Пензенский государственный университет

Факультет вычислительной техники

Кафедра «Вычислительная техника»

**Отчёт**

по лабораторной работе №3

по дисциплине «Технологии глубоких нейронных сетей»

на тему: «Основы python\_обучение нейрона»

Выполнил: ст.гр.23ВВВ1

Евстифеев Д.А

Приняли:

д.т.н. доцент Митрохин М.А.

к.т.н. доцент Юрова О.В.

Пенза 2025

**1 задание**

import numpy as np

import random

import torch

x=torch.randint(0, 10, (5, 3))

print(x)

x\_to\_float32 = x.type(torch.float32)

print(x\_to\_float32)

y = x\_to\_float32\*\*3

print(y)

a = random.randint(1, 10)

z = x\_to\_float32\*a

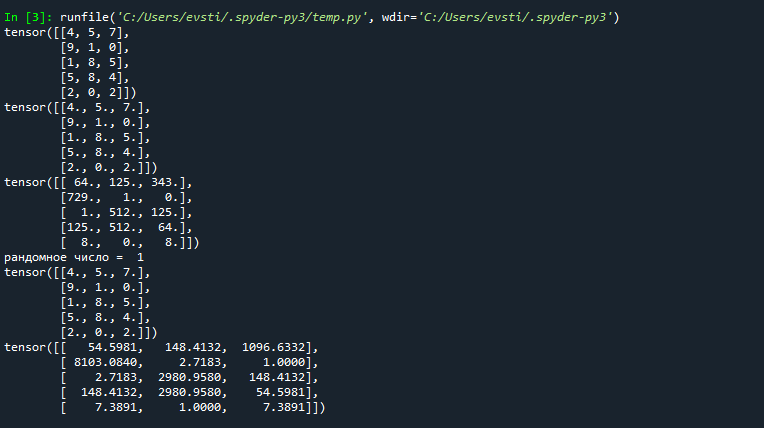
print("рандомное число = ",a)

print(z)

torch.set\_printoptions(precision= 4, sci\_mode= False)

b = torch.exp(x\_to\_float32)

print(b)



**2 задание**

import torch

import torch.nn as nn

import pandas as pd

import numpy as np

df = pd.read\_csv('data.csv')

print(df.head())

x = df.iloc[:, :4].values

y = df.iloc[:, 4].values

tmp = {"Iris-setosa": 0, "Iris-versicolor": 1}

y = np.array([tmp[tmp\_arr] for tmp\_arr in y])

x\_tensor = torch.tensor(x, dtype = torch.float32)

y\_tensor = torch.tensor(y, dtype = torch.long)

linear = nn.Linear(4, 2)

loss\_fn = nn.CrossEntropyLoss()

optimizer = torch.optim.SGD(linear.parameters(), lr=0.01)

for i in range(100):

pred = linear(x\_tensor)

loss = loss\_fn(pred, y\_tensor)

optimizer.zero\_grad()

loss.backward()

optimizer.step()

if(i + 1) % 5 == 0:

print(f'{i + 1} / {100} err: {loss.item():4f}')

with torch.no\_grad():

predictions = linear(x\_tensor)

\_, predicted\_classes = torch.max(predictions, 1)

t = torch.tensor(y, dtype=torch.long)

print("Предсказанные классы:", predicted\_classes)

print("Метки:", t)

