Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе № 1

по курсу «Основы глубокого обучения»

Выполнил:  
студент группы 22ВВИм1:

Милованов А.С.

Приняли:

Митрохин М.А  
Панков А.А.

Пенза 2023

# Порядок выполнения работы:

# Задание:

# Реализовать правила остановки алгоритма обучения перцептрона при его зацикливании или сходимости.

# Теорема о сходимости персептрона. Если существует вектор параметров α, при котором персептрон правильно решает все примеры обучающей выборки, то при обучении персептрона по правилу Хебба решение будет найдено за конечное число шагов.

# Теорема о «зацикливании» персептрона. Если не существует вектора параметров α, при котором персептрон правильно решает все примеры обучающей выборки, то при обучении персептрона по правилу Хебба через конечное число шагов вектор весов начнет повторяться.

# Листинг программы:

import pandas as pd  
import matplotlib.pyplot as plt  
import numpy as np  
  
  
# загружаем и подготавляваем данные  
df = pd.read\_csv('data.csv')  
  
df = df.iloc[np.random.permutation(len(df))]  
y = df.iloc[0:100, 4].values  
y = np.where(y == "Iris-setosa", 1, -1)  
  
#берем только первые два признака  
X = df.iloc[0:100, [0, 2]].values  
  
  
inputSize = X.shape[1] # количество входных сигналов равно количеству признаков задачи   
hiddenSizes = 10 # задаем число нейронов скрытого (А) слоя   
outputSize = 1 if len(y.shape) else y.shape[1] # количество выходных сигналов равно количеству классов задачи  
  
print('inputSize:',inputSize)  
print('hiddenSizes:',hiddenSizes)  
print('outputSize:',outputSize)  
  
  
# создаем матрицу весов скрытого слоя  
Win = np.zeros((1+inputSize,hiddenSizes))   
# пороги w0 задаем случайными числами  
Win[0,:] = (np.random.randint(0, 3, size = (hiddenSizes)))   
# остальные веса задаем случайно -1, 0 или 1   
Win[1:,:] = (np.random.randint(-1, 2, size = (inputSize,hiddenSizes)))   
  
print('Win:',Win)  
  
# случайно инициализируем веса выходного слоя  
Wout = np.random.randint(0, 2, size = (1+hiddenSizes,outputSize)).astype(np.float64)  
print('Wout:',Wout)  
   
# функция прямого прохода (предсказания)   
def predict(Xp):  
 # выходы первого слоя = входные сигналы \* веса первого слоя  
 hidden\_predict = np.where((np.dot(Xp, Win[1:,:]) + Win[0,:]) >= 0.0, 1, -1).astype(np.float64)  
 # выходы второго слоя = выходы первого слоя \* веса второго слоя  
 out = np.where((np.dot(hidden\_predict, Wout[1:,:]) + Wout[0,:]) >= 0.0, 1, -1).astype(np.float64)  
 return out, hidden\_predict  
  
  
  
n\_iter=0  
eta = 0.01  
  
  
# обучение  
# у перцептрона Розенблатта обучаются только веса выходного слоя  
# как и раньше обучаем подавая по одному примеру и корректируем веса в случае ошибки  
while(True):  
 print('iteration:',n\_iter)  
 n\_iter+=1  
 Wout\_copy = np.copy(Wout)  
 for xi, target, j in zip(X, y, range(X.shape[0])):  
 pr, hidden = predict(xi)   
 Wout[1:] += ((eta \* (target - pr)) \* hidden).reshape(-1, 1)  
 Wout[0] += eta \* (target - pr)  
  
 if (np.array\_equal(Wout, Wout\_copy)):  
 print('Произошло повторение веса')  
 break  
  
 y = df.iloc[:, 4].values  
 y = np.where(y == "Iris-setosa", 1, -1)  
 X = df.iloc[:, [0, 2]].values  
 pr, hidden = predict(X)  
  
 sum = 0  
 i = 0  
 for predic\_result in pr:  
 if (predic\_result[0] != y[i]):  
 sum += 1  
 i += 1  
 if(sum==0):  
 print('Все примеры обучающей выборки решены:')  
 break  
  
  
  
y = df.iloc[:, 4].values  
y = np.where(y == "Iris-setosa", 1, -1)  
X = df.iloc[:, [0, 2]].values  
pr, hidden = predict(X)  
  
sum = 0  
i =0  
for predic\_result in pr:  
 if(predic\_result[0]!=y[i]):  
 sum+=1  
 i+=1  
  
print('sum error',sum)

# Результат работы программы:

# 

Рисунок 1 Остановка обучения по первой теореме

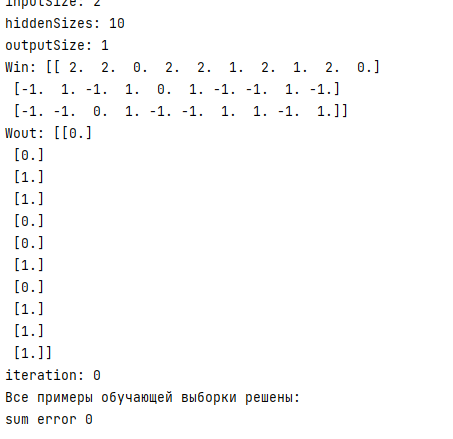


Рисунок 2 Остановка обучения по второй теореме

# Вывод:

# В ходе выполнения лабораторной работы был реализованы правила остановки алгоритма обучения перцептрона при его зацикливании или сходимости.

# 