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**

**Отчёт**

**по ЛАБОРАТОРНОЙ РАБОТЕ 12**

**Дисциплина: АЛГОРИТМ ЦИФРОВОЙ ОБРАБОТКИ МУЛЬТИМЕДИА**

**Тема: «РАСПОЗНАВАНИЕ РУКОПИСНЫХ ЦИФР»**

Работу выполнила \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. Ш. Пивоварова

Направление подготовки 02.03.02 Фундаментальная информатика и

информационные технологии

Направленность (профиль) Математическое и программное обеспечение

компьютерных технологий

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.С. Жук

**Цель работы**

Построить нейронную сеть средствами библиотеки Keras языка Python (или соответствующей библиотеки на другом языке), позволяющую распознавать цифры. Для обучения и тестирования использовать базу MNIST.

**Ход работы**

Для решения задачи будем использовать Python 3.10 и библиотеку Keras. На вход будут поступать изображения цифр из базы MNIST. На рисунке 1 показан пример такого изображения.

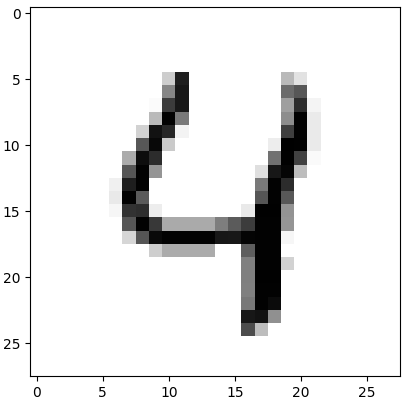


Рисунок 1 – Пример изображения

Каждое изображение имеет размер 28 х 28 пикселей и представлено в градациях серого. Всего в базе 60 000 изображений в обучающей выборке и 10 000 в тестовой.

Для распознавания цифр было принято решение использовать простую полносвязную сеть, состоящую из:

* 28\*28=784 входных нейронов;
* 128 нейронов скрытого слоя;
* 10 нейронов выходного слоя.

На рисунке 2 изображена структура выбранной нейронной сети.

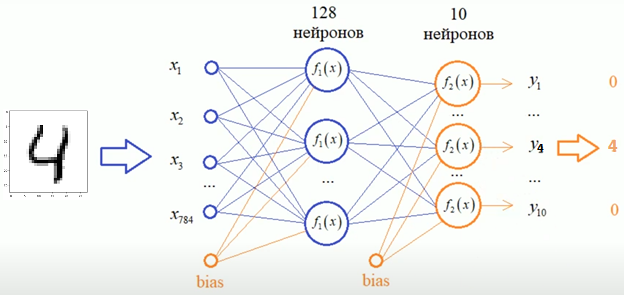


Рисунок 2 – Структура нейронной сети

В качестве функции активации на скрытом слое используется функция relu, которая возвращает значение х, если х положительно, и 0 в противном случае.



В качестве функции активации на выходном слое используется функция softmax. Функция преобразует вектор z размерности K в вектор той же размерности, где каждая координата полученного вектора представлена вещественным числом в интервале [0,1] и сумма координат равна 1.



Функция softmax применяется в машинном обучении для задач классификации, когда количество возможных классов больше двух (для двух классов используется логистическая функция). Координаты полученного вектора  при этом трактуются как вероятности того, что объект принадлежит к классу i.

В качестве обучения был выбран способ «обучение с учителем», а метод – обратное распространение. На рисунке 3 показан примеры работы программы.

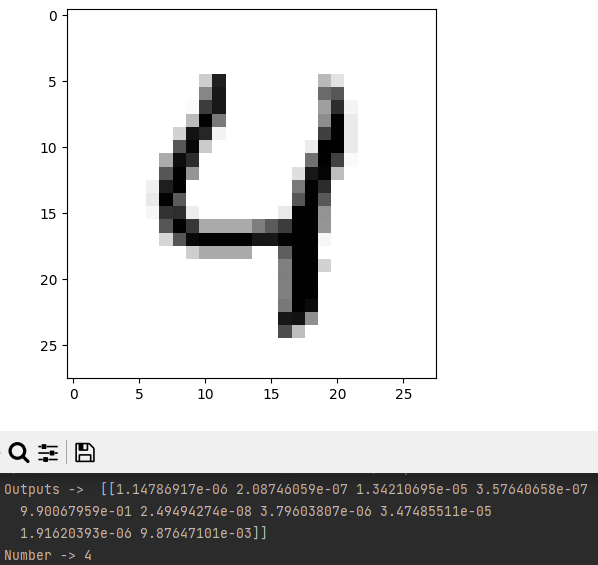


Рисунок 3 – Пример работы программы

**Вывод**

Были изучены базовые принципы работы нейронных сетей, а также их основные составляющие. Построена простая нейронная сеть для распознавания рукописных цифр. Изучена библиотека Keras.

**Листинг программы**

from tensorflow import keras

from keras.datasets import mnist

from keras.layers import Dense, Flatten

import numpy as np

import matplotlib.pyplot as plt

import os

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

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

x\_train = x\_train / 255

x\_test = x\_test / 255

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

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

model = keras.Sequential([

Flatten(input\_shape=(28, 28, 1)),

Dense(128, activation='relu'),

Dense(10, activation='softmax')

]) # tanh

model.compile(optimizer='adam',

loss='categorical\_crossentropy',

metrics=['accuracy'])

model.fit(x\_train, y\_train\_cat, epochs=5, validation\_split=0.2)

model.evaluate(x\_test, y\_test\_cat)

n = int(input("Picture number from MNIST or -1 to exit -> "))

if n != -1:

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

res = model.predict(x)

print("Outputs -> ", res)

print("Number -> " + str(np.argmax(res)))

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

plt.show()