

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

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

Студент гр. 8382

Янкин Д.О.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2021

Цель работы.

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

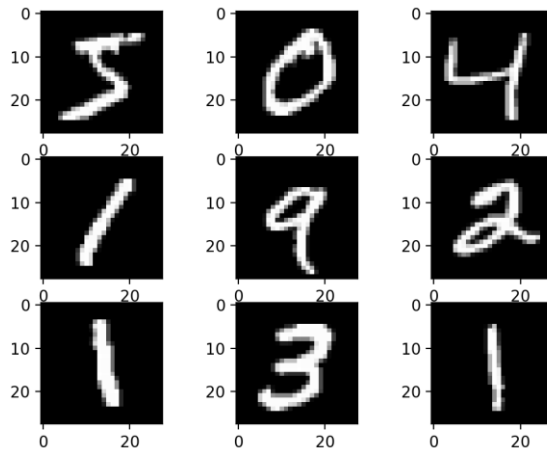


Рисунок 1. Пример входных данных

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

Постановка 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

Выполнение работы.

Датасет изображений цифр MNIST загружается из библиотеки Keras. Выполняется нормализация данных, задаются векторы для десяти категорий. Задаются модель сети и параметры обучения:

```

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)
test_loss, test_acc = model.evaluate(test_images, test_labels)
print('test_acc:', test_acc)

```

Данная сеть дает точность на тестовых данных >97.6%.

Тут у оптимизатора Adam значение learning_rate равно 0.001

Точность на тестовых данных при обучении с разными оптимизаторами и их параметрами при прочих равных:

SGD(learning_rate=0.001) >80.3%

SGD(learning_rate=0.01) >90.8%

SGD(learning_rate=0.1) >95.9%

SGD(learning_rate=1) >97.5%

SGD(learning_rate=1.1) >96.0%

Adagrad(learning_rate=0.0001) >53.5%

Adagrad (learning_rate=0.001) >86.8%

Adagrad (learning_rate=0.01) >93.3%

Adagrad (learning_rate=0.1) >97.3%

Adagrad (learning_rate=0.5) >96.4%

RMSprop(learning_rate=0.0001, rho=0.9) >94.1%

RMSprop(learning_rate=0.001, rho=0.9) >97.7%

RMSprop(learning_rate=0.01, rho=0.9) >96.7%

RMSprop(learning_rate=0.001, rho=0.75) >97.5%

RMSprop(learning_rate=0.001, rho=0.95) >97.6%

RMSprop(learning_rate=0.001, rho=0.99) >97.2%

Adam(learning_rate=0.0001, beta_1=0.9, beta_2=0.999) >93.9%

Adam(learning_rate=0.001, beta_1=0.9, beta_2=0.999) >97.6%

Adam(learning_rate=0.01, beta_1=0.9, beta_2=0.999) >96.7%

Adam(learning_rate=0.001, beta_1=0.8, beta_2=0.999) >97.4%

Adam(learning_rate=0.001, beta_1=0.99, beta_2=0.999) >97.2%

Adam(learning_rate=0.001, beta_1=0.9, beta_2=0.9) >97.6%

Adam(learning_rate=0.001, beta_1=0.9, beta_2=0.7) >97.6%

Изначальный вариант оптимизатора был одним из самых оптимальных в данной ситуации.

Для загрузки изображений в программу используется класс Image из пакета PIL, цветовая модель преобразуется в greyscale. Принимаются только изображения размера 28 на 28. Объект Image преобразуется в np.array, значения цвета приводятся к диапазону [0; 1]. Метод predict возвращает вектор вероятностей принадлежности объекта к каждому из 10 классов. С помощью функции argmax определяется самый вероятный класс. Были нарисованы десять картинок цифр в Paint, сеть правильно опознала восемь из них.

Выводы.

В ходе лабораторной работы была найдена модель сети и параметры обучения, обеспечивающие точность на тестовых данных 97.5%. Были проверены результаты обучения сети при использовании нескольких разных оптимизаторов и их параметров. Была написана функция для загрузки пользовательского изображения и определения изображенной на ней цифры.