Руководство пользователя программы NN

Оглавление

[Установка и запуск 2](#_Toc63333013)

[Windows 2](#_Toc63333014)

[Linux (на примере Ubuntu) 3](#_Toc63333015)

[Описание функций программы 3](#_Toc63333016)

[Функция “Open file” 3](#_Toc63333017)

[Функция “NN LM” 4](#_Toc63333018)

[Функция “NN Lin” 4](#_Toc63333019)

[Функция “Predict” 5](#_Toc63333020)

[Функция “Test” 5](#_Toc63333021)

[Функция “Save NN” 6](#_Toc63333022)

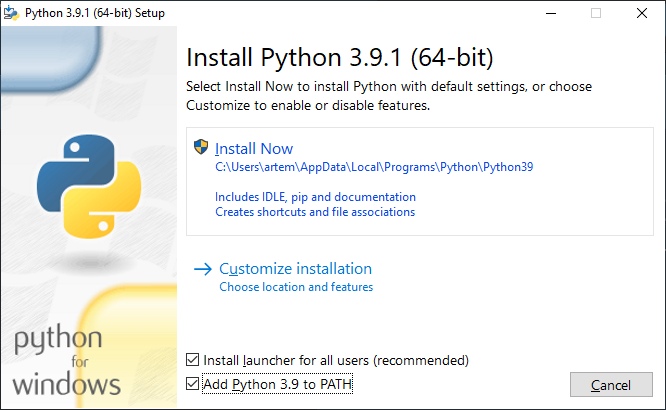
[Функция “Load NN” 6](#_Toc63333023)

[Функция “Script” 6](#_Toc63333024)

# Установка и запуск

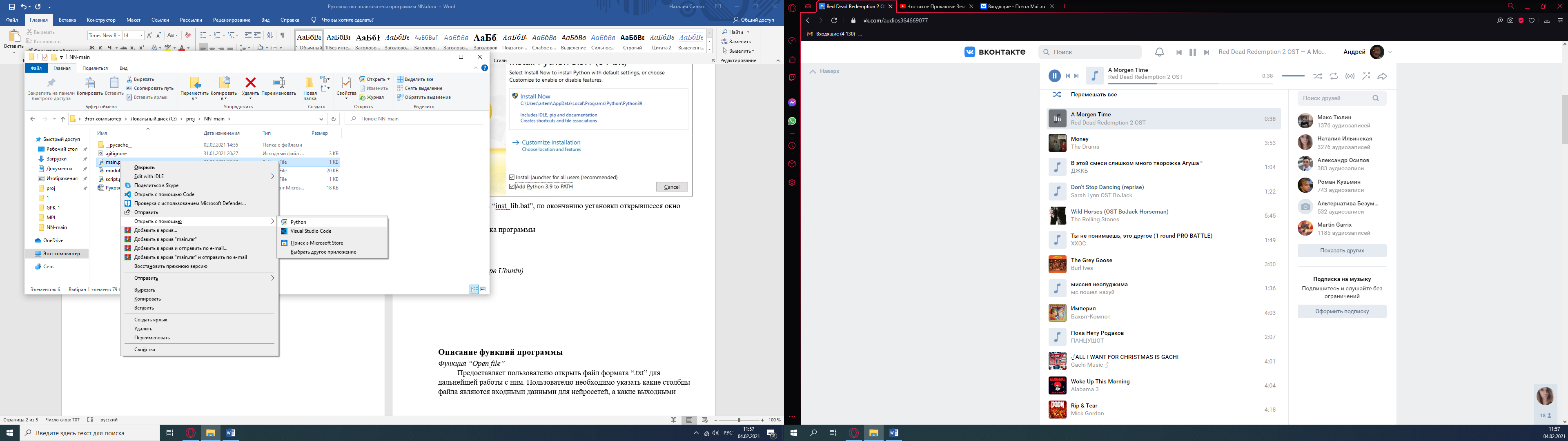
## Windows

Скачать и установить [python](https://www.python.org)(рис.1).



Запустить “inst\_lib.bat”, по окончанию установки открывшееся окно можно закрыть.

Для запуска программы нажать двойным кликом по файлу “main.py”. Если запуск не произвелся, то нажать правую кнопку мыши на файле и на вкладке “Открыть с помощью” выбрать “Python” (рис.2).



## Linux (на примере Ubuntu)

В дистрибутивах Ubuntu python уже установлен. Для проверки введите в терминале:

python3 --version

Терминал выведет установленную версию Python. Далее перейти в терминале в директорию с программой и выполнить:

sudo pip3 install -r requirements.txt

Для запуска программы выполнить в той же директории:

python3 main.py

# Описание функций программы

## Функция “Open file”

Открывает файл формата “.txt” для дальнейшей работы с ним. Пользователю необходимо указать какие столбцы файла являются входными данными для нейросетей, а какие выходными (рис.1). Программа сама предлагает пользователю указать за входные данные все столбцы файла кроме последнего, а в качестве выходного последний.

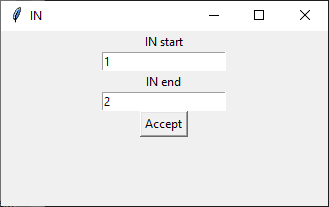
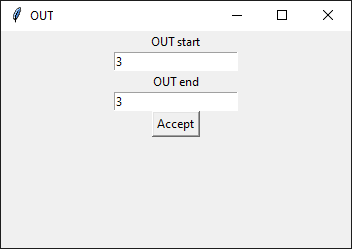
 

Рис.1. Окна указания входных/выходных данных.

## Функция “NN LM”

На основе полученных данных из открытия файла ([*Open file*](#_Функция_“Open_file”)) создается и обучается нейросеть прямого распространения (feed forward neural network). В качестве оптимизатора используется алгоритм [Левенберга — Марквардта](https://ru.wikipedia.org/wiki/Алгоритм_Левенберга_—_Марквардта).

Пользователь может выбрать количество скрытых слоев, количество нейронов на каждом скрытом слое, требуемую СКО (средняя квадратичная ошибка), максимальное количество эпох обучения, выбрать, будет ли осуществляться автоматический подбор количества нейронов, и также указать минимально и максимальное количество нейронов для слоев ([подбор количества нейронов](#_Подбор_нейронов)), делить ли входную выборку на обучающую и тестовую ([деление на выборки](#_Деление_на_выборки)), стоит ли включить защиту от переобучения ([переобучение](#_Переобучение)) (рис.2). Алгоритм создания и обучения ИНС основан на библиотеке pyrenn ([документация pyrenn](https://pyrenn.readthedocs.io/en/latest/index.html)).

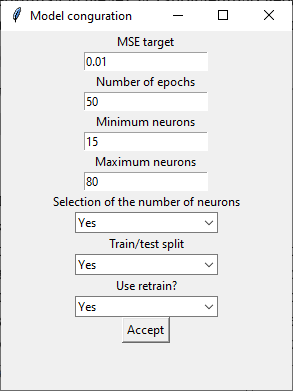


Рис.2. Окно настройки параметров ИНС функции “NN LM”

Также в процессе обучения в консоль выводятся данные СКО по каждой эпохе.

По окончанию работы функции пользователю предоставляется график ([графики](#_Графики)).

## Функция “NN Lin”

На основе полученных данных из открытия файла ([*Open file*](#_Функция_“Open_file”)) создается и обучается нейросеть обратного распространения ошибки (feed forward neural network). В качестве оптимизатора используется алгоритм [ADAM](https://ru.wikipedia.org/wiki/Стохастический_градиентный_спуск#Adam).

При запуске этой функции пользователю необходимо указать минимум и максимум для каждого столбца выходных данных, по умолчанию используются минимумы и максимумы входных данных. Пользователь может выбрать количество скрытых слоев, количество нейронов на каждом скрытом слое, функцию активацию для каждого скрытого слоя ([Sigm](https://ru.wikipedia.org/wiki/Сигмоида), [Tanh](https://ru.wikipedia.org/wiki/Гиперболические_функции), [ReLu](https://en.wikipedia.org/wiki/Rectifier_(neural_networks)), [LeakyReLu](https://en.wikipedia.org/wiki/Rectifier_(neural_networks)#Leaky_ReLU)), требуемую СКО (средняя квадратичная ошибка), максимальное количество эпох обучения, скорость обучения, выбрать, будет ли осуществляться автоматический подбор количества нейронов, и также указать минимально и максимальное количество нейронов для слоев ([подбор количества нейронов](#_Подбор_нейронов)), делить ли входную выборку на обучающую и тестовую ([деление на выборки](#_Деление_на_выборки)), стоит ли включить защиту от переобучения ([переобучение](#_Переобучение)) (рис.3).

Алгоритм создания и обучения ИНС основан на библиотеке pytorch ([документация pytorch](https://pytorch.org/docs/stable/index.html)).

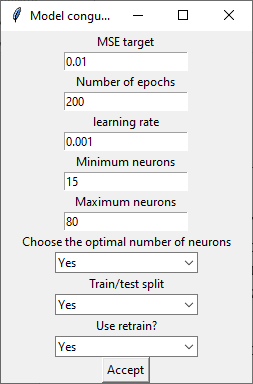


Рис.3. Окно настройки параметров ИНС функции “NN Lin”

Также в процессе обучения в консоль выводятся данные СКО каждые 10 эпох.

По окончанию работы функции пользователю предоставляется график ([графики](#_Графики)).

## Функция “Predict”

Производит прогноз с помощью созданной или открытой (функцией [Load NN](#_Функция_“Load_NN”)) нейросети и входов открытого файла (для прогноза используются входы, указанные пользователем, а не весь файл). И сохраняет полученный массив в указанном пользователе файле с форматом “.txt”.

## Функция “Test”

Производит прогноз с помощью созданной или открытой (функцией “[Load NN](#_Функция_“Load_NN”)”) нейросети и входов открытого файла (для прогноза используются входы, указанные пользователем, а не весь файл). И выводит СКО для прогноза (сравнивает столбцом/столбцами, указанными пользователем как выходные данные).

## Функция “Save NN”

Сохраняет созданную ИНС в файл.

## Функция “Load NN”

Загружает ИНС из файла созданную функцией “[Save NN](Save_NN#_Функция_)”.

## Функция “Script”

Используется для вызова написанного разработчиком скрипта в файле “script.py”. По умолчанию неактивна. Подробности по написанию скриптов см. «Руководство разработчика программы NN».

# Дополнительно

## Подбор нейронов

При указание пользователем параметра “selection of the number of neurons” при создании ИНС, перед обучением производится перебор всех значений количества нейронов в промежутке, указанном пользователем. Для этого для каждой конфигурации создается объект ИНС, обучается одна эпоха и сохраняется результат СКО. Конфигурация с наименьшей СКО обучается далее.

В случае, если пользователь выбрал автоматический подбор количества нейронов, то количество нейронов, указанное пользователем, ранее будет игнорироваться, поэтому если пользователь планирует использовать автоматический подбор количества нейронов, то на моменте указания количества нейронов для каждого слоя можно указывать значения, предложенные программой по умолчанию.

## Деление на выборки

При указании пользователем параметра “train/test split” при создании ИНС, перед обучением оригинальная выборка делится на обучающую и тестовую. В качестве обучающих данных берется 90% случайных значений начальной выборки, а в качестве тестовых остальные 10%. Значения выборок берутся случайно, но паттерн получения этих случайных значений одинаков, так что использование одних и тех же входных данных приводит к одному и тому же результату на одном и том же устройстве.

## Переобучение

При указании пользователем параметра “use retrain?” при создании ИНС, перед обучением оригинальная выборка делится на обучающую и тестовую, аналогично [деление на выборки](#_Деление_на_выборки). Затем в процессе обучения проверяется СКО для тестовой выборки, если СКО растет на протяжении 3 эпох, обучения прерывается.

Если указан параметр “use retrain?”, но не указан “train/test split”, то разделение на обучающую и тестовую выборку все равно происходит.

## Графики

После обучения пользователю предоставляются графики.

Если пользователь указал обучение без [деления на выборки](#_Деление_на_выборки), то рисунок состоит из графика оригинальных значений, и графика спрогнозированных значений. Если пользователем указано [деление на выборки](#_Деление_на_выборки) или защита от [переобучения](#_Переобучение), то предоставляется 2 рисунка, на 1-ом: точечный график значений обучающей выборки, и точечный график значений на основе входов обучающей выборки (пример рис. 4), на 2-ом: точечный график значений тестовой выборки, и точечный график значений на основе входов тестовой выборки (пример рис. 5).

Если в оригинальных данных имеется более одного входа или выхода, то для построения графиков используется только первый вход и первый выход.

Также в окне рисунка имеются данные СКО и конфигурации модели.

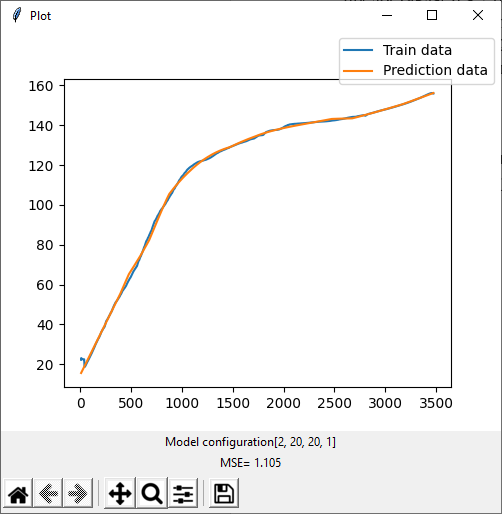
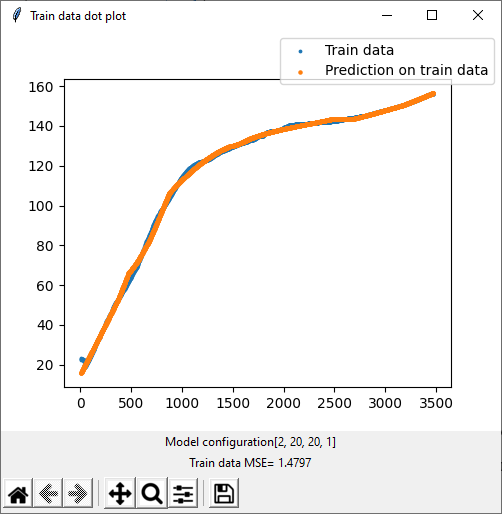


Рис.4, пример графиков для обучения без деления на выборки



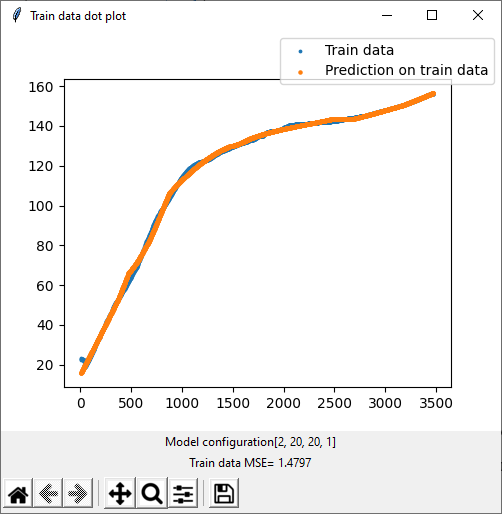


Рис.5, пример графиков для обучения с делением на выборки