Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Пермский национальный исследовательский политехнический университет» (ПНИПУ)

Кафедра вычислительной математики и механики

**Лабораторная работа № 5**

**по дисциплине: «Интеллектуальные ИСИТ»**

Выполнил

студент группы ИСТ-19-2б

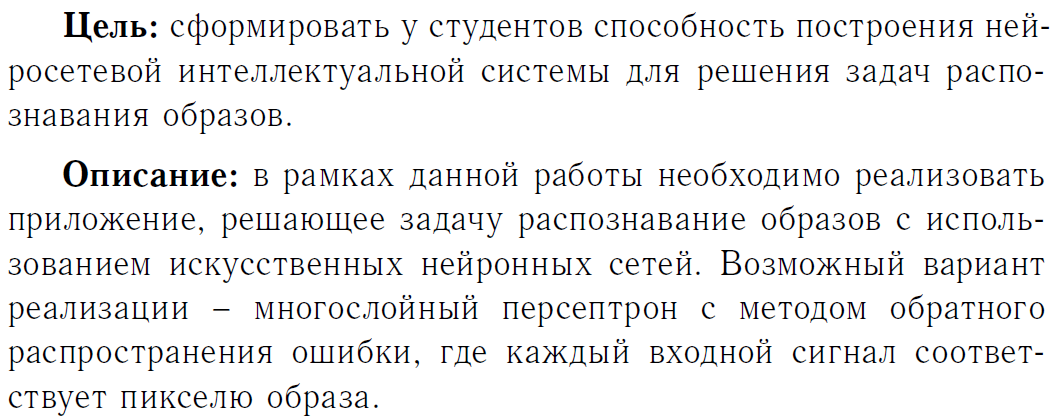
Лебедева А.А.

Проверил

ассистент кафедры ВММБ

Нетбай Г.В.

Пермь, 2022

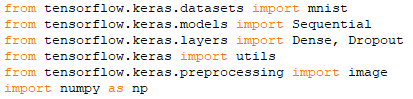


Реализуем модель Персептрона с использованием библиотеки Keras на языке python.

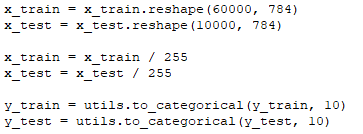
Будем решать задачу распознавания рукописных цифр от 0 до 9. Базу тестирования загружаем из MNIST:



Импортируем библиотеки для обучения Персептрона:



Далее приводим данные к значениям от нуля до единицы:



Создаем модель и добавляем слои:



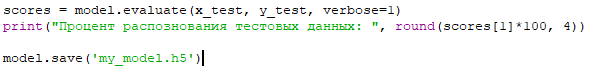
Далее собираем модель:



Запускаем обучение модели:



Вычисляем значение точности модели и сохраняем ее:



Результат программы:



Точность модели составляет 94,83%

Нарисуем в paint цифры от нуля до девяти по три варианта (все варианты лежат в папке image):

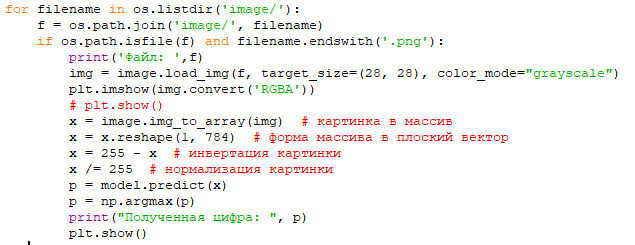


Рис.1. Цифры, нарисованные в paint

Создаем новую программу и загружаем модель:

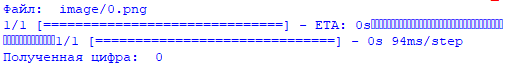


Перебираем все изображения из папки в цикле, преобразуем изображение в массив, прогнозируем, используя полученный вектор, выводим цифру и начальную картинку:



Результат программы:

ЦИФРА НОЛЬ



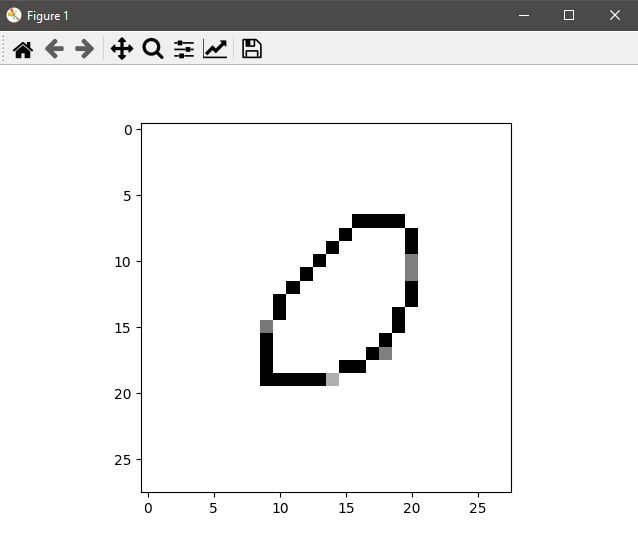
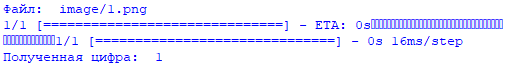


Рис.2. Исходное изображение и преобразованное изображение из вектора

ЦИФРА ОДИН



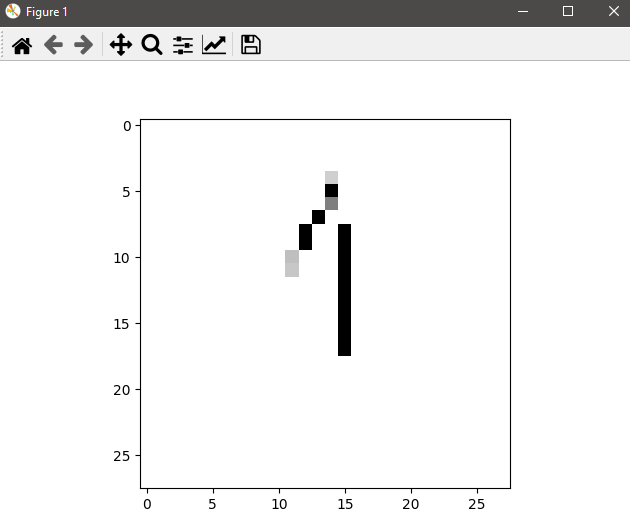
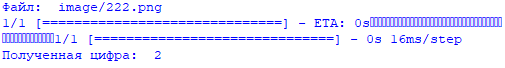


Рис.3. Исходное изображение и преобразованное изображение из вектора

ЦИФРА ДВА



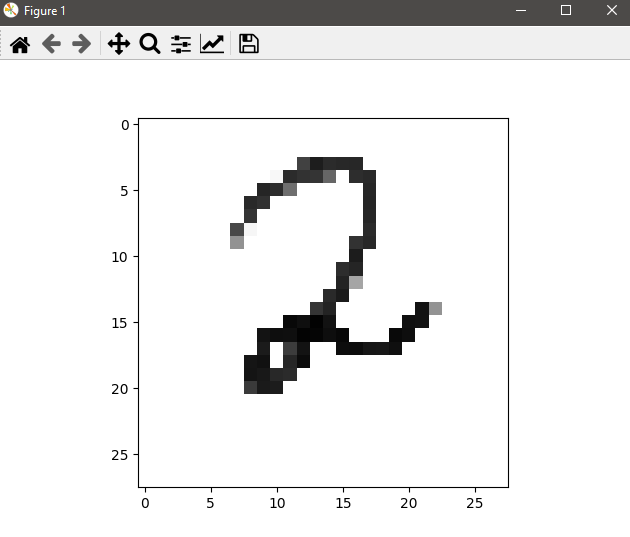
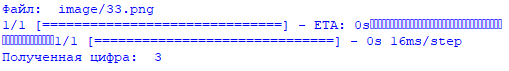
 

Рис.4. Исходное изображение и преобразованное изображение из вектора

ЦИФРА ТРИ



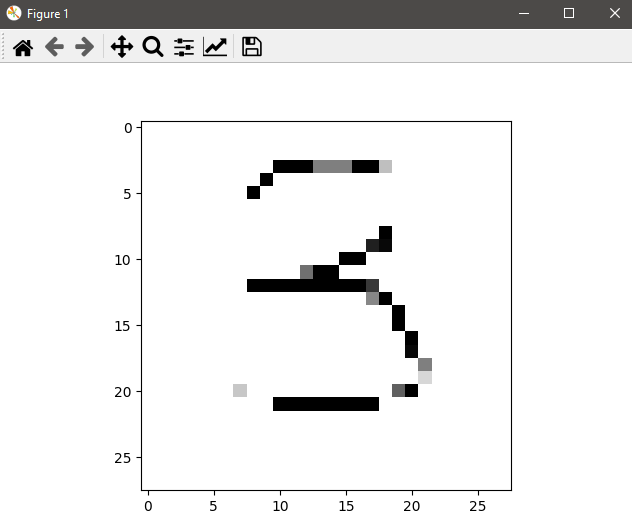
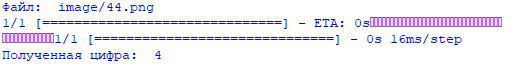
 

Рис.5. Исходное изображение и преобразованное изображение из вектора

ЦИФРА ЧЕТЫРЕ



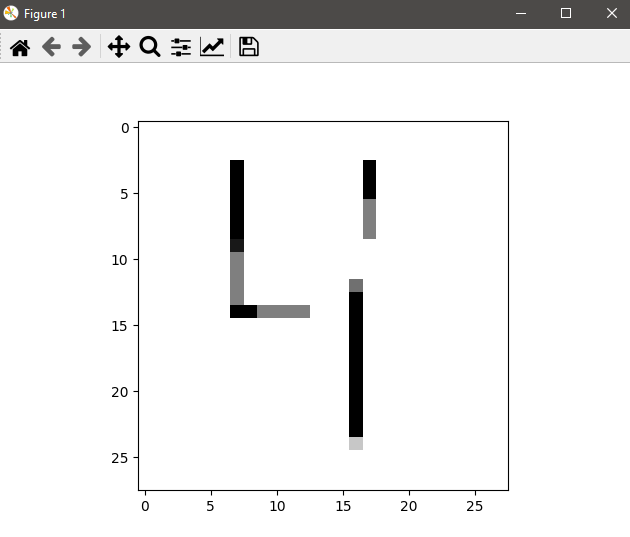
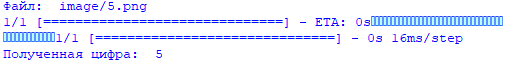
 

Рис.6. Исходное изображение и преобразованное изображение из вектора

ЦИФРА ПЯТЬ



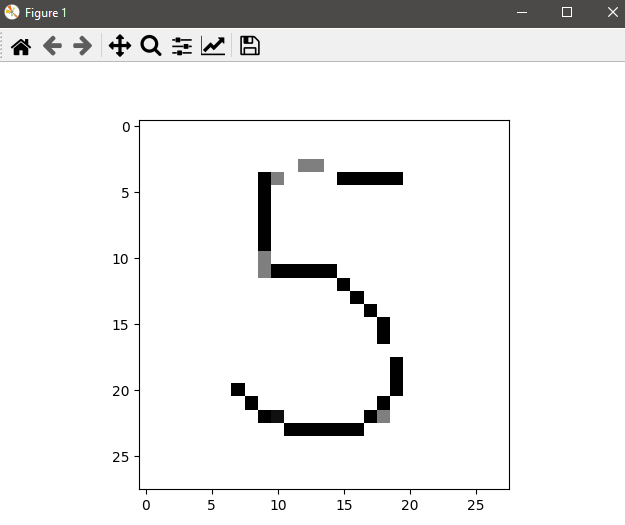
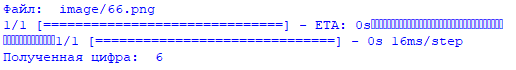
 

Рис.7. Исходное изображение и преобразованное изображение из вектора

ЦИФРА ШЕСТЬ



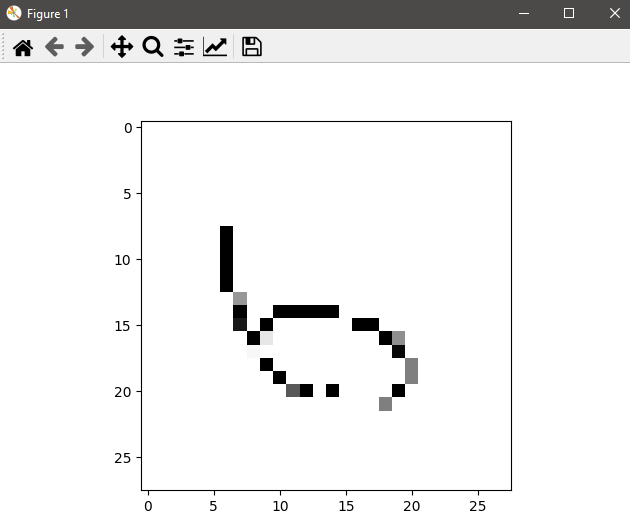
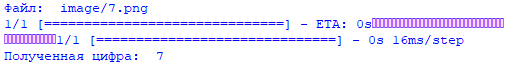
 

Рис.8. Исходное изображение и преобразованное изображение из вектора

ЦИФРА СЕМЬ



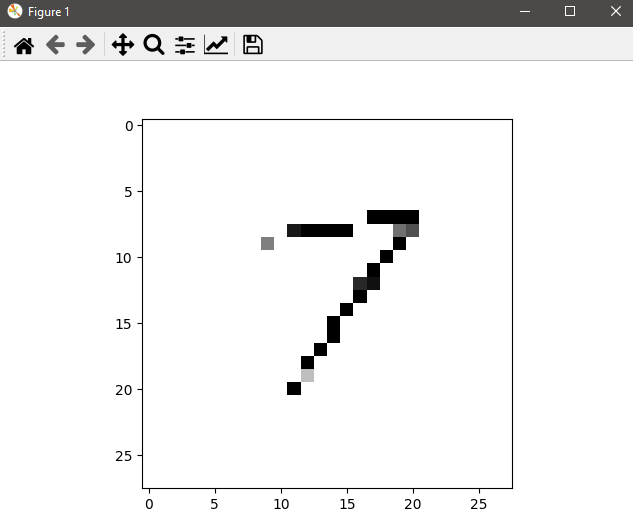
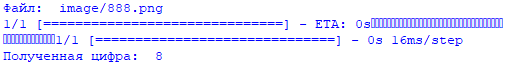


Рис.9. Исходное изображение и преобразованное изображение из вектора

ЦИФРА ВОСЕМЬ



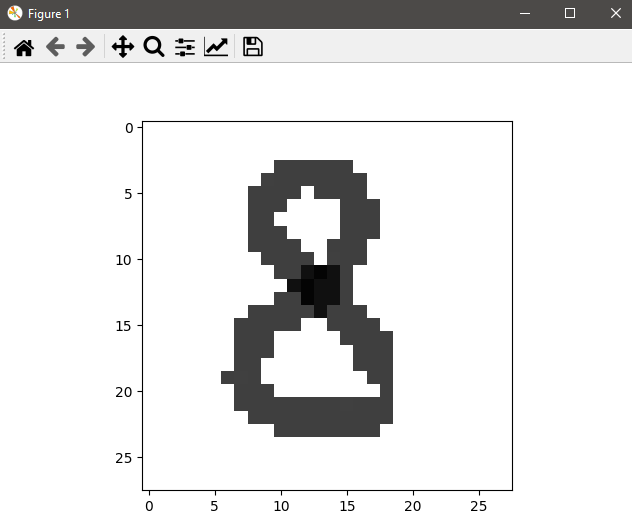
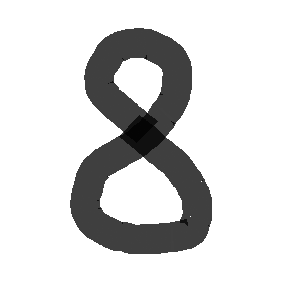
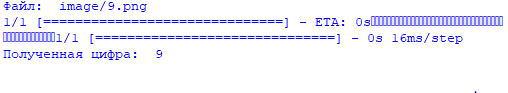


Рис.10. Исходное изображение и преобразованное изображение из вектора

ЦИФРА ДЕВЯТЬ



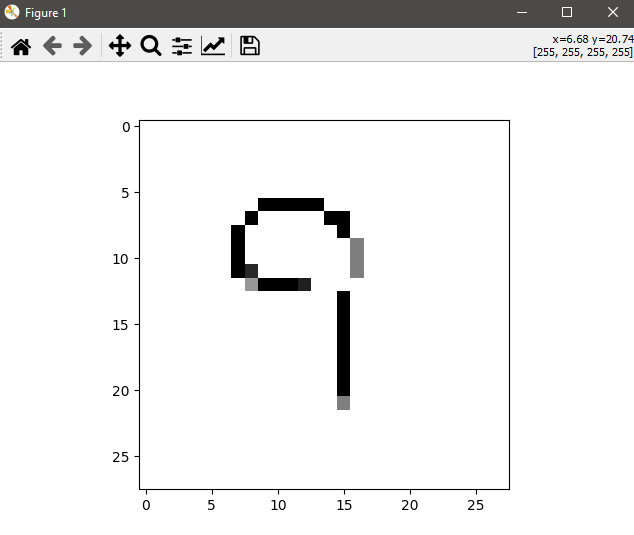
 

Рис.11. Исходное изображение и преобразованное изображение из вектора

Листинг программы main:

from tensorflow.keras.datasets import mnist

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Dropout

from tensorflow.keras import utils

from tensorflow.keras.preprocessing import image

import numpy as np

import matplotlib.pyplot as plt

from PIL import Image

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

x\_train = x\_train.reshape(60000, 784)

x\_test = x\_test.reshape(10000, 784)

x\_train = x\_train / 255

x\_test = x\_test / 255

y\_train = utils.to\_categorical(y\_train, 10)

y\_test = utils.to\_categorical(y\_test, 10)

model = Sequential()

model.add(Dense(800, input\_dim = 784, activation = "relu"))

model.add(Dense(10, activation = "softmax"))

model.compile(loss="categorical\_crossentropy", optimizer = "SGD", metrics = ["accuracy"])

#print(model.summary())

history = model.fit(x\_train, y\_train, batch\_size=200, epochs=50,validation\_split=0.2,verbose=1)

scores = model.evaluate(x\_test, y\_test, verbose=1)

print("Процент распознования тестовых данных: ", round(scores[1]\*100, 4))

model.save('my\_model.h5')

Листинг программы lab5\_2:

from tensorflow.keras.datasets import mnist

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Dropout

from tensorflow.keras import utils

from tensorflow.keras.preprocessing import image

from tensorflow.keras import models

import numpy as np

import matplotlib.pyplot as plt

from PIL import Image

import os

model = models.load\_model('my\_model.h5')

for filename in os.listdir('image/'):

f = os.path.join('image/', filename)

if os.path.isfile(f) and filename.endswith('.png'):

print('Файл: ',f)

img = image.load\_img(f, target\_size=(28, 28), color\_mode="grayscale")

plt.imshow(img.convert('RGBA'))

# plt.show()

x = image.img\_to\_array(img) # картинка в массив

x = x.reshape(1, 784) # форма массива в плоский вектор

x = 255 - x # инвертация картинки

x /= 255 # нормализация картинки

p = model.predict(x)

p = np.argmax(p)

print("Полученная цифра: ", p)

plt.show()