99) = 98.2%
- Nadam(learning\_rate=0.01, beta\_1=0.9, beta\_2=0.999) = 98.0%
- Nadam(learning\_rate=0.0001, beta\_1=0.9, beta\_2=0.999) = 94.0%
- Nadam(learning\_rate=0.01, beta\_1=0.99, beta\_2=0.999) = 98.5%
- Nadam(learning\_rate=0.01, beta\_1=0.999, beta\_2=0.999) = 99.3%

- $\text{Nadam}(\text{learning\_rate}=0.01, \text{beta\_1}=0.999, \text{beta\_2}=0.9999) = 99.4\%$

Исходя из этого, алгоритм Nadam оказался наиболее применим для данного набора данных, финальная кривая обучения представлена на рисунке 1.

Первые 25 изображений, распознанных неправильно, представлены на рисунке 2.

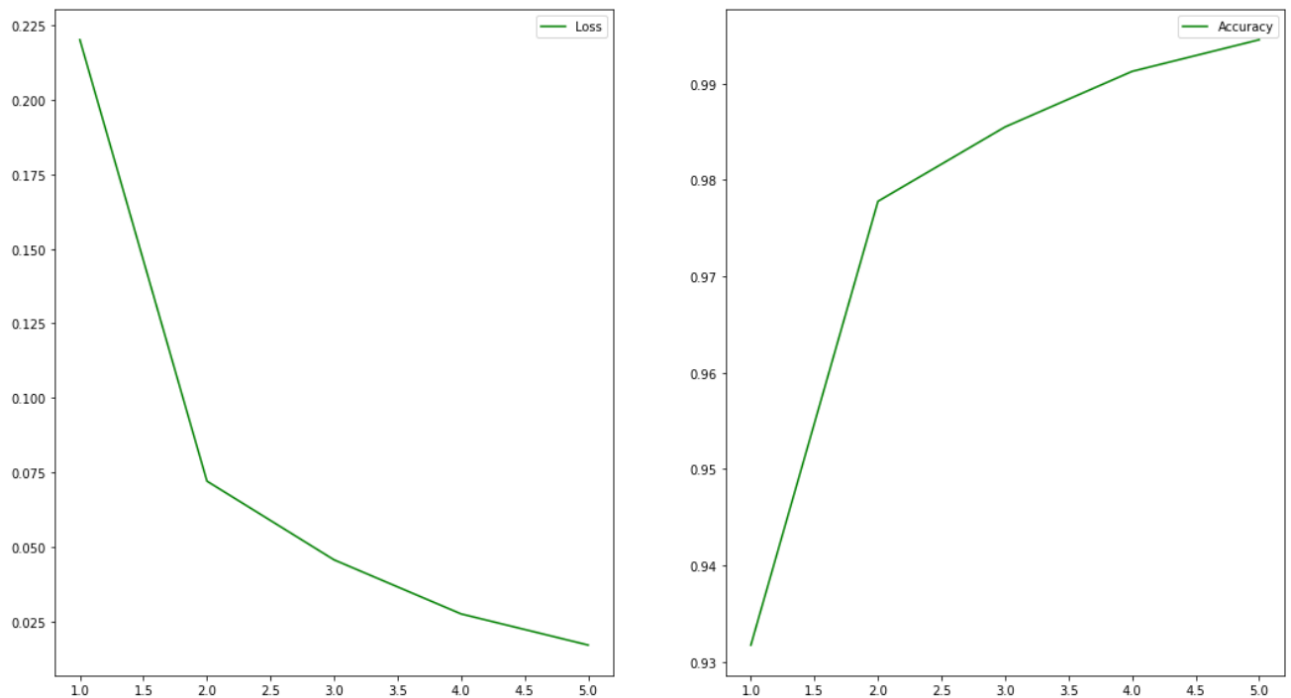


Рисунок 1 – графики кривых обучения

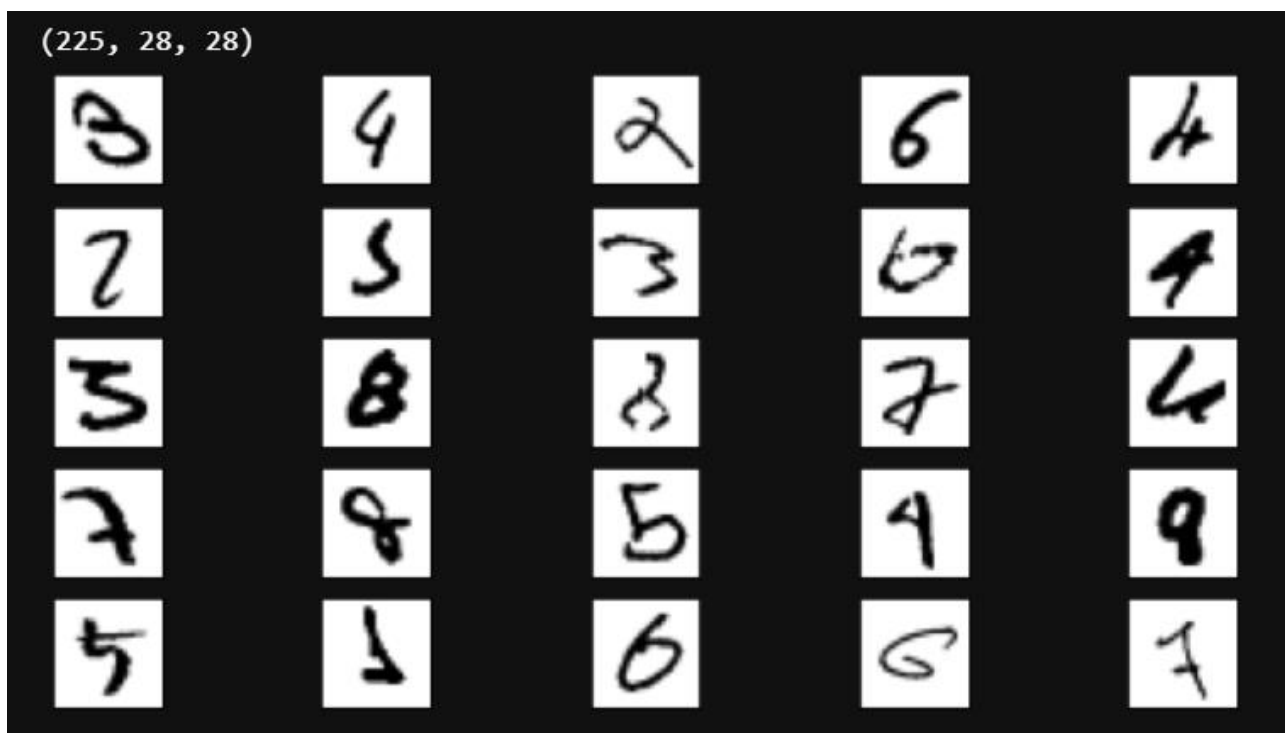


Рисунок 2 – неправильно распознанные рисунки

Также был реализован ввод собственных изображений, пример исходного изображения представлен на рисунке 3, пример вывода программы представлен на рисунке 4.



Рисунок 3 – собственное изображение

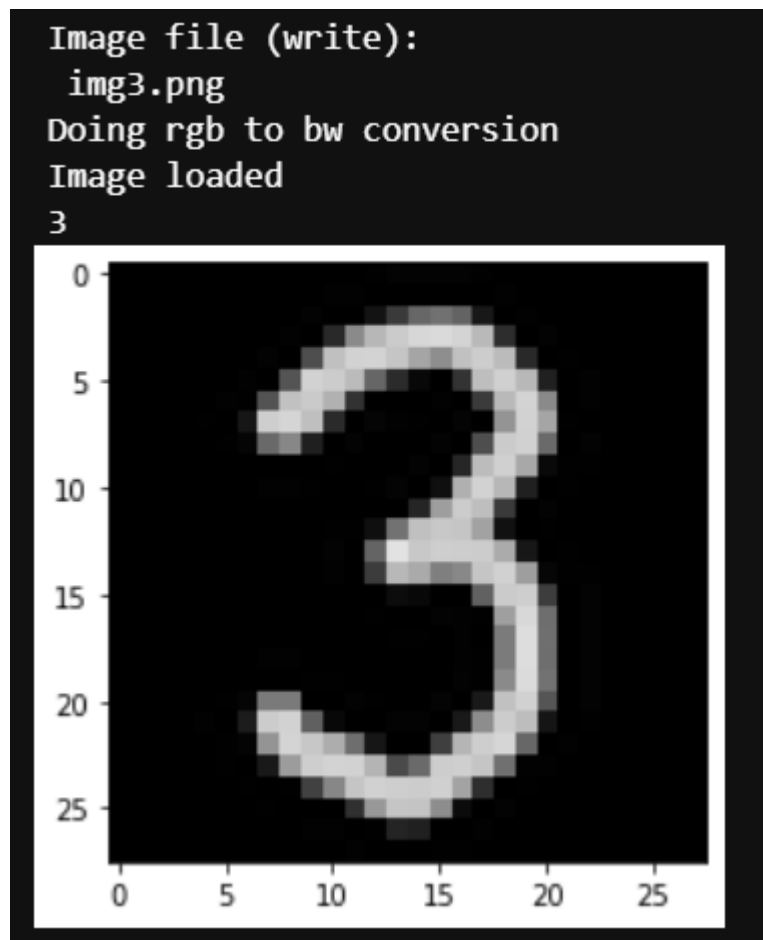


Рисунок 4 – распознавание собственного изображения

### **Выводы.**

В ходе выполнения работы была реализована классификация черно-белых изображений рукописных цифр. Было исследовано влияние оптимизаторов с разными параметрами на точность работы нейросети, а также был реализован интерфейс ввода пользовательского изображения.