

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. 8382

Мирончик П.Д.

Преподаватель

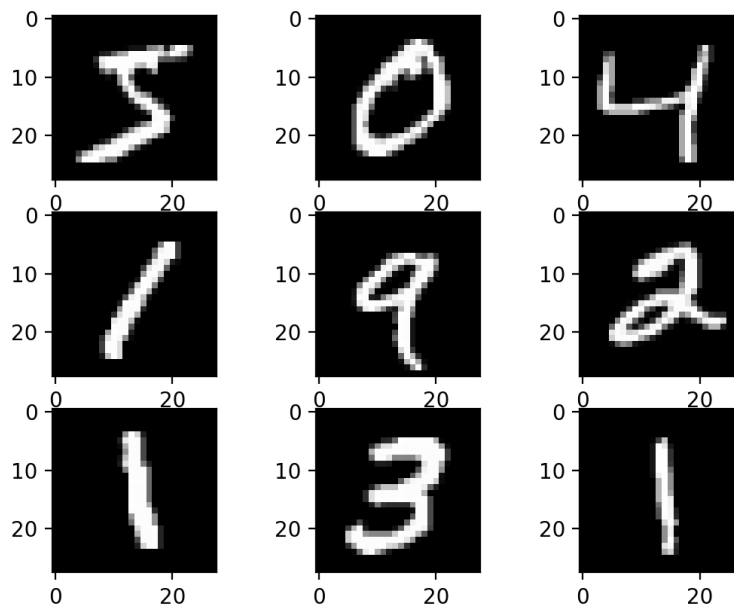
Жангиров Т.Р.

Санкт-Петербург

2021

ЗАДАНИЕ

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



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

ТРЕБОВАНИЯ

1. Найти архитектуру сети, при которой точность классификации будет не менее 95%
2. Исследовать влияние различных оптимизаторов, а также их параметров, на процесс обучения

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

ХОД РАБОТЫ

Начальная структура нейронной сети не требует внесения изменений,
т.к. выдаваемая точность классификации превышает требуемые 95%.

Далее приводится точность при использовании различных
оптимизаторов на 5 эпохах:

```
optimizersSGD = [  
    SGD(), # 0.91  
    SGD(learning_rate=0.001), # 0.82  
    SGD(learning_rate=0.1), # 0.96  
    SGD(momentum=0.1), # 0.91  
    SGD(momentum=0.6), # 0.93  
    SGD(momentum=0.9), # 0.95  
    SGD(nesterov=True), # 0.91  
    SGD(learning_rate=0.1, momentum=0.9), # 0.977  
]
```

```
optimizersRMSprop = [  
    RMSprop(), # 0.975  
    RMSprop(learning_rate=0.0001), # 0.94  
    RMSprop(learning_rate=0.01), # 0.97  
    RMSprop(learning_rate=0.1), # 0.89  
    RMSprop(rho=0.7), # 0.977  
    RMSprop(rho=0.5), # 0.975  
]
```

```
optimizersAdagrad = [  
    Adagrad(), # 0.87  
    Adagrad(0.1), # 0.973  
    Adagrad(0.001), # 0.877  
]
```

```
optimizersAdadelat = [  
    Adadelat(), # 0.6  
    Adadelat(learning_rate=1.5), # 0.978  
    Adadelat(learning_rate=0.8), # 0.974  
    Adadelat(learning_rate=0.6), # 0.972  
    Adadelat(rho=0.99), # 0.72  
    Adadelat(rho=0.8), # 0.41  
]
```

```
optimizersAdam = [  
    Adam(), # 0.977  
    Adam(learning_rate=0.01), # 0.966  
]
```

```

Adam(learning_rate=0.1), # 0.76
Adam(learning_rate=0.0001), # 0.94
Adam(amsgrad=True), # 0.977
]

optimizersAdamax = [
    Adamax(), # 0.967
    Adamax(learning_rate=0.001), # 0.964
    Adamax(learning_rate=0.01), # 0.977
    Adamax(learning_rate=0.1), # 0.956
    Adamax(learning_rate=0.0002), # 0.93
]

```

Заметно, что каждый из оптимизаторов при правильной подборке параметров выдает хороший результат точности - около 0.97.

Для дальнейшей работы выберем оптимизатор

Adadelata(learning_rate=1.5). Схема слоев ИНС выглядит следующим образом:

```

model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dense(10, activation='softmax'))

```

Здесь слой Flatten нужен для выравнивания входящего двумерного массива и преобразования его в одномерный 28*28 элементов.

Для проверки на реальных изображениях, взятых не из предлагаемого датасета mnist, реализуем функцию по загрузке изображения из файла:

```

def load_image(path):
    img = tf.keras.preprocessing.image.load_img(
        path,
        color_mode='grayscale',
        target_size=(28, 28),
    )
    return 1 - np.reshape(img_to_array(img) / 255, (28, 28))

```

Таким образом можно провести проверку модели на реальных изображениях, нарисованных собственноручно:

9 8 7 6 5 4 3 2 1 0

Видно, что нейронная сеть успешно распознает все изображения:

```

test_acc: 0.9815000295639038
img/0.png is probably 0
img/1.png is probably 1
img/2.png is probably 2
img/3.png is probably 3

```

```
img/4.png is probably 4  
img/5.png is probably 5  
img/6.png is probably 6  
img/7.png is probably 7  
img/8.png is probably 8  
img/9.png is probably 9
```

Для повышения точности предсказания количество эпох было увеличено с 5 до 10.

ВЫВОДЫ

В ходе лабораторной работы были изучены возможности работы с изображениями, их загрузка в нейронную сеть, приведение к grayscale формату и инвертирование при возникновении такой необходимости, получение предсказания о содержимом изображения с использованием нейронных сетей. Разобраны различные оптимизаторы и их параметры, а также найден лучший оптимизатор по результатам обучения на 5 эпохах. Рассмотрена встроенная библиотека mnist, содержащая датасет из нарисованных от руки цифр от 0 до 9, определены способы загрузки данных из нее в ИНС.