Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Рязанский государственный радиотехнический университет

имени В.Ф.Уткина»

Кафедра «Электронные вычислительные машины»

Отчет по практическому занятию №6

на тему

«Проектирование нейронной сети для распознавания рукописных чисел»

по дисциплине  
**«Машинное обучение»**

Выполнили:

Студенты группы №140

Бригада 7

Сафаров Д. А.

Тимохин Е. С.

Проверила:  
ас. Панина И.С.

**Цель работы**

Научиться составлять модели нейронных сетей для работы с изображениями.

**Задание**

Изменить приведённую нейронную сеть так, чтобы результатом был ответ, есть ли на изображении последняя цифра вашего варианта. Затем вывести результат для случайных изображений.

**Практическая часть**

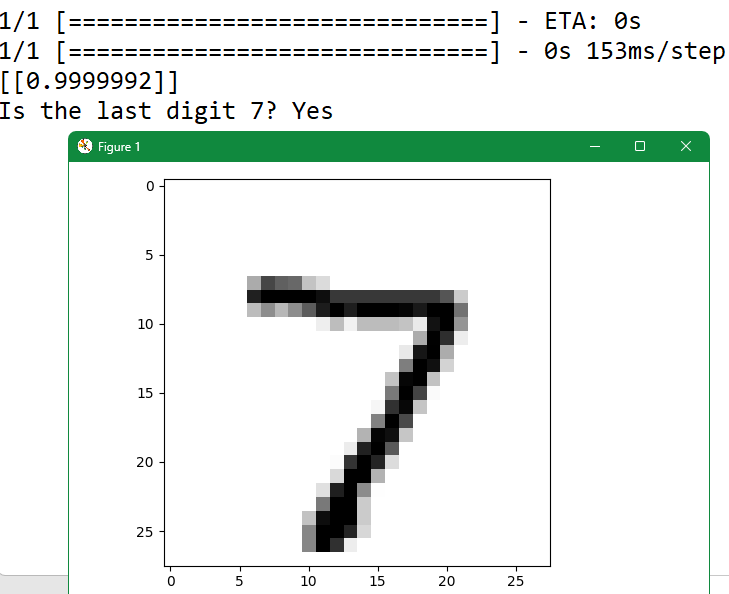
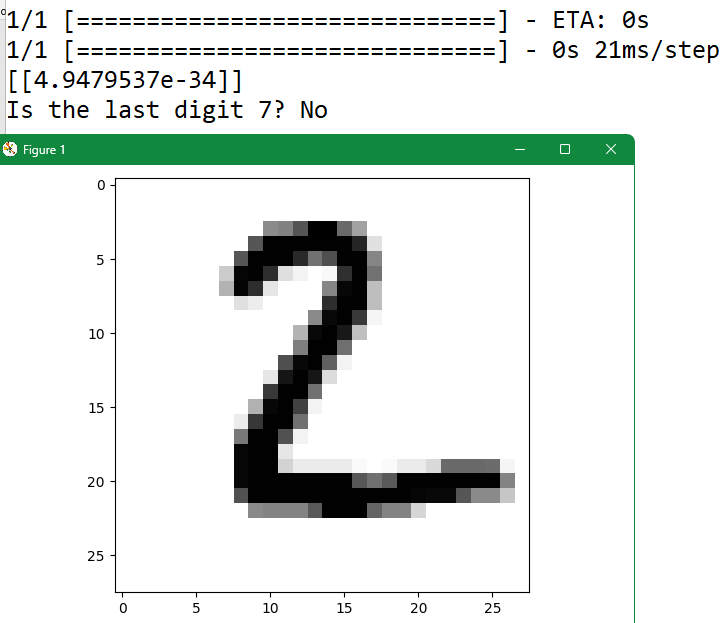
Код программы представлен в приложении А. Результат её работы представлен на рисунке 1.

Рисунок 1 – Результат работы нейросети

Приложение А. Текст программы

import os

os.environ['TF\_CPP\_MIN\_LOG\_LEVEL'] = '2'

import numpy as np

import matplotlib.pyplot as plt

from tensorflow.keras.datasets import mnist # библиотека базы выборок Mnist

from tensorflow import keras

from tensorflow.keras.layers import Dense, Flatten

(x\_train, y\_train), (x\_test, y\_test) = mnist.load\_data()

# стандартизация входных данных

x\_train = x\_train / 255

x\_test = x\_test / 255

yes="Yes"

no="No"

y\_train\_last\_digit\_is\_seven = np.array([yes if str(label)[-1] == '7' else no for label in y\_train])

y\_test\_last\_digit\_is\_seven = np.array([yes if str(label)[-1] == '7' else no for label in y\_test])

model = keras.Sequential([

Flatten(input\_shape=(28, 28, 1)), # Выравнивание входных данных

Dense(128, activation='relu'), # Скрытый слой с 128 нейронами и функцией

активации ReLU

Dense(1, activation='sigmoid')

])

model.compile(optimizer='adam',

loss='binary\_crossentropy', # Функция потерь для бинарной

классификации

metrics=['accuracy'])

model.fit(x\_train, (y\_train\_last\_digit\_is\_seven == yes).astype(int), batch\_size=32,

epochs=10, validation\_split=0.2)

count\_numb = 20

for n in range(count\_numb):

x = np.expand\_dims(x\_test[n], axis=0)

res = model.predict(x)

print(res)

predicted\_label = yes if res >= 0.5 else no

print('Is the last digit 7? ' + str(predicted\_label))

plt.imshow(x\_test[n], cmap=plt.cm.binary)

plt.show()

print()