

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»



Инженерная школа информационных технологий и робототехники

Отделение информационных технологий

09.04.01 «Информатика и вычислительная техника»

Отчет по практической работе №2
«Распознавание объектов на изображениях»

МАШИННОЕ ОБУЧЕНИЕ

Исполнитель:

студент группы

8BM22

Ямкин Н.Н.

09.09.2023

Руководитель:

Доцент (ОИТ, ИШИТР)

Друки А.А.

Содержание

1. Цель работы	3
2. Задачи	3
3. Ход работы.....	3
4. Заключение	7

1. Цель работы

Получить навыки распознавания объектов на изображениях с использованием искусственных нейронных сетей в библиотеке машинного обучения Keras. В качестве объектов на изображениях будут использоваться рукописные цифры.

2. Задачи

1. Изучение основных элементов библиотеки Keras:

- Реализация топологии НС;

- Обучение НС;

- Тестирование НС;

2. Изучение основных этапов обучения НС:

- Подготовка данных для обучения НС;

- Реализация топологии НС;

- Обучение НС;

- Тестирование НС;

- Расчеты на основе обученной НС.

3. Ход работы

Импортируем необходимые библиотеки:

```
import numpy as np
from keras.datasets import mnist
from matplotlib import pyplot as plt

from tensorflow import keras
from tensorflow.keras import layers

from PIL import Image
from keras.optimizers import Adam
```

Импортируем датасет Fashion MNIST:

```
(x_train, y_train), (x_test, y_test) =  
keras.datasets.fashion_mnist.load_data()
```

Получаем данные датасета в виде массива вещественных чисел и нормализуем их:

```
# Нормализация данных в интервал [0, 1]  
x_train = x_train.astype("float32") / 255  
x_test = x_test.astype("float32") / 255
```

Подготовим желаемые выходные данные. В наборе данных MNIST каждому изображению соответствует число, хранящееся в массиве `mnist.target`. Разрабатываемая нейронная сеть будет иметь 10 выходов, согласно количеству видов одежды в наборе. Таким образом число, соответствующее изображению, нужно привести к вектору, состоящему из 10 элементов. В данном векторе все элементы, кроме одного, соответствующего числу на изображении, должны быть равны нулю. Элемент, соответствующий числу на изображении должен быть равен единице.

```
# преобразование выходных целевых данных нейронной сети к категориальному  
виду  
# y_train - вектор из перекодированных леблов # число 5 будет выглядеть  
вот так [0,0,0,0,0,1,0,0,0,0];  
y_train = keras.utils.to_categorical(y_train, num_classes)  
y_test = keras.utils.to_categorical(y_test, num_classes)
```

Отобразим первые 16 изображений из обучающей выборки:

```
plt.figure(figsize=(6,6))  
for i in range(16):  
    plt.subplot(4,4,i+1)  
    plt.xticks([])  
    plt.yticks([])  
    plt.imshow(x_train[i], cmap=plt.cm.binary)
```



Рисунок 1 – Элементы из обучающего набора

Создадим нейронную сеть:

```
model = keras.Sequential(
    [
        # указываем размерность входа нейросети
        keras.Input(shape=input_shape),
        # перевод в одномерный массив
        layers.Flatten(),
        # ПОЛНОСВЯЗНЫЙ СКРЫТЫЙ СЛОЙ НС; в скобках (количество нейронов,
        # функция активации)
        layers.Dense(100, activation="relu"),
        # ПОЛНОСВЯЗНЫЙ ВЫХОДНОЙ СЛОЙ НС; в скобках (количество нейронов,
        # функция активации)
        layers.Dense(10, activation="softmax"),
    ]
)

model.summary()
```

Добавим два слоя. Первый слой: 784 входа (согласно входному набору), 100 нейронов, функция активации - relu. Второй слой: 10 нейронов (согласно количеству классов), функция активации – softmax.

Определим для нейросети функцию ошибки, оптимизатор и метрику. Подготовим сеть к обучению: зададим количество изображений в батче и число эпох.

Далее скомпилируем нейронную сеть с заданными параметрами и обучим её.

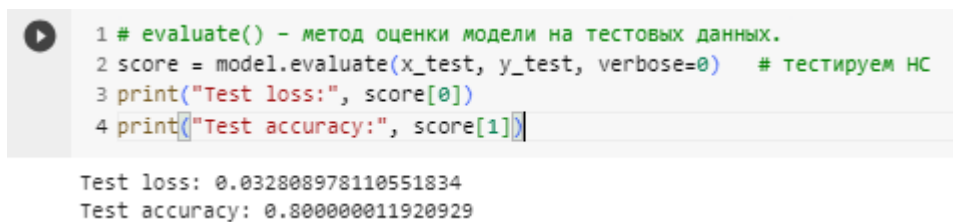
```
batch_size = 10 # количество подаваемых картинок за раз
epochs = 3

model.compile(loss="MSE", optimizer=keras.optimizers.Adam(learning_rate=0.01), metrics=["accuracy"]) # компилируем НС(?); в скобках (Функция ошибки - MSE; алгоритм обучения - Adam; метрика оценки качества - процентная точность)
# обучаем НС
model.fit(x_train, y_train, batch_size=batch_size, epochs=epochs)
```

Протестируем обученную сеть на тестовых данных:

```
# evaluate() - метод оценки модели на тестовых данных.
score = model.evaluate(x_test, y_test, verbose=0) # тестируем НС
print("Test loss:", score[0])
print("Test accuracy:", score[1])
```

В результате получим следующий результат:



```
1 # evaluate() - метод оценки модели на тестовых данных.
2 score = model.evaluate(x_test, y_test, verbose=0) # тестируем НС
3 print("Test loss:", score[0])
4 print("Test accuracy:", score[1])
```

Test loss: 0.032808978110551834
Test accuracy: 0.800000011920929

Рисунок 2 – Результат обучения нейронной сети

Выведем результат работы сети для первого изображения из тестового набора:

```
I = 1 # номер изображения для вывода
# отрисовка изображения
plt.imshow(x_test[I].reshape([28, 28]), cmap='gray')
print("Мнение нейронной сети: ",
      np.argmax(model.predict(x_test[I].reshape([1, 28, 28]))))
print("Верный ответ: ", y_test[I])
```

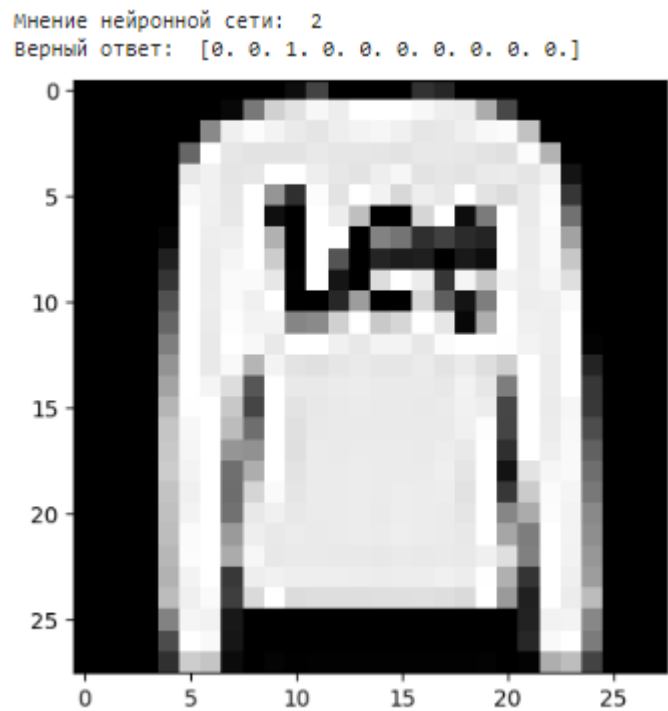


Рисунок 3 – Элемент из тестового набора

4. Заключение

В результате выполнения данной лабораторной получен навык решения задачи классификации элементов одежды Fashion MNIST на изображениях с помощью нейронных сетей. В рамках данной работы были получены навыки работы с Google Colab, iPython Notebooks и библиотекой машинного обучения Keras.