

Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет Систем Управления и Робототехники

Группа R41333с

# ПРОЕКТИРОВАНИЕ НЕЙРОННЫХ СЕТЕЙ С ПОМОЩЬЮ PYTHON

Лабораторная работа №2  
Вариант № 9

Выполнил: студент гр. R41333с  
Крылатых Д.Н.

Преподаватель: Нуждин К.А.

Санкт-Петербург  
2021

## Задание

Одной из задач киберфизических систем является распознавание изображений с помощью машинного зрения. В этой лаборатории мы рассмотрим решение такого рода задач.

Мы будем использовать два файла данных с массивом рукописных чисел: **mnist\_train.csv** и **mnist\_test.csv**. Первый файл необходимо использовать для обучения нейронной сети, второй – для проверки функционирования. Каждый файл содержит строки (60000 в первом файле и 10000 во втором файле), в каждой из которых хранится массив пикселей 28x28 с цифровым изображением и номером, соответствующим изображению. Пример данных показан на следующем рисунке:

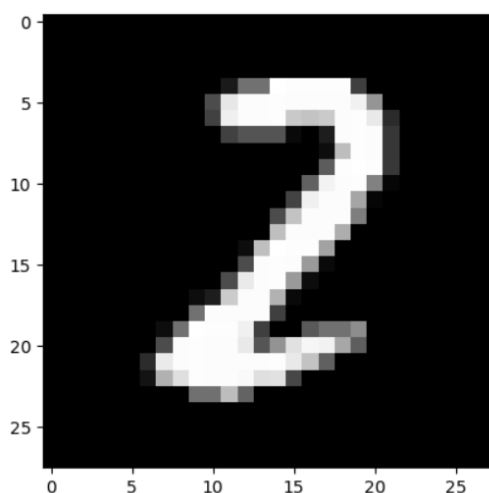


Рисунок 1 – Пример изображения с номером

Основная задача – спроектировать нейронную сеть на Python, способную распознавать изображения чисел. Для этого необходимо:

1. Импортировать библиотеки в Python.
2. Написать функцию для установки основных параметров сети.
3. Создать функцию, которая задает начальные значения весов нейронной сети.
4. Создать функцию, которая вычисляет выход нейронной сети.
5. Создать функцию для обучения нейронной сети.
6. Написать функцию для обучения сети на реальных данных.
7. Написать функцию проверки сети.
8. Написать функцию, которая отображает изображения чисел из набора данных.
9. Обучить сеть и рассчитать ее эффективность.

## Ход работы

### 1. Импорт библиотек

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.special import expit as fAct
```

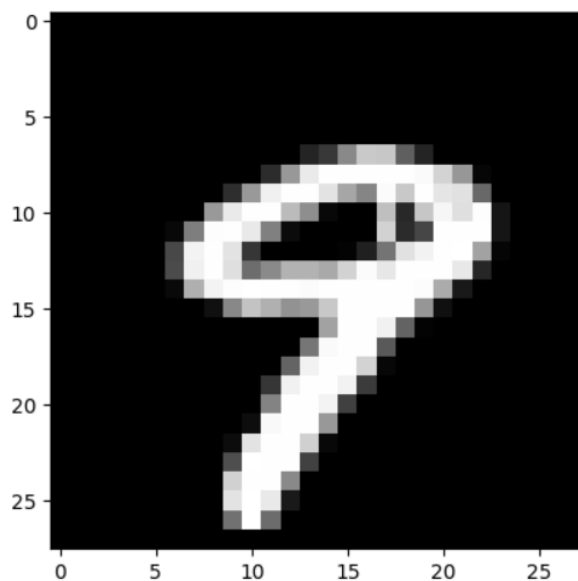
### 2. Напишем функцию для установки основных параметров сети

```
def initNetPar():
    inputNodes = 784
    print('Введите число скрытых нейронов: ')
    hiddenNodes = int(input())
    outNodes = 10
    print('Введите коэф. обучения: ')
    learnSpeed = float(input())
    return inputNodes, hiddenNodes, outNodes, learnSpeed
```

### 3. Напишем функцию, задающую начальные значения весов нейронной сети

```
def createNetW(inputNodes, hiddenNodes, outNodes):
    wInHid = np.random.uniform(-0.5, 0.5, (hiddenNodes, inputNodes))
    wHidOut = np.random.uniform(-0.5, 0.5, (outNodes, hiddenNodes))
    return wInHid, wHidOut
```

Установим все веса сети равными 0,5. Найдём выходной результат (цифру) нейронной сети для 9-й строки тестового набора данных.





#### 4. Напишем функцию, которая вычисляет выходные данные нейронной сети

Выход нейронной сети вычисляется по формуле:

$$O = F(W_{OL} \cdot F(W_{HL} \cdot I))$$

```
def netOut(wInHid, wHidOut, inputSignal, returnHid):
    inputs = np.array(inputSignal, ndmin=2).T
    hidIn = np.dot(wInHid, inputs)
    hidOut = fAct(hidIn)
    finalIn = np.dot(wHidOut, hidOut)
    finalOut = fAct(finalIn)

    if returnHid==0:
        return finalOut
    else:
        return finalOut, hidOut
```

#### 5. Создадим функцию для обучения нейронной сети.

Нейронная сеть обучается по следующим формулам:

$$\Delta w_{OL.jk} = -2 \cdot \alpha \cdot e_{OL.k} \cdot o_{OL.k} \cdot (1 - o_{OL.k}) \cdot o_{HL.j}$$

$$\Delta w_{HL.jk} = -2 \cdot \alpha \cdot e_{HL.k} \cdot o_{HL.k} \cdot (1 - o_{HL.k}) \cdot i_j$$

```
def netTrain(targetList, inputSignal, wInHid, wHidOut, learnSpeed):
    targets = np.array(targetList, ndmin=2).T
    inputs = np.array(inputSignal, ndmin=2).T
    finalOut, hidOut = netOut(wInHid, wHidOut, inputSignal, 1)

    outErrors = targets - finalOut
    hidErrors = np.dot(wHidOut.T, outErrors)

    wHidOut += learnSpeed*np.dot((outErrors*finalOut*(1-finalOut)), hidOut.T)
    wInHid += learnSpeed*np.dot((hidErrors*hidOut*(1-hidOut)), inputs.T)

    return wInHid, wHidOut
```

## 6. Напишем функцию для обучения сети на реальных данных

```
def trainSet(wInHid, wHidOut, learnSpeed):
    dataFile = open("mnist_train.csv", 'r')
    trainingList = dataFile.readlines()
    dataFile.close()

    for record in trainingList:
        allValues = record.split(',')
        inputs = (np.asfarray(allValues[1:])/255.0*0.999)+0.001
        targets = np.zeros(10)+0.001
        targets[int(allValues[0])]=1.0
        wInHid, wHidOut = netTrain(targets, inputs, wInHid, wHidOut, learnSpeed)

    return wInHid, wHidOut
```

## 7. Напишем функцию проверки сети

```
def testSet(wInHid, wHidOut):
    dataFile = open("mnist_test.csv", 'r')
    testList = dataFile.readlines()
    dataFile.close()

    test=[]
    for record in testList:
        allValues = record.split(',')
        inputs = (np.asfarray(allValues[1:])/255.0*0.999)+0.001
        outSession = netOut(wInHid, wHidOut, inputs, 0)

        if int(allValues[0])==np.argmax(outSession):
            test.append(1)
        else:
            test.append(0)
    test = np.asarray(test)
    print('Эффективность % = ', (test.sum()/test.size)*100)
```

## 8. Напишем функцию, которая отображает изображения чисел из набора данных

```
def plotImage(pixels: np.array):
    plt.imshow(pixels.reshape((28,28)), cmap='gray')
    plt.show()
```

## 9. Обучим сеть и рассчитаем ее эффективность

```
inputNodes, hidNodes, outNodes, learnSpeed = initNetPar()
wInHid, wHidOut = createNetW(inputNodes, hidNodes, outNodes)
for i in range(5):
    print('Тест № ', i+1)
    wInHid, wHidOut = trainSet(wInHid, wHidOut, learnSpeed)
    testSet(wInHid, wHidOut)

dataFile = open("mnist_test.csv", 'r')
testList = dataFile.readlines()
dataFile.close()
allValues = testList[int(np.random.uniform(0,9999))].split(',')
inputs = (np.asfarray(allValues[1:])/255.0*0.999)+0.001
outSession = netOut(wInHid, wHidOut, inputs, 0)
print(np.argmax(outSession))
plotImage(np.asfarray(allValues[1:]))
```

Введите число скрытых нейронов:

>? 100

Введите коэф. обучения:

>? 0.5

Тест № 1

Эффективность % = 92.12

Тест № 2

Эффективность % = 93.17

Тест № 3

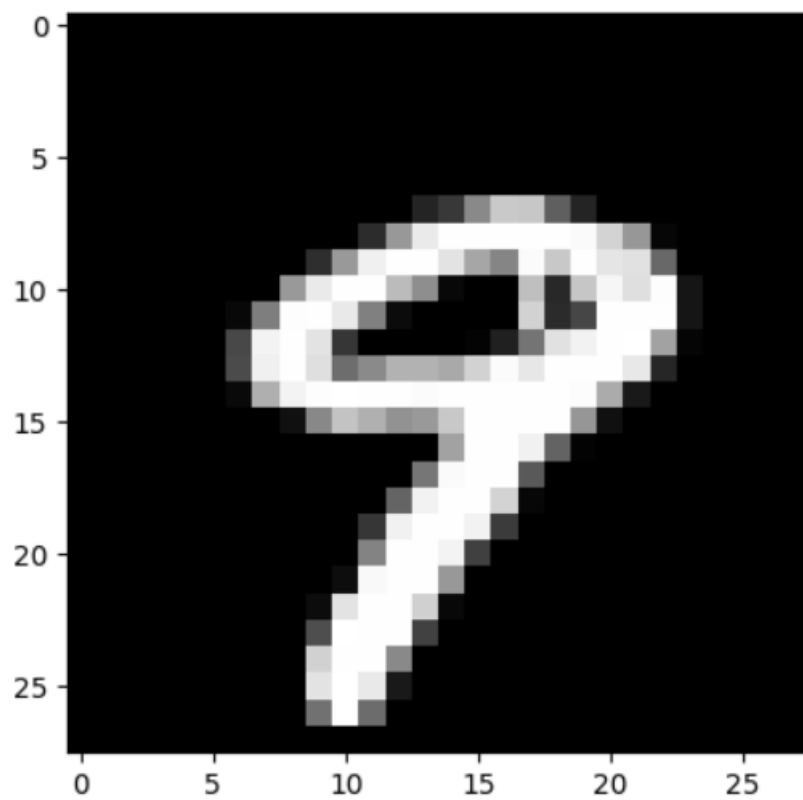
Эффективность % = 92.97999999999999

Тест № 4

Эффективность % = 92.33

Тест № 5

Эффективность % = 93.94



```
.....  
9  
>>>
```