

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 объектов на фотографиях"

Студент гр. 8383

Степанов В.Д.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2021

Цель.

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

Выполнения работы.

Была построена модель нейронной сети по методическим материалам.

Модель №1.

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

model.compile(optimizer='adam',loss='categorical_crossentropy',
metrics=['accuracy'])

model.fit(train_images, train_labels, epochs=5, batch_size=128)
```

Результаты запусков показали, что точность модели на тестовых данных в среднем 97%, а значение ошибки 0.08.

Рассмотрим влияние разных оптимизаторов на точность модели.

SGD

| Конфигурация | Точность |
|-------------------------|----------|
| SGD(learning_rate=0.01) | 91% |
| SGD(learning_rate=0.1) | 96% |

RMSprop

| Конфигурация | Точность |
|---------------------------------------|----------|
| RMSprop(learning_rate=0.001, rho=0.9) | 97% |
| RMSprop(learning_rate=0.01, rho=0.9) | 97% |
| RMSprop(learning_rate=0.001, rho=0.5) | 97% |

Adam

| Конфигурация | Точность |
|---------------------------|----------|
| Adam(learning_rate=0.001) | 98% |
| Adam(learning_rate=0.01) | 97% |

Nadam

| Конфигурация | Точность |
|----------------------------|----------|
| Nadam(learning_rate=0.001) | 98% |
| Nadam(learning_rate=0.01) | 97% |

Видно, что хорошую точность показали оптимизаторы Adam и Nadam при параметре learning_rate равном 0.001. Следовательно архитектура, которая показывает наилучшую точность, выглядит так:

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

model.compile(optimizer= optimizers.Adam(learning_rate=0.001),
              loss='categorical_crossentropy', metrics=['accuracy'])

model.fit(train_images, train_labels, epochs=5, batch_size=128)
```

Была написана функция, которая позволяет загружать пользовательские изображения.

```
def loadImage(path):
    img = Image.open(path).convert('L')
    arr = np.array(img) / 255.0
    arr = np.array([arr])

    if arr.shape != (1, 28, 28): raise Exception("Размер изображения неверный")

    return arr
```

Данная функция открывает изображение и преобразует к виду, который будет передан в сеть.

```
predicted_results = model.predict(loadImage(path))
```

Выводы.

В ходе выполнения лабораторной работы была создана нейронная сеть, которая классифицирует черно-белые изображения рукописных цифр по 10 категориям. Была найдена оптимальная конфигурация обучения сети, при которой сеть показывала максимальную точность. Была написана функция, позволяющая пользователю загрузить собственное изображение.