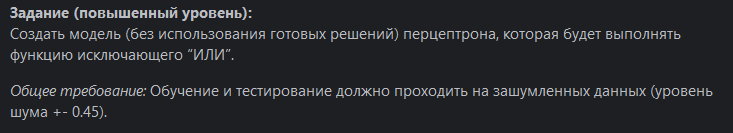
# Группа 5 Панфилов Валерий Александрович Лабораторная работа №1

«Создание нейронной сети, выполняющей бинарную классификацию»

Задание: 

Код:

import copy  
import random  
import numpy as np  
  
  
class Perceptron:  
 def \_\_init\_\_(self,epochs):  
 self.weights = np.ones((2,1))  
 self.epochs = epochs  
  
 def creat\_dataset(self):  
 dataset = list()  
  
 dataset.append(([0, 0], 0))  
 dataset.append(([0, 1], 1))  
 dataset.append(([1, 0], 1))  
 dataset.append(([1, 1], 0))  
  
 noise\_dataset = list()  
 for i in range(10000):  
 noise\_dataset.append(copy.deepcopy(random.choice(dataset)))  
 for j in range(2):  
 noise\_dataset[i][0][j] += (random.randint(-45, 45) \* 0.01)  
  
 return noise\_dataset  
  
 def bin\_classification(self,input\_ind):  
 ind = []  
 if input\_ind[0]-0.5 > 0:  
 ind.append(1)  
 else:  
 ind.append(0)  
 if input\_ind[1]-0.5 > 0:  
 ind.append(1)  
 else:  
 ind.append(0)  
 return ind  
 def summator(self,input\_ind):  
 summator = 0  
 for i in range(2):  
 summator += input\_ind[i]\*self.weights[i]  
 if summator >0.5 and summator < 1.5:  
 return True  
 else:  
 return False  
  
 def learn(self):  
 dataset = self.creat\_dataset()  
 for i in range(len(dataset)):  
 if bool(dataset[i][1]) != self.summator(self.bin\_classification(dataset[i][0])):  
 self.weights[0][i] = self.weights[0][i]+0.1  
  
  
 def train(self):  
 for i in range(self.epochs):  
 self.learn()  
  
 def test(self,input\_dataset,test\_res):  
 res = 0  
 for i in range(len(input\_dataset)):  
 print(f"Входные данные {input\_dataset[i]} Предсказание: {self.summator(self.bin\_classification(input\_dataset[i]))}")  
 if self.summator(self.bin\_classification(input\_dataset[i])) == test\_res[i]:  
 res+=1  
 print(self.weights)  
 print("Accurancy: ",res/len(test\_res))  
  
p = Perceptron(10)  
p.train()  
  
test\_data = list()  
test\_res = list()  
dataset = list()  
dataset.append(([0, 0],0))  
dataset.append(([0, 1],1))  
dataset.append(([1, 0],1))  
dataset.append(([1, 1],0))  
noise\_dataset = list()  
for i in range(5000):  
 noise\_dataset.append(copy.deepcopy(random.choice(dataset)))  
 for j in range(2):  
 noise\_dataset[i][0][j] += (random.randint(-45, 45) \* 0.01)  
for ind in range(len(noise\_dataset)):  
 test\_data.append(noise\_dataset[ind][0])  
 test\_res.append(noise\_dataset[ind][1])  
p.test(test\_data,test\_res)

Результат выполнения:

