

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. 7381

Павлов А.П.

Преподаватель

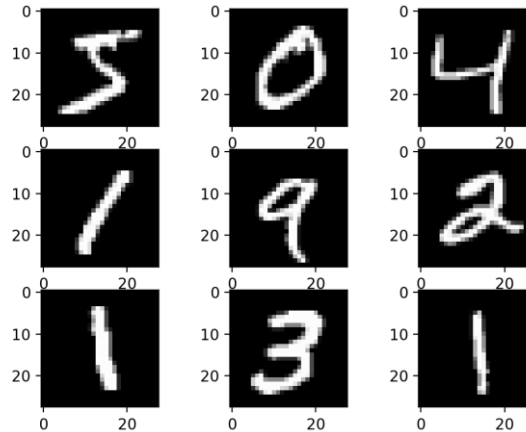
Жукова Н.А..

Санкт-Петербург

2020

Цель работы.

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



Набор данных содержит 60,000 изображений для обучения и 10,000 изображений для тестирования.

Задачи.

- Ознакомиться с представлением графических данных
- Ознакомиться с простейшим способом передачи графических данных нейронной сети
- Создать модель
- Настроить параметры обучения
- Написать функцию, позволяющая загружать изображение пользователя и классифицировать его

Ход работы.

1. Найти архитектуру сети, при которой точность классификации будет не менее 95%.

Изначальная архитектура даёт точность 0.9778, поэтому оставим ее без изменений.

Сеть состоит из 4-ех слоев:

- Входной, заданный неявно при вызове конструктора

- Flatten, преобразующий двумерное входное изображение в одномерный вектор
- Скрытый слой (256 нейронов)
- Выходной слой (10 нейронов)

Графики точности и ошибки представлены на рис.1–2.

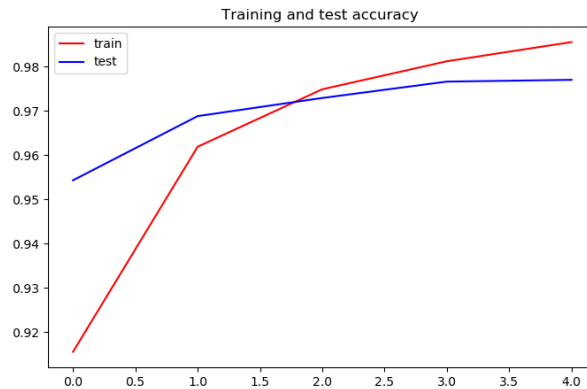


Рисунок 1 – График точности построенной сети

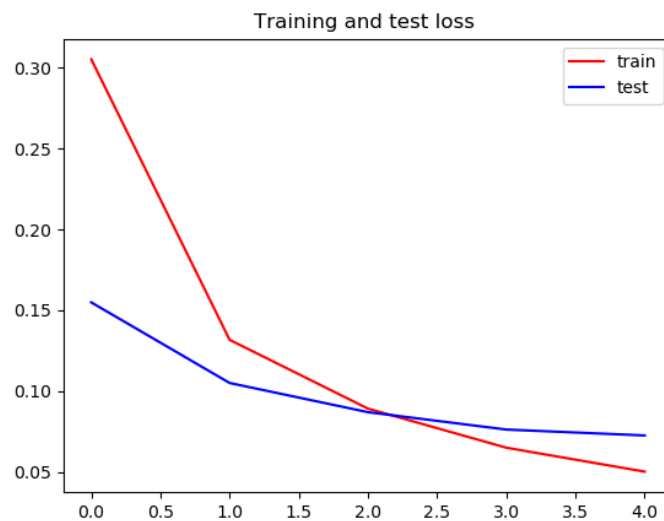


Рисунок 1 – График ошибок построенной сети

2. Исследовать влияние различных оптимизаторов, а также их параметров, на процесс обучения.

- Оптимизатор SGD

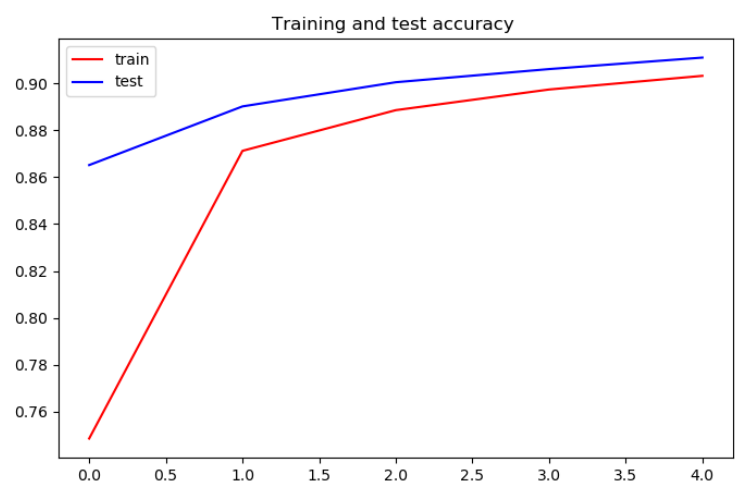


Рисунок 3 – График точности SGD

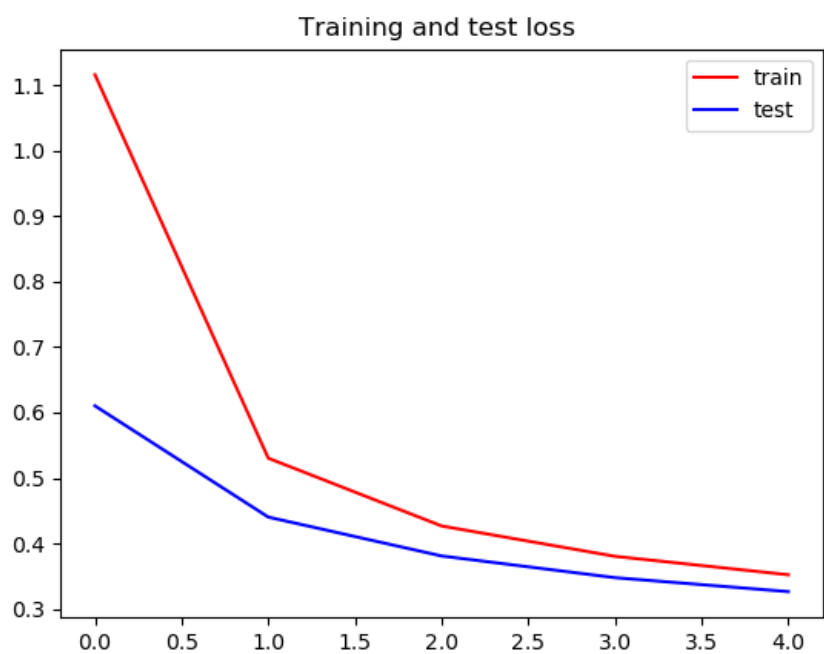


Рисунок 4 – График ошибки SGD

- Оптимизатор RMSprop

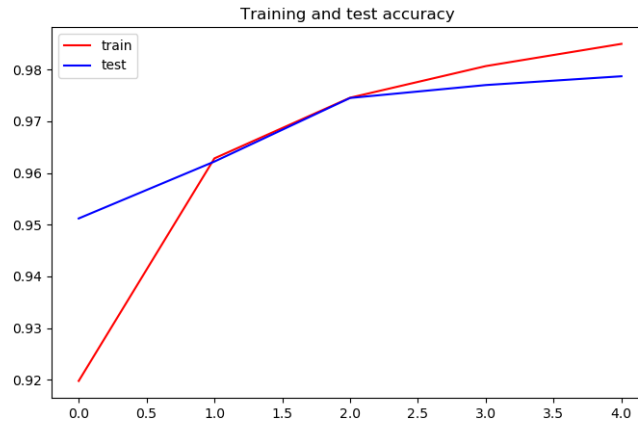


Рисунок 5 – График точности RMSprop

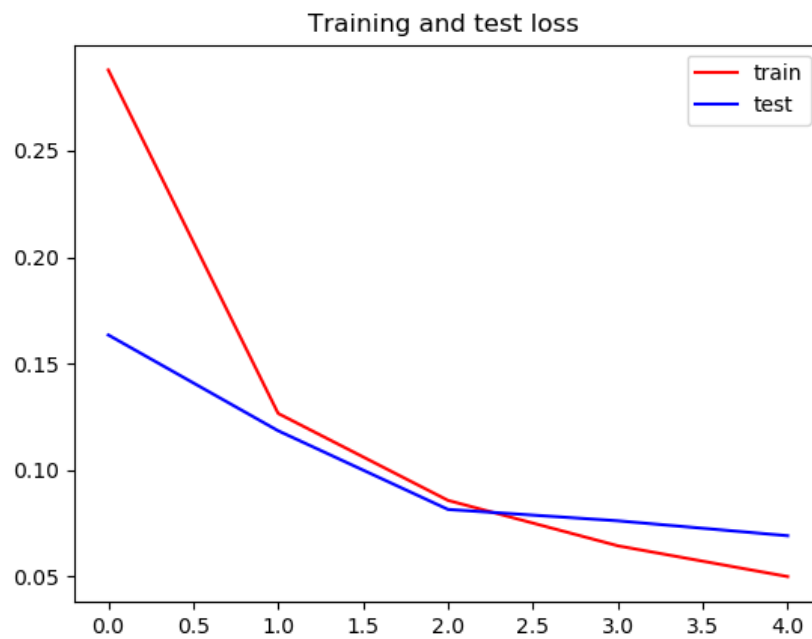


Рисунок 6 – График ошибки RMSprop

- Оптимизатор Adagrad

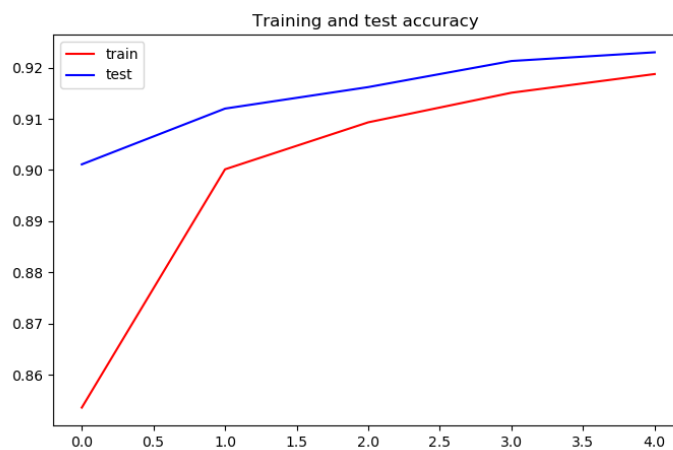


Рисунок 7 – График точности Adagrad

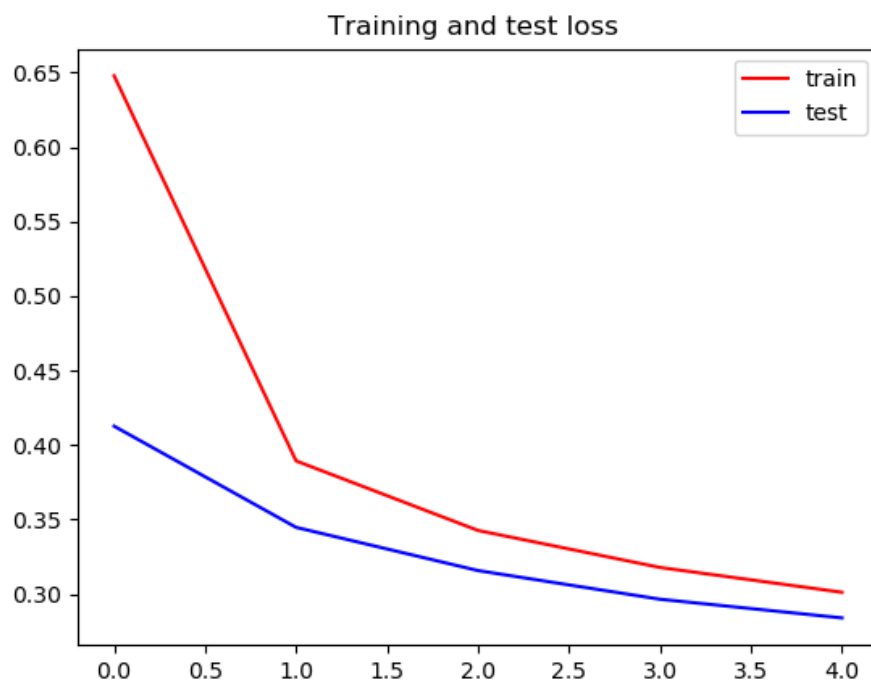


Рисунок 8 – График ошибки Adagrad

- Оптимизатор Nadam

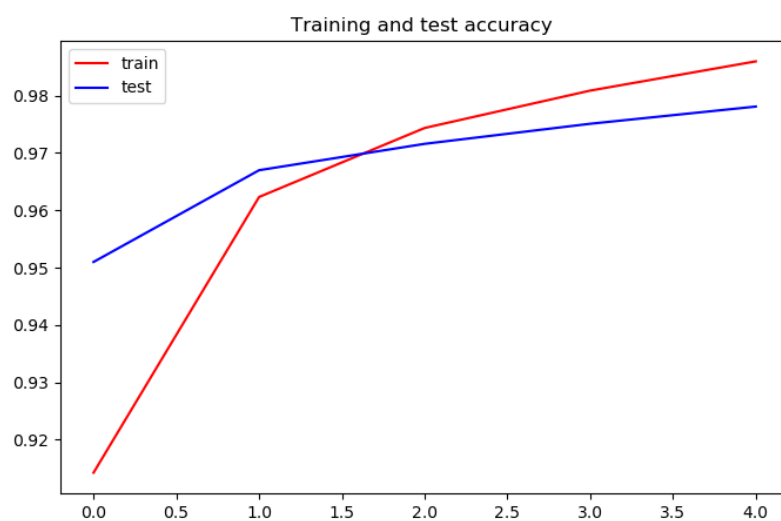


Рисунок 9 – График точности Nadam

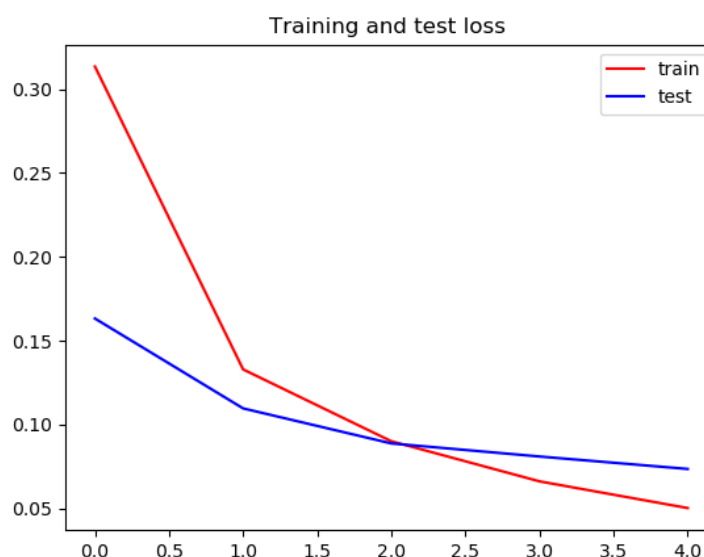


Рисунок 10 – График ошибки Nadam

Оптимизаторы Nadam и RMSprop дали точность примерно 98%, остальные же Adagrad и SGD показали точность гораздо хуже около 91%.

3. Написать функцию, которая позволит загружать пользовательское изображение не из датасета.

Была написана функция, в которую необходимо передать путь к изображению. Функция представлена ниже:

```
def loadImage(filename):
    image = Image.open(filename)
    image = image.resize((28, 28))
    image = np.dot(np.asarray(image), np.array([1/3, 1/3, 1/3]))
    image /= 255
    image = 1 - image
    image = image.reshape((1, 28, 28))
    return image
```

Выводы.

В ходе выполнения данной работы было изучено представление графических данных. Была построена и протестирована на пользовательских изображениях сеть с точностью $\approx 98\%$.

