Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ИНДИВИДУАЛЬНАЯ РАБОТА №3**

**Дисциплина: Нечеткий анализ и моделирование**

Работу выполнила: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. А. Ромашкина

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О. В. Руденко

**Тема:** Распознавание рукописных цифр из базы MNIST с помощью нейронной сети.

**Цель работы:** научиться строить нейронные сети для распознавания рукописных цифр.

**Задание:** построить нейронную сеть средствами библиотеки Keras языка Python (или соответствующей библиотеки на другом языке), позволяющую распознавать цифры. Для обучения и тестирования использовать базу MNIST. Возможно использовать любую функции активации, любую структуру сети и любой алгоритм обучения.

**Ход работы**

1. На рисунках 1 и 2 показана структура разработанной нейронной сети.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1 – Результат model.summary()

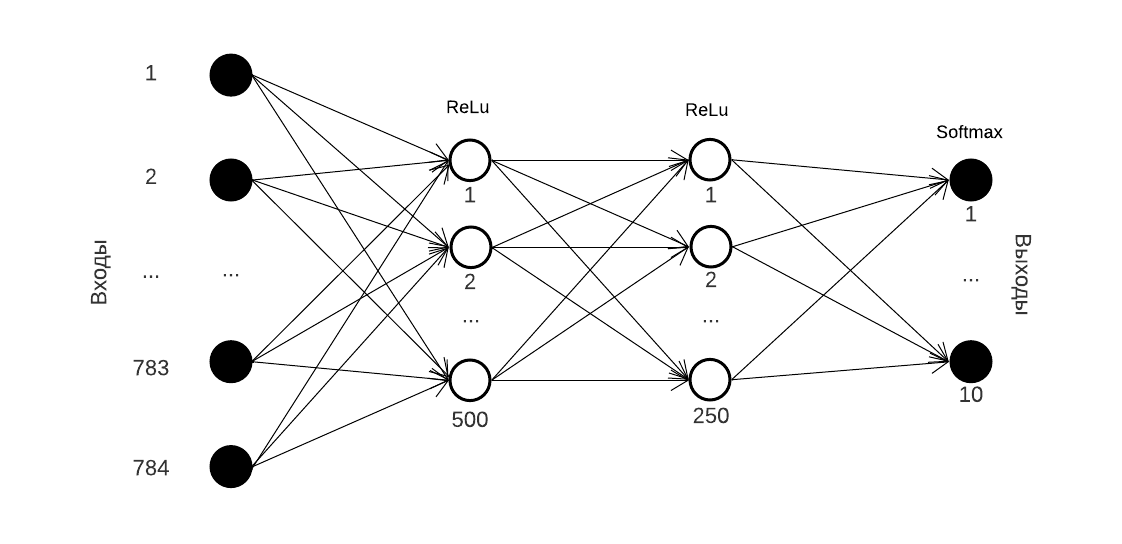


Рисунок 2 – Структура нейронной сети

1. На первом и втором скрытых слоях используется функция активации ReLu, которая имеет следующий вид:



На выходном слое используется функция Softmax:



1. Для обучения нейронной сети использовались метод обратного распространения ошибки и данные из базы MNIST. База данных образцов рукописного написания цифр MNIST содержит 60 000 образцов наборов данных для обучения и тестовый набор из 10 000 образцов. Цифры нормализованы по размеру и расположены в центре изображения фиксированного размера.

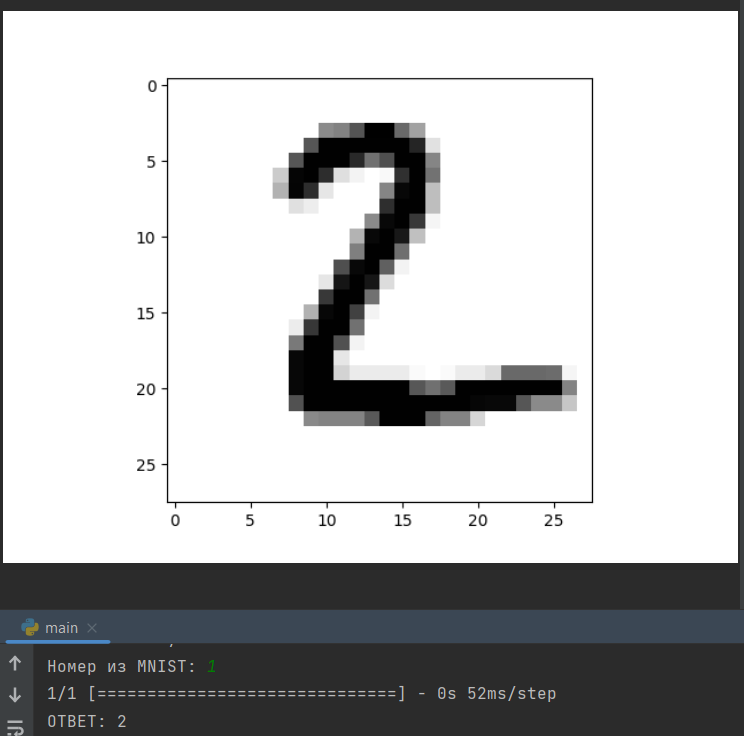


Рисунок 3 – Результат работы программы

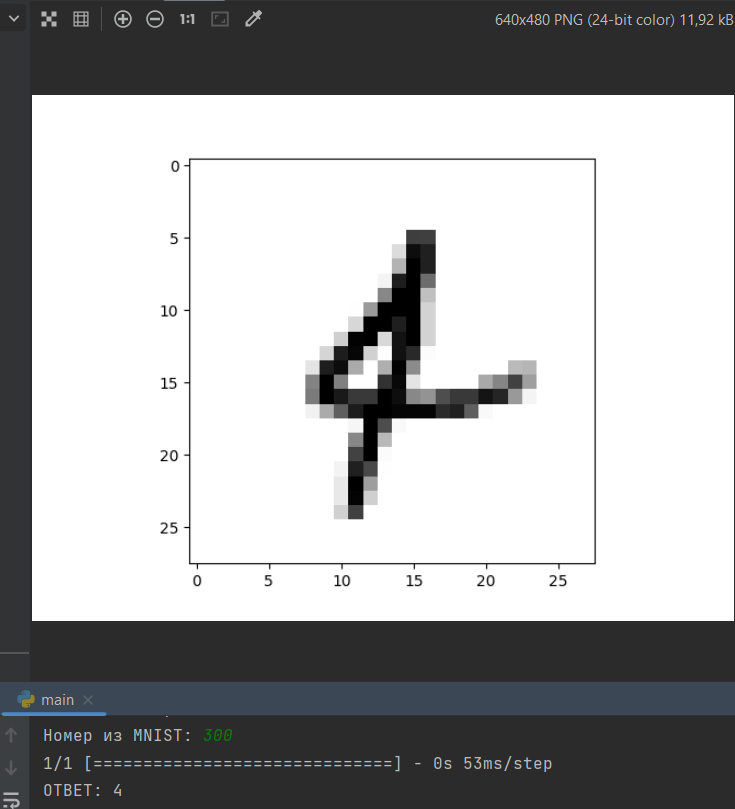


Рисунок 4 – Результат работы программы

Ниже представлен листинг разработанной программы.

import os

from tensorflow import keras

import numpy as np

import matplotlib.pyplot as plt

from keras import Sequential

from keras.layers import Dense, Flatten

from keras.utils import np\_utils

num\_classes = 10

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

x\_train = x\_train / 255.0

x\_test = x\_test / 255.0

x\_train = np.expand\_dims(x\_train, -1)

x\_test = np.expand\_dims(x\_test, -1)

y\_train\_cat = keras.utils.to\_categorical(y\_train, num\_classes)

y\_test\_cat = keras.utils.to\_categorical(y\_test, num\_classes)

model = keras.Sequential()

model.add(Flatten(input\_shape=(28, 28, 1)))

model.add(Dense(500, activation='relu', use\_bias=True,name='Dense\_first'))

model.add(Dense(250, activation='relu', use\_bias=True,name='Dense\_second'))

model.add(Dense(num\_classes, activation='softmax', use\_bias=True,name='Dense\_result'))

model.compile(optimizer='adam',

loss='categorical\_crossentropy',

metrics=['accuracy'])

model.fit(x\_train, y\_train\_cat, batch\_size=32, epochs=7, validation\_split=0.2)

score = model.evaluate(x\_test, y\_test\_cat, verbose=0)

print(model.summary())

print("Test loss:", score[0])

print("Test accuracy:", score[1])

n = int(input("Номер из MNIST: "))

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

res = model.predict(x)

print("ОТВЕТ: " + str(np.argmax(res)))

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

plt.show()

**Вывод:** при выполнении данной лабораторной работы была построена и обучена нейронная сеть для распознавания рукописных цифр.