Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе № 3

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:

ЗАДАНИЕ 1: создать класс, имеющий количество скрытых слоев >2.

ЗАДАНИЕ 2: в созданном классе изменить функцию активации слоев на ReLU и оценить изменение скорости обучения по сравнению с функцией активации Sigmoid.

# 

Рисунок Модифицированный класс MLPtorch с тремя скрытыми слоями

# Листинг программы:

import pandas as pd  
import numpy as np  
import torch  
import torch.nn as nn  
  
  
class MLPptorch(nn.Module):  
 def \_\_init\_\_(self,  
 in\_size,  
 first\_hidden\_size,  
 second\_hidden\_size,  
 third\_hidden\_size,  
 out\_size):  
 nn.Module.\_\_init\_\_(self)  
 self.layers = nn.Sequential(nn.Linear(in\_size, first\_hidden\_size),  
 nn.ReLU(),  
 nn.Linear(first\_hidden\_size, second\_hidden\_size),  
 nn.ReLU(),  
 nn.Linear(second\_hidden\_size, third\_hidden\_size),  
 nn.ReLU(),  
 nn.Linear(third\_hidden\_size, out\_size),  
 nn.Sigmoid())  
  
 # прямой проход  
 def forward(self, x):  
 return self.layers(x)  
  
  
# функция обучения  
def train(x, y, num\_iter):  
 for i in range(0, num\_iter):  
 pred = net.forward(x)  
 loss = lossFn(pred, y)  
 loss.backward()  
 optimizer.step()  
 if i % 500 == 0:  
 print('Ошибка на ' + str(i) + ' итерации: ', loss.item())  
 return loss.item()  
  
  
df = pd.read\_csv('data.csv')  
df = df.iloc[np.random.permutation(len(df))]  
  
X = df.iloc[0:100, 0:3].values  
y = df.iloc[0:100, 4]  
y = y.map({'Iris-setosa': 1, 'Iris-virginica': 2, 'Iris-versicolor': 3}).values.reshape(-1, 1)  
Y = np.zeros((y.shape[0], np.unique(y).shape[0]))  
for i in np.unique(y):  
 Y[:, i - 1] = np.where(y == i, 1, 0).reshape(1, -1)  
  
X\_test = df.iloc[100:150, 0:3].values  
y = df.iloc[100:150, 4]  
y = y.map({'Iris-setosa': 1, 'Iris-virginica': 2, 'Iris-versicolor': 3}).values.reshape(-1, 1)  
Y\_test = np.zeros((y.shape[0], np.unique(y).shape[0]))  
for i in np.unique(y):  
 Y\_test[:, i - 1] = np.where(y == i, 1, 0).reshape(1, -1)  
  
inputSize = X.shape[1] # количество входных сигналов равно количеству признаков задачи  
first\_hidden\_size = 50 # задаем число нейронов скрытого слоя  
second\_hidden\_size = 20 # задаем число нейронов скрытого слоя  
third\_hidden\_size = 10 # задаем число нейронов скрытого слоя  
outputSize = Y.shape[1] if len(Y.shape) else 1 # количество выходных сигналов равно количеству классов задачи  
  
net = MLPptorch(inputSize,  
 first\_hidden\_size,  
 second\_hidden\_size,  
 third\_hidden\_size,  
 outputSize)  
lossFn = nn.MSELoss()  
  
optimizer = torch.optim.SGD(net.parameters(), lr=0.009)  
  
loss\_ = train(torch.from\_numpy(X.astype(np.float32)),  
 torch.from\_numpy(Y.astype(np.float32)), 5000)  
  
  
pred = net.forward(torch.from\_numpy(X.astype(np.float32))).detach().numpy()  
err = sum(abs((pred > 0.5) - Y))  
print(err)  
  
pred = net.forward(torch.from\_numpy(X\_test.astype(np.float32))).detach().numpy()  
err = sum(abs((pred > 0.5) - Y\_test))  
print(err)

# Результат работы программы:

# 

Рисунок Процесс обучения с одним скрытым слоем и функцией активации sigmoid

# 

Рисунок Процесс обучения с тремя скрытыми слоями и функцией активации sigmoid

# 

Рисунок Процесс обучения с тремя скрытыми слоями и функцией активации RELU

# При функции активации RELU видно, что в процессе обучения произошел выход из точки локального минимума, также произошло «застраивание» весов, при котором ошибка постоянно равняется 0.33333432644088.

# Вывод:

В ходе выполнения лабораторной работы был реализован реализовать класс, имеющий количество скрытых слоев равное 3, с помощью MLPptorch.