

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

Мирсков А. А.

Преподаватель

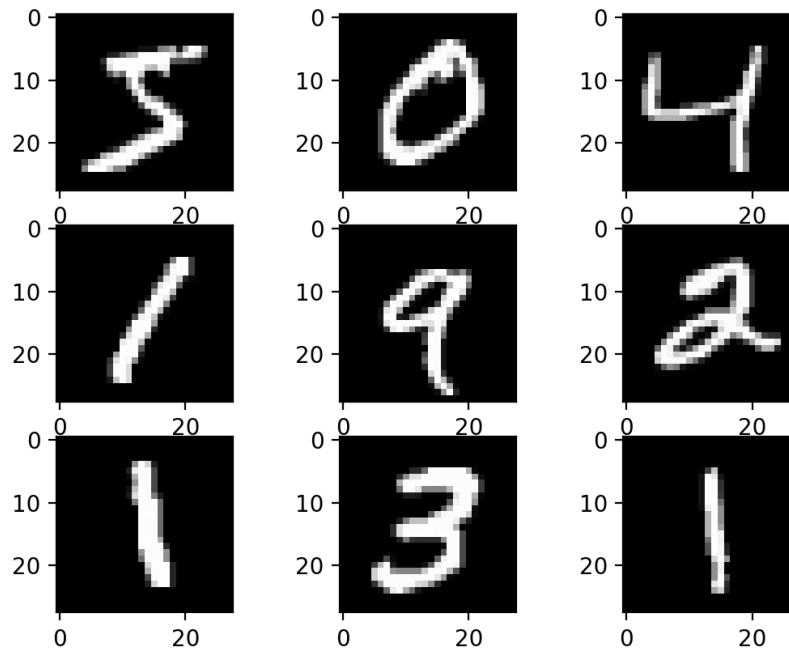
Жангиров Т. Р.

Санкт-Петербург

2021

Цель работы.

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



Выполнение работы.

Была реализована модель на основе методических указаний.

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

optimizer = tf.keras.optimizers.Adam(learning_rate = 0.001)

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

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

Данная модель показывает точность на тестовых данных около 97.5%.

Исследуем влияние различных оптимизаторов и их параметров на точность модели. Ниже представлена точность на тестовом наборе для различных оптимизаторов.

Adam(learning_rate = 0.001): 0.98%

Adam(learning_rate = 0.1): 0.84%

SGD(learning_rate = 0.001): 0.81%

SGD(learning_rate = 0.1): 0.95%

SGD(learning_rate = 0.1, momentum = 0.1): 96%

RMSprop(learning_rate = 0.001): 98%

RMSprop(learning_rate = 0.1): 88%

RMSprop(learning_rate = 0.001, rho = 0.09): 97%

Из полученных значений видно, что наибольшую точность показали Adam и RMSprop с параметром learning_rate = 0.001. Также можно увидеть, что Adam и RMSprop показывают лучшую точность при маленьком значении learning_rate, а SGD при значении 0.1.

Далее была написана функция, которая позволяет загружать пользовательские изображения и, с помощью реализованной нейросети, предсказывает, какая цифра находится на изображении.

```
def predict_user_image():
```

```
    path = './image2.png'
```

```
    img = tf.keras.preprocessing.image.load_img(path, color_mode = 'grayscale',  
target_size = (28,28))
```

```
    input_arr = np.array([tf.keras.preprocessing.image.img_to_array(img)]) / 255.0
```

```
    prediction_result = model.predict(input_arr)
```

```
    return np.argmax(prediction_result, 1)[0]
```

Выводы.

В ходе выполнения лабораторной работы была реализована нейронная сеть, которая классифицирует чёрно-белые изображения рукописных цифр по 10 категориям. Было исследовано влияние различных оптимизаторов, а так же их параметров на процесс обучения. Была написана функция, которая позволяет загружать пользовательские изображения.