Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»(ВлГУ)

Кафедра информационных систем и программной инженерии

Лабораторная работа №3

по дисциплине: "Введение в искусственный интеллект"

Тема работы: Модель нейрона. Графическая визуализация вычислений в

системе MATLAB.

Выполнил:

ст. гр. ПРИ-120

Богдан С. С.

Принял:

зав. каф. ИСПИ

Озерова М. И.

Владимир, 2024

*Цель работы***:**

Изучение структурных схем модели нейрона и средств системы MATLAB, используемых для построения графиков функций активации нейрона.

*Практическое задание 1 (Вариант 5)***:**

1. Описание задания:
   1. Создайте нейронную сеть для выполнения операций, указанных на рисунке 1.1.1.
   2. Обучите нейронную сеть.
   3. Сделайте прогнозирование значений процесса.



Рис. 1.1.1 Вариант задания.

1. Задание выполнено на Python. Файл среды (рис. 1.2.1). Полный код доступен по ссылке: [vlsu-labs/iiai-labs (github.com)](https://github.com/vlsu-labs/iiai-labs)

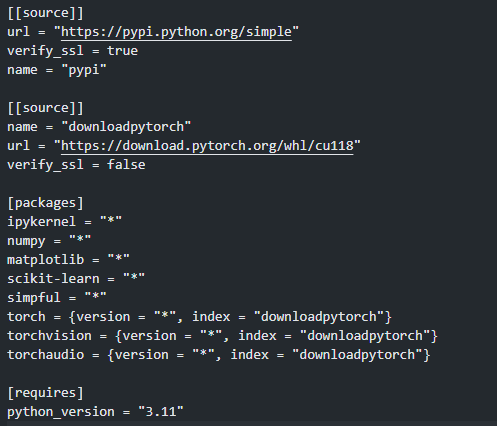


Рис. 1.2.1 Файл виртуального окружения.

1. Инициализируем и обучим нейронную сеть (рис. 1.3.1)

import numpy as np

import random

from sklearn.neural\_network import MLPRegressor

def f(x):

return np.sin(x - 2)

def init\_x(step=0, diff=0.05):

return np.arange(-1-step, 1+step, diff)

init\_arr = np.arange(-5, 5, 1)

train\_x = np.array(init\_arr)

train\_y = np.array(list(map(f, train\_x)))

model = MLPRegressor(

activation='tanh',

solver='lbfgs',

alpha=1e-5,

hidden\_layer\_sizes=(5, 2),

random\_state=1

)

model.fit([[x] for x in train\_x], [[x] for x in train\_y])

score = model.score([[x] for x in train\_x], [x for x in train\_y])

Рис. 1.3.1 Блок обучения.

1. Получим предсказания от модели и сопоставим с ожидаемыми результатами (рис. 1.4.1-1.4.2)

from matplotlib import pyplot as plt

from matplotlib.pyplot import subplots\_adjust

x = init\_x(0.5, diff=0.005)

preds = np.array(model.predict([[xi] for xi in x]))

acts = np.array(list(map(f, x)))

fig = plt.figure(num=1, facecolor='w', edgecolor='k')

fig.set\_figwidth(10)

fig.set\_figheight(10)

subplots\_adjust(hspace=0.4)

ax = fig.add\_subplot(2, 1, 1)

ax.plot(x, acts, color='red', label='act', linestyle='-.')

ax.plot(x, preds, color='green', label='predict', linestyle='-')

ax.set\_title('Сравнение')

ax.set\_xlabel('ось x')

ax.set\_ylabel('ось y')

ax.grid()

ax.legend()

ax = fig.add\_subplot(2, 1, 2)

ax.stem(x, [a - p for p, a in zip(preds, acts)], label=f'error', markerfmt='o', linefmt='-.')

ax.set\_title(f'{score=}')

ax.set\_xlabel('ось x')

ax.set\_ylabel('ось y')

ax.grid()

ax.legend()

Рис. 1.4.1 Блок предсказания.

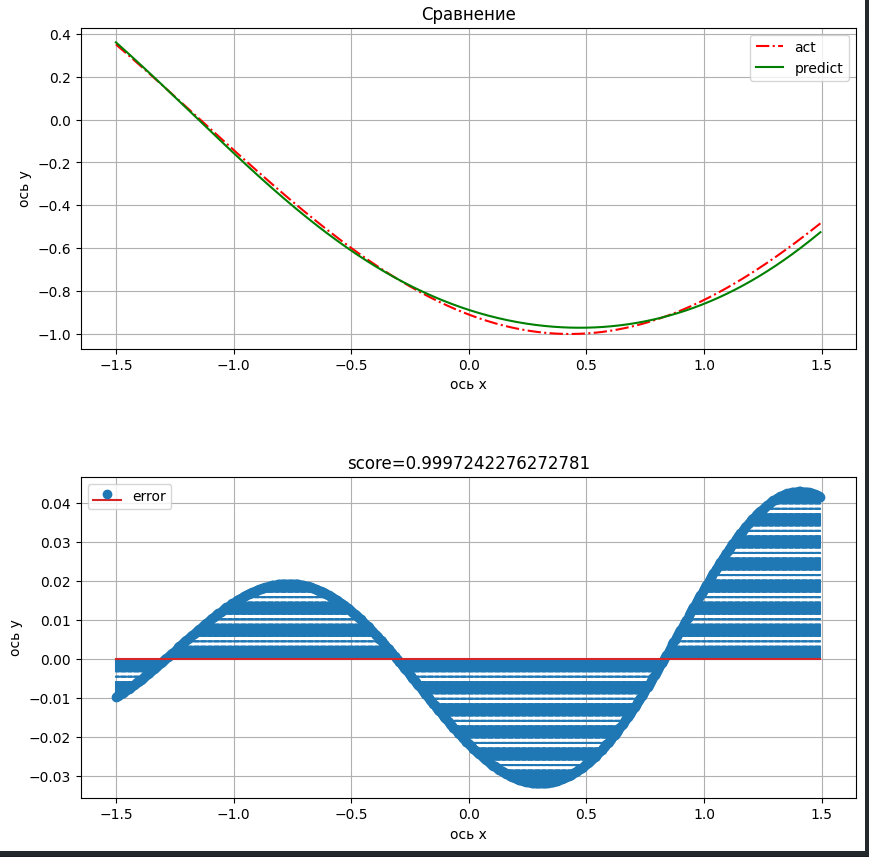


Рис. 1.4.2 Результаты предсказания.

*Вывод***:**

В ходе выполнения лабораторной работы были изучены структурные схемы модели нейрона и средства системы Python, которые используются для построения графиков функций активации нейрона.