УДК 004.032.26

Использование программ Neural Network Wizard 1.7 и библиотеки Scikit-learn в Phyton для создания и обучения нейронной сети

Осмонова Бермет Майрамбековна
Нарынский государственный университет им. С.Нааматова
Преподаватель кафедры «Информационные технологии»
Приамурский государственный университет имени Шолом-Алейхема
магистрант

Аннотация
Цель исследования: приобретение практических навыков применения нейронных сетей с использованием пакета Neural Network Wizard (программный эмулятор нейронных сетей) и программы Jupyter Notebook (Anaconda Navigator). Методика исследования: с помощью программы Neural Network Wizard реализована многослойная нейронная сеть, обучаемая по алгоритму обратного распространения ошибки (back propagation error). Использован язык программирования Python для создания проекта машинного обучения, в котором существует разнообразие открытых библиотек, что дает возможность анализировать данные. Проведено исследование с целью выбора оптимальной конфигурации нейронной сети, позволяющей наилучшим образом решить поставленную задачу. Результатом работы программы Jupyter Notebook является файл, который хранит в себе все параметры полученной нейронной сети. В работе показан простой пример с двумя входами и одним выходом и обучение данных с использованием библиотеки Scikit-learn. Было наглядно показано библиотека для обработки и анализа данных, библиотека с поддержкой многомерных массивов, разделение столбцов, строк входных и выходных данных, а также использованы команды на выборку, обучение и предсказание нейронных сетей.

Ключевые слова: нейрон, нейронная сеть, искусственный интеллект, области применения нейронных сетей, эмулятор, библиотеки, модуль, слои, эпохи.

Using Neural Network Wizard 1.7 and Scikit-learn libraries in Phyton for creating and training a neural network

Osmonov Bermet Meirambekov
Naryn State University named after S. Naamatov
Teacher of the Department " Information technologies»
Sholom-Alechem Priamursky State University
master's student

Abstract
The purpose of the research: to acquire practical skills in the use of neural networks using the Neural Network Wizard package (software emulator of neural networks) and the Jupyter Notebook program (Anaconda Navigator). Research methodology: using the Neural Network Wizard program, a multi-layer neural network is implemented, which is trained using the back propagation error algorithm. We used the Python programming language to create a machine learning project that has a variety of open libraries, which makes it possible to analyze data. A study was conducted to select the optimal configuration of the neural network, which allows to solve the problem in the best way. The result of the Jupyter Notebook program is a file that stores all the parameters of the resulting neural network. This paper shows a simple example with two inputs and one output and data training using the Scikit-learn library. A library for data processing and analysis, a library with support for multidimensional arrays, separation of columns and rows of input and output data, and commands for sampling, training, and prediction of neural networks were clearly shown.
Keywords: neuron, neural network, artificial intelligence, applications of neural networks, emulator, libraries, module, layers, epochs.

Научный руководитель:
Баженов Руслан Иванович
Приамурский государственный университет имени Шолом-Алейхема
К.п.н., доцент, зав. кафедрой информационных систем, математики и правовой информатики

Введение.

Актуальность исследования

Изучение и исследование нейронных сетей, сравнение с человеческим мозгом отличается от цифрового компьютера тем, что обработка информации в человеческом мозгу превышает миллионы раз. Тем не менее, ученые хорошо продвинулись в направлении искусственного интеллекта. Человеческий мозг – это сложный, нелинейный компьютер который состоит из структурных компонентов - нервных клеток, нейронами. Нейроны создают соединения, и сигналы по нейронной сети вызывают активность человеческого мозга за короткий интервал времени, также обладает способностью организовывать работу нейронов, так, чтобы они могли выполнять конкретные задачи (такие как распознавание образов, обработку сигналов органов чувств, моторные функции) во много раз быстрее, чем могут позволить самые быстродействующие современные компьютеры. Благодаря искусственному интеллекту и при помощи современных суперкомпьютеров может решать самостоятельно алгоритмические, математические, сверхсложные программные задачи и оптимизации процессов управления. А нейронные сети являются основным направлением по изучению возможности моделирования естественного интеллекта с

помощью алгоритмов. Стоит отметить, что нейросеть не программируются, а обучается [2].

Нейронная сеть — это последовательность нейронов, соединенных между собой синапсами. Нейронная сеть и его структура, использование в программировании пришла из биологии. Структура нейронных сетей помогает компьютеру анализировать и запоминать информацию различного вида, а также способна воспроизводить ее.

Нейронные сети – одно из направлений искусственного интеллекта, цель которого – смоделировать аналитические механизмы, осуществляемые человеческим мозгом. Задачи, которые решает типичная нейронная сеть – классификация, предсказание и распознавание[14].

Обзор исследований

Ю. П. Малыгина в своей статье анализировала и написала, что одной из главных особенностей нейросетей является то, что они обучаемы. Существуют различные методы обучения (с учителем, без наставника, смешанные), но все они основаны на изучении примеров из загруженной базы данных. Процесс обучения достаточно прост: из базы данных выбирается пример, который проходит через нейронную сеть в виде сигнала, затем на выходе сеть выдает ответ, и если ошибка для данного ответа мала, то сеть обучена, иначе происходит подстройка весов, и обучение начинается сначала. Этой способностью к обучению нейронные сети и отличаются от традиционных алгоритмов, у которых есть четкий порядок вычислений, наличие формул и т.д.[5]. В телекоммуникационных сетях связи В.В. Федотов рассматривает в своей работе, что нейронная сеть и область применения искусственных нейронных сетей постоянно расширяется. В телекоммуникационных системах они находят применение при решении следующих важных задач: управление коммутацией, адаптивная маршрутизация, управление трафиком, оптимальное распределение загрузки каналов сети. Кроме того, перспективным является использование нейросетевых алгоритмов в задачах кодирования и декодирования информации [3]. П.П. Степанов написал про искусственные нейронные сети и трактует что, для решения какой-либо прикладной задачи с помощью нейронной сети необходимо, во-первых, определить тип решаемой задачи. Во-вторых, выявить входные и выходные данные в задаче. В-третьих, подобрать топологию нейронной сети. В-четвертых, нормализовать данные под выбранную нейросеть. В-пятых, экспериментально подобрать параметры. В-шестых, обучить нейронную сеть. И на последнем этапе необходимо проверить качество работы нейронной сети, проанализировать количество ошибок на общее число проверок. Как правило, после одной итерации такого алгоритма полученная сеть не удовлетворяет всем требованиям задачи. Обычно необходимо около 10 таких итераций.

Основные задачи, для решения которых используются нейронные сети, это распознавание образов (текстов, звуков, изображения), прогнозирование, нейросетевое сжатие данных, попытка создания ИИ Джефом Хокинсом

(НТМ), принятие решений и управление (автомобили, роботы), ассоциативная память [7].

Использование нейронных сетей и положительные примеры рассматривают иностранные исследователи как, L. M. Manevitz, в своих трудах о классификация когнитивных состояний мозговой деятельности с помощью одноклассных нейронных сетей с выбором признаков с помощью генетических алгоритмов [9], Jürgen Schmidhuber рассматривал глубокое контролируемое обучение (также резюмируя историю обратного распространения), неконтролируемое обучение, усиленное обучение и эволюционные вычисления, а также косвенный поиск коротких программ, кодирующих глубокие и большие сети[11] и т.д.

В современном мире нейронные сети имеют колоссальный охват, ученые считают исследования, проводимые в области изучения поведенческих особенностей и состояний нейронных сетей, крайне перспективными. Перечень областей, в которых нейронным сетям нашлось применение, огромен. Это и распознавание и классификация образов, и прогнозирование, и решение аппроксимационных задач, и некоторые аспекты сжатия данных, анализа данных и, конечно, применение в системах безопасности различного характера[6].

Нейрон — это вычислительная единица, которая получает информацию, производит над ней простые вычисления и передает ее дальше. Они делятся на три основных типа: входной (синий), скрытый (красный) и выходной (зеленый). В том случае, когда нейросеть состоит из большого количества нейронов, вводят термин слоя. Соответственно, есть входной слой, который получает информацию, n скрытых слоев (обычно их не больше 3), которые ее обрабатывают и выходной слой, который выводит результат. У каждого из нейронов есть 2 основных параметра: входные данные (input data) и выходные данные (output data). В случае входного нейрона: input=output. В остальных, в поле input попадает суммарная информация всех нейронов с предыдущего слоя, после чего, она нормализуется, с помощью функции активации (пока что просто представим ее f(x)) и попадает в поле output [12].

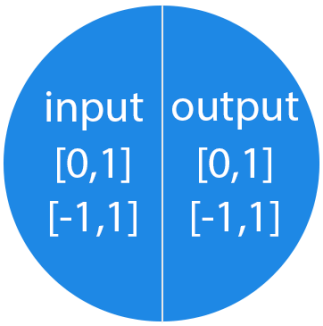

Рис.1. Параметры нейронов

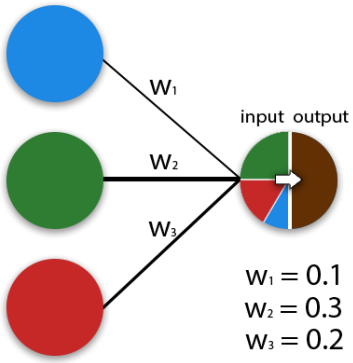

Рис.2. Синапс и его результат

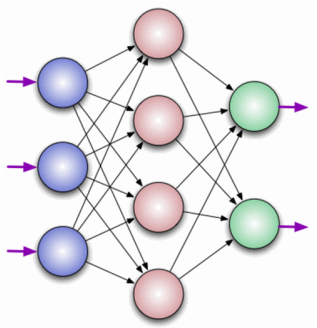

Рис.3. Входной и выходной слой

Рассмотрим понятие синапс. Синапс – обспечивает связь между нейронами. И у него сеть параметр - вес. С помощью синапса изменяется входная информация, при передачи отодного нейрона к другому нейрону. И совокупонсть весов в нейронной сети – это мозг всей системы. Именно благодаря этим весам, входная информация обрабатывается и превращается в результат [12].

На Рис.4. показана схема искусственного нейрона, где:

n — число входов нейрона,

x_i — значение i-го входа нейрона,

w_i — вес i-го синапса.

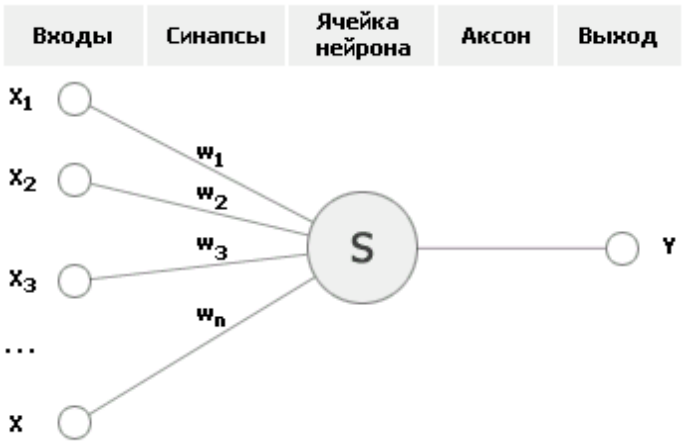

Рис.4. Схема искусственного нейрона

Также с помощью нейронных сетей есть возможность проводить распознавание лиц в реальном времени, т.е. распознавания лиц с веб камеры, видеонаблюдения, или любых других фото/видео источников.

Основные области применения нейронных сетей: автоматизация процесса классификации, автоматизация прогнозирования, автоматизация процесса распознавания, автоматизация процесса принятия решений; управление, кодирование и декодирование информации; аппроксимация зависимостей и др. С помощью нейронных сетей решается задача в области телекоммуникаций (нахождение оптимального пути трафика между узлами) проектировании сетей, также в распознавание речи – одна из наиболее популярных областей применения нейронных сетей. Нейронные сети применяются также для прогнозирования краткосрочных и долгосрочных тенденций в различных областях (финансовой, экономической, банковской и др.). В управлении цен и производства т.е. нейронная сеть, система выявляет зависимость и объем продажи, цены, маркетинг и.т.д. В медицине с помощью нейронных сетей определяют диагностирование слуха у младенцев.

Принципы работы программного эмулятора нейрокомпьютера Neural Network Wizard 1.7

Neural Network Wizard 1.7 – программный эмулятор нейрокомпьютера. В Neural Network Wizard реализована многослойная нейронная сеть, обучаемая по алгоритму обратного распространения ошибки. Программа может применяться для анализа информации, построения модели процессов и прогнозирования. Для работы с системой необходимо проделать следующие операции:

- осуществить сбор статистики по процессу;
- выполнить обучение нейросети на приведенных данных;
- произвести проверку полученных результатов.
- На примере рассмотрим сложение двух чисел и их решение.

На 1-м этапе в текстовом файле с расширением .txt записываем входные и выходные данные. В нашем случае s1, s2 – это переменные, res-это выходная переменная. Все значения вводятся в колонках. Количество примеров должен быть большим. Вся информация подается в числовом виде. Текстовая информация должна каким либо методом конвертироваться в числовой.

Выбираем файл с обучающей выборкой.

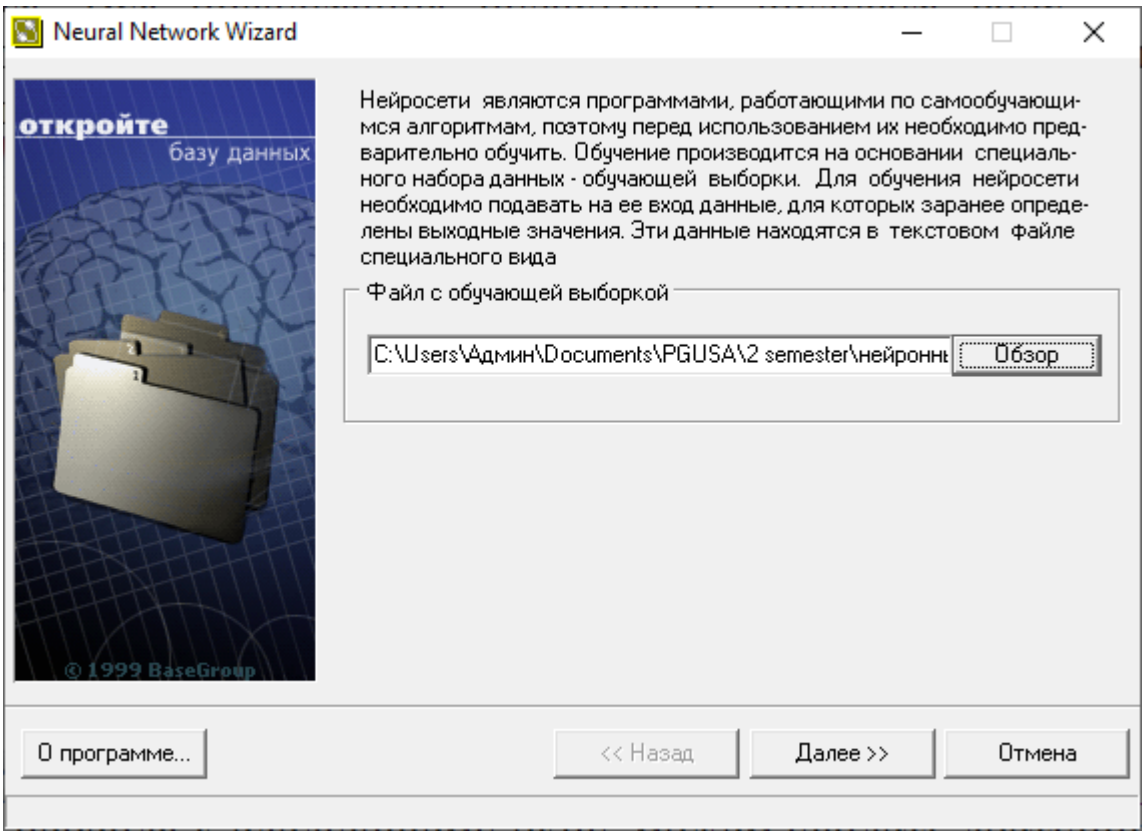

Рис.5

На 2-м этапе, перейдя к следующему окну, можем увидеть доступные поля. В поле *«Список доступных полей»* указываем переменную. Если данная переменная входная, то в группе *«Использовать поле как»* выбирается вариант «Входное», если выходная – «Целевое». На вкладке *«Нормализовать поле как…»*. На вход нейросети должна подаваться информация в нормализованном виде, т.е. в виде чисел в диапазоне от 0 до 1. Вы можете выбрать метод нормализации:

- (X-MIN)/(MAX-MIN) – линейная нормализация;
- 1/(1+exp(ax)) – экспоненциальная нормализация;
- авто – нормализация, основанная на статистических характеристиках выборки;
- без нормализации – нормализация не производится.

Указываем, что поле RES – целевое. Т.е. нейросеть будет пытаться определить, каким образом значения полей S1 и S2 влияют на поле RES.

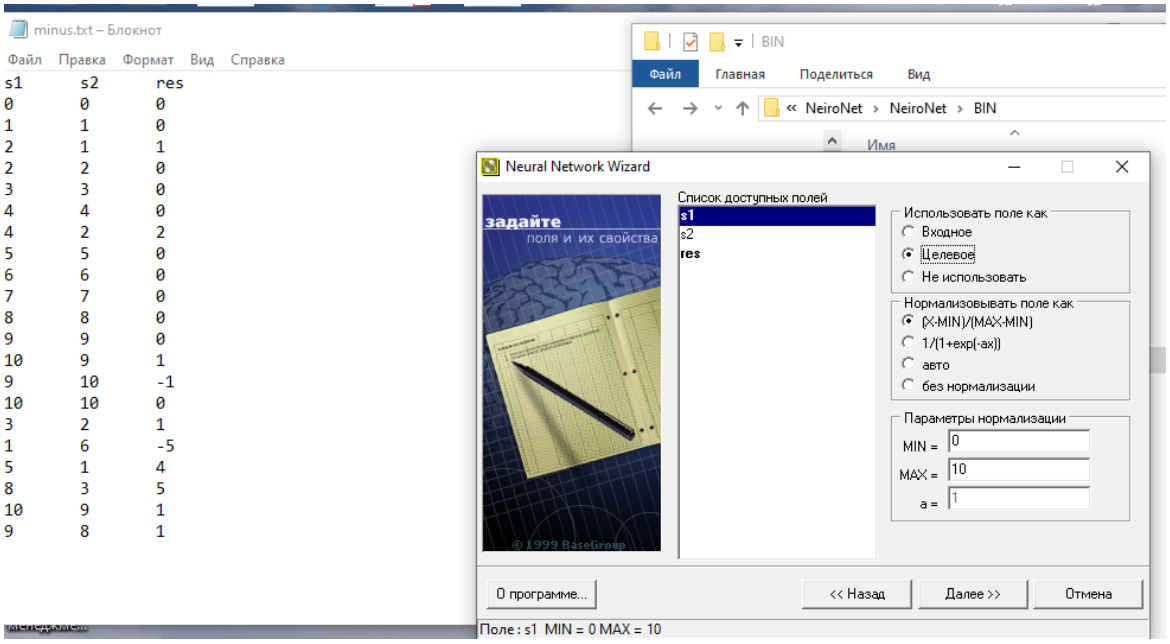

Рис.6

На 3-м этапе, определяем конфигурацию нейронной сети. Зададим количество скрытых слоев и число нейронов во входном и выходном слое.

«Вид сигмоиды…». Симоида - это гладкая монотонная возрастающая нелинейная функция, имеющая форму буквы «S», которая часто применяется для «сглаживания» и применяется для обеспечения нелинейного преобразования данных. В противном случае, нейросеть сможет выделить только линейно разделимые множества. Чем выше параметр, тем больше переходная функция походит на пороговую.

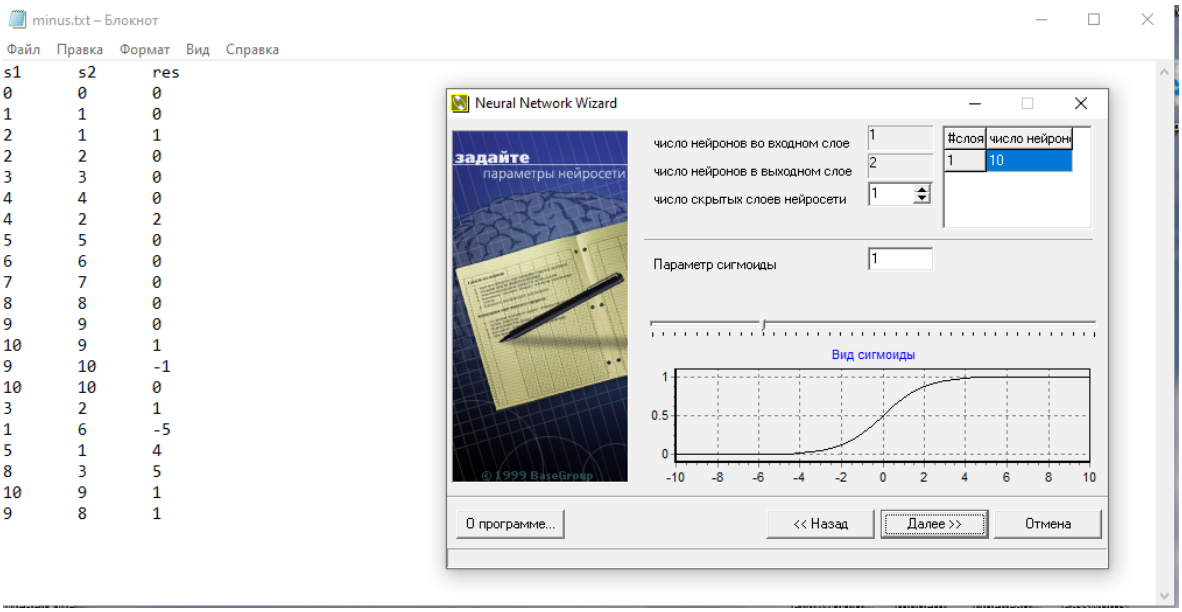

Рис.7

На 4-м этапе, рассмотрим параметры обучения:

Скорость обучения. Параметр определяет амплитуду коррекции весов на каждом шаге обучения.

Момент. Параметр определяет степень воздействия i-ой коррекции весов на i+1-ую.

Если результат прогнозирования отличается от значения из обучаемого множества меньше указанной величины, то пример считается распознанным.

Использовать тестовое множество как валидационное. При установке этого флага обучение будет прекращено с выдачей сообщения, как только ошибка на тестовом множестве начнет увеличиваться. Это помогает избежать ситуации переобучения нейросети.

Критерии остановки обучения. Необходимо определить момент, когда обучение будет закончено.

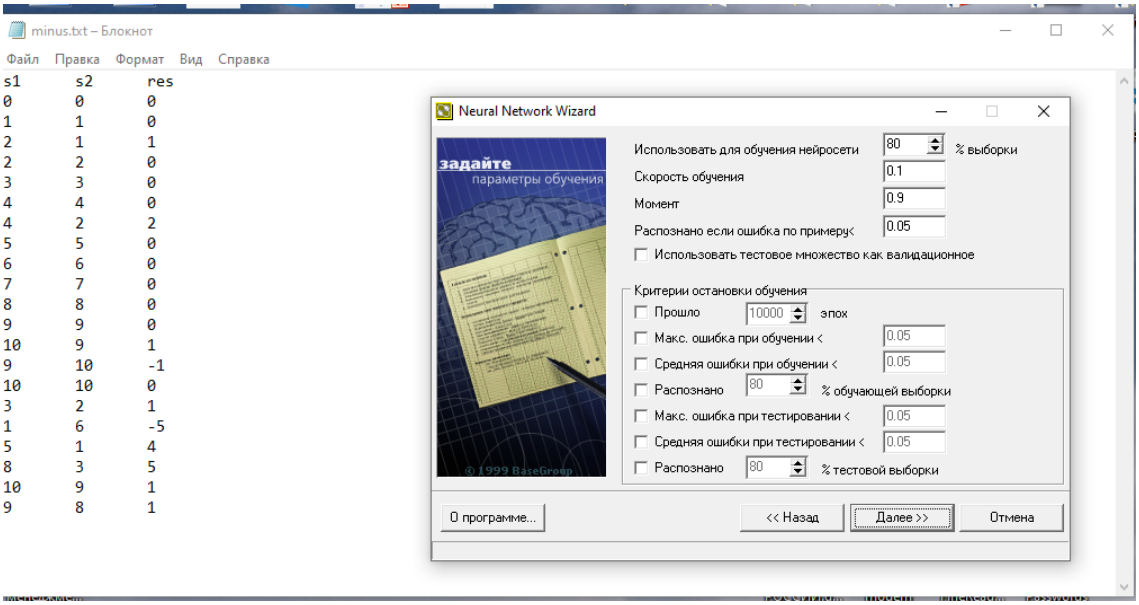

Рис.8

На 6-м этапе, Запускаем систему на обучение, в процессе которого система построит модель операции сложения.

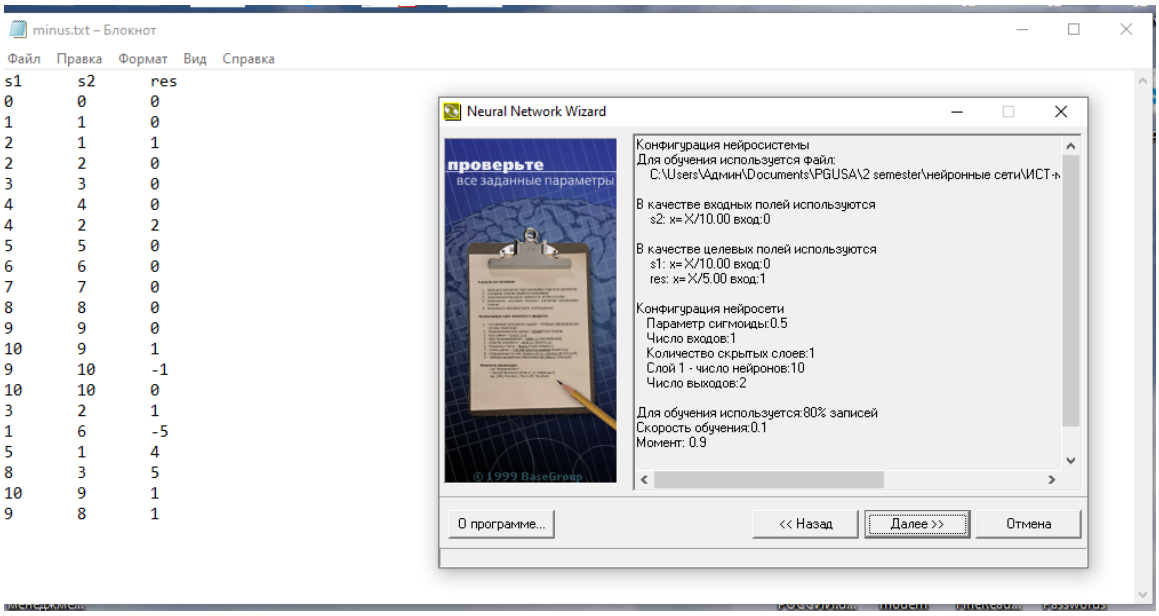

Рис.9

На 7-м этапе, можем увидеть построение модели операции сложения. «Распределение примеров в обучающей/тестовой выборке…». На этих графиках вы можете отслеживать, насколько результаты, предсказанные нейронной сетью, совпадают со значениями в обучающей (слева) и тестовой (справа) выборке. Каждый пример обозначен на графике точкой. Если точка попадает на выделенную линию (диагональ), то значит нейросеть предсказала результат с достаточно высокой точностью. Если точка находится выше диагонали, значит нейросеть недооценила, ниже – переоценила. Необходимо добиваться, чтобы точки располагались как можно ближе к диагонали. В нашем случае, точка находится выше диагонали.

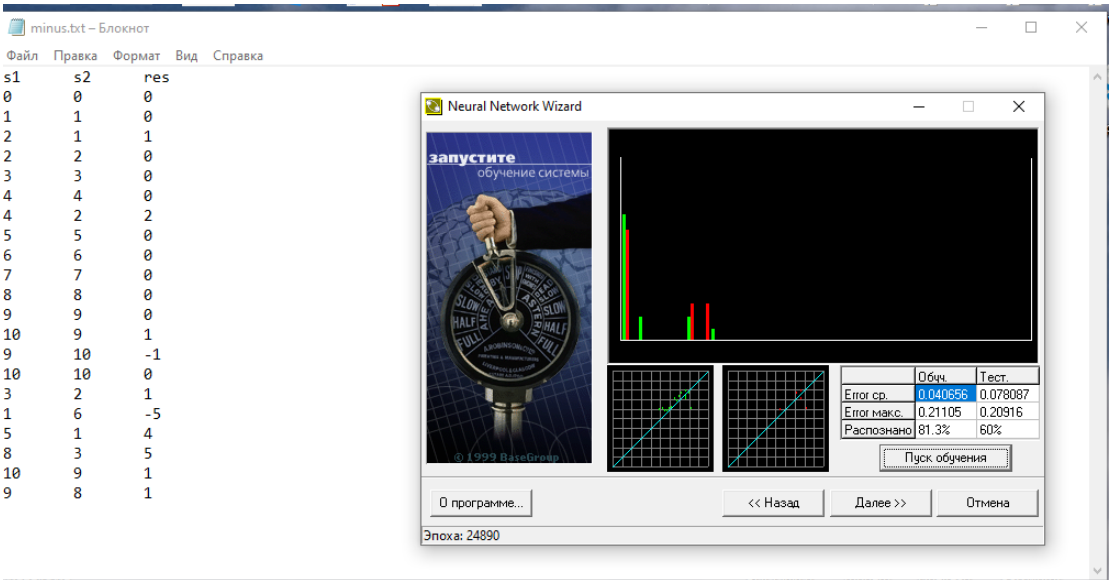

Рис.10

На последнем этапе, вводим входные параметры и делаем расчет. Neural Network Wizard сохранить файл с расширением nnw.

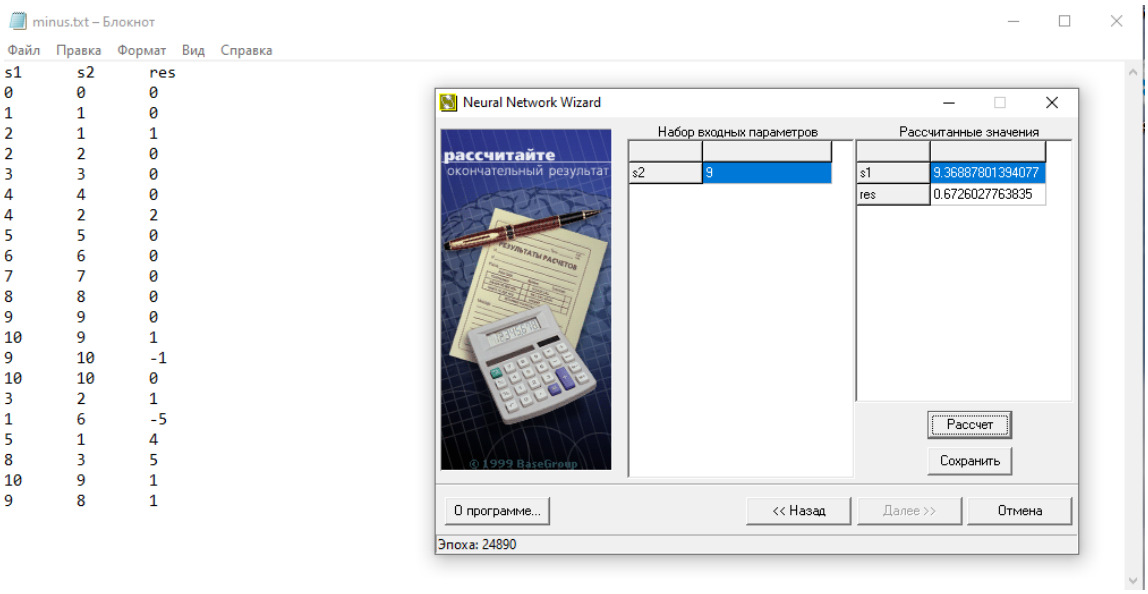

Рис.11

Работа с библиотекой sckit-learn

Для работы с Python и написания программы установить дистрибутив Anaconda (использовать для работы Jupiter Notebook)

Шаг 1. Установка на компьютер Anaconda Navigator и открываем Anaconda Prompt. Conda написан на чистом Python, что облегчает его использование в виртуальных средах Python. Кроме того, Conda подходит для библиотек C, пакетов R, Java и т.д. Anaconda — это дистрибутивы Python и R. Он предоставляет все необходимое для решения задач по анализу и обработке данных (с применимостью к Python).

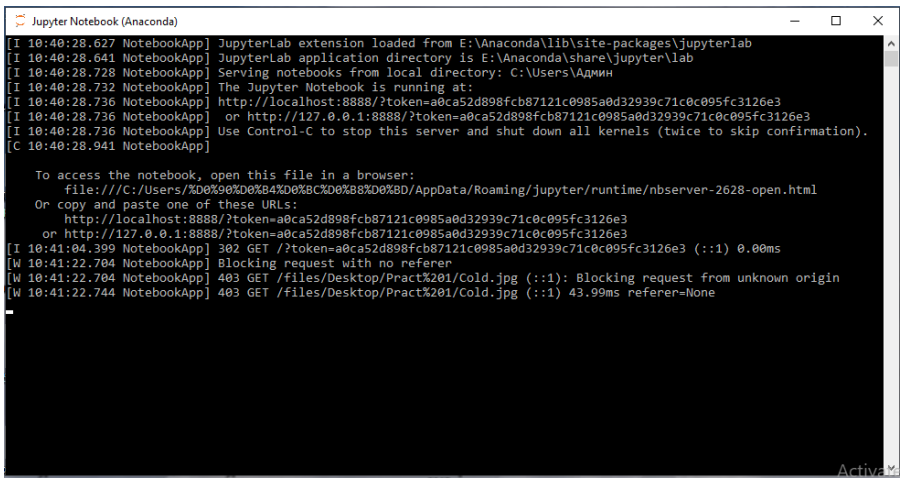

Рис.12

В открывшемся консоли набираем команду python version и проверяем версию программы. Также при наборе команды conda info выодится полная информация о программе: месторасположение, версия python, conda, используемая платформа и т.д.

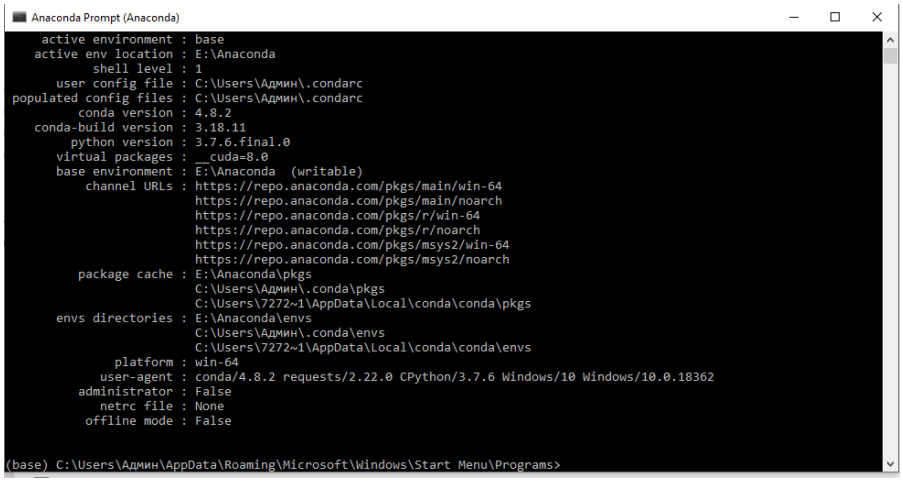

Рис.13. Шаг 2. Инициализация сети

Первым делом инициализируем компоненты сети. Импортируем numpy — это фундаментальный пакет, необходимый для научных вычислений на Python, добавляющая поддержку больших многомерных массивов и матриц.

Установливаем пакет numpy с помощью команды conda install numpy или import numpy.

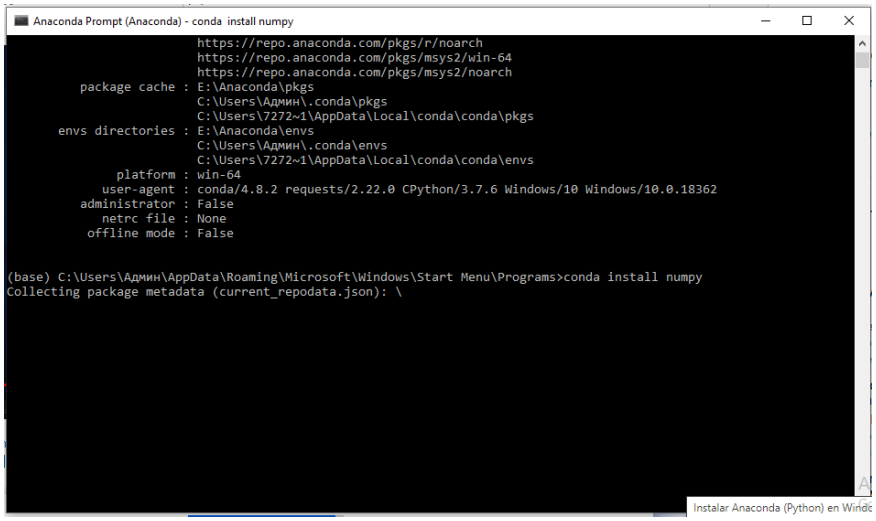

Рис.14

После # импортируем scipy.special, scipy модуль для оптимизации, интегрирования, специальных функций, обработки изображений и ит.д.

import scipy.special

import matplotlib.pyplot - для визуализации данных.

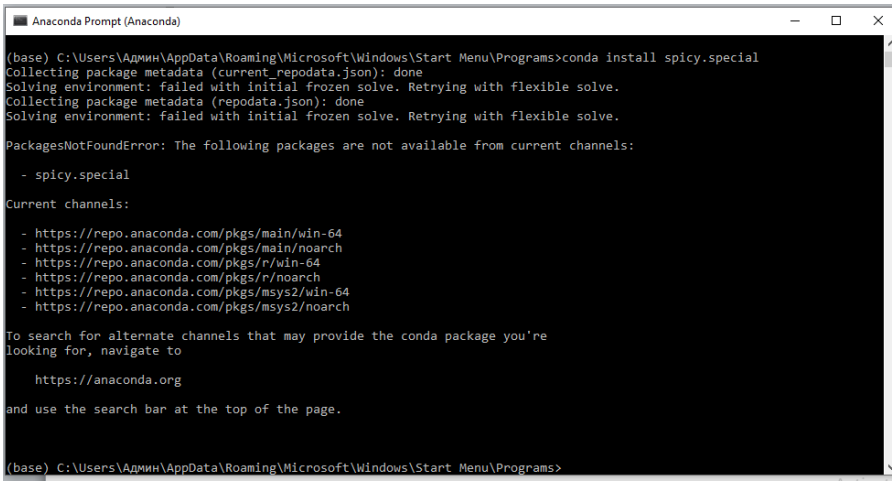

Рис.15

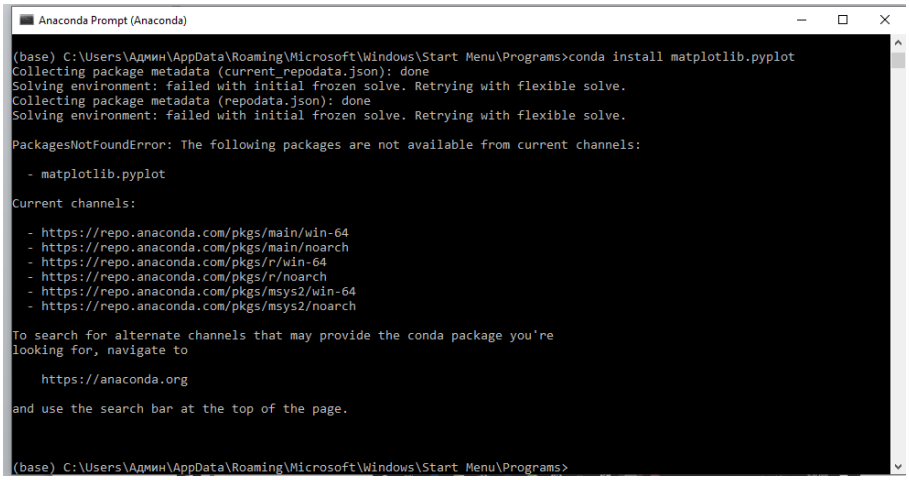

Рис.16

После загрузки пакета мы обновили conda.

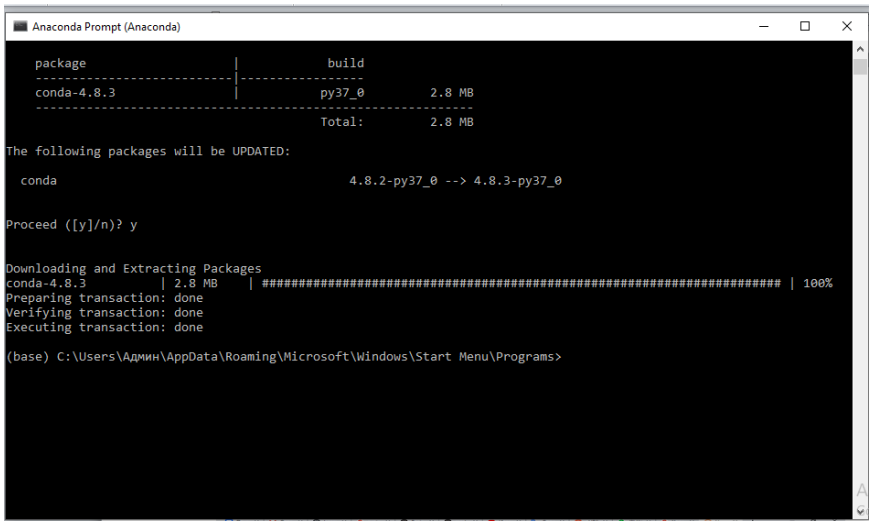

Рис.17

После обновления для открытия блокнота консоль предоставил ссылку на браузер.

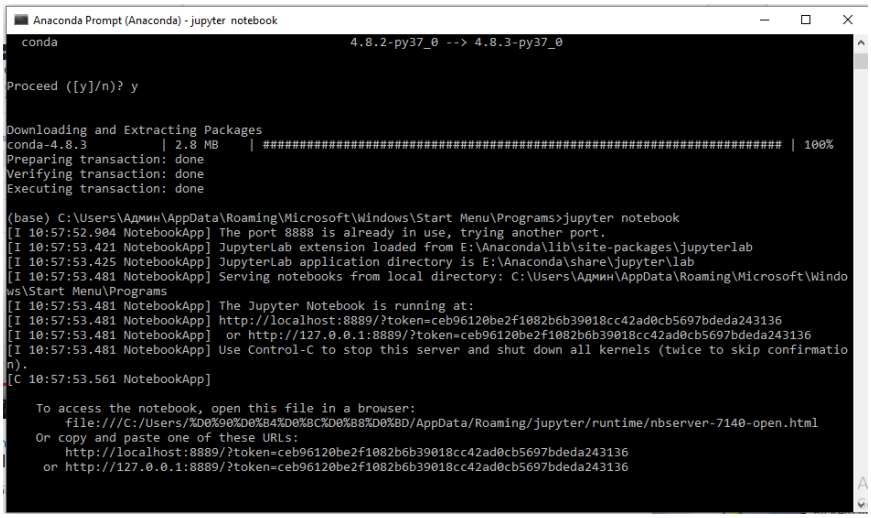

Рис.18

В окне браузера можем увидеть localhost - стандартное, официально зарезервированное доменное имя для частных IP-адресов.

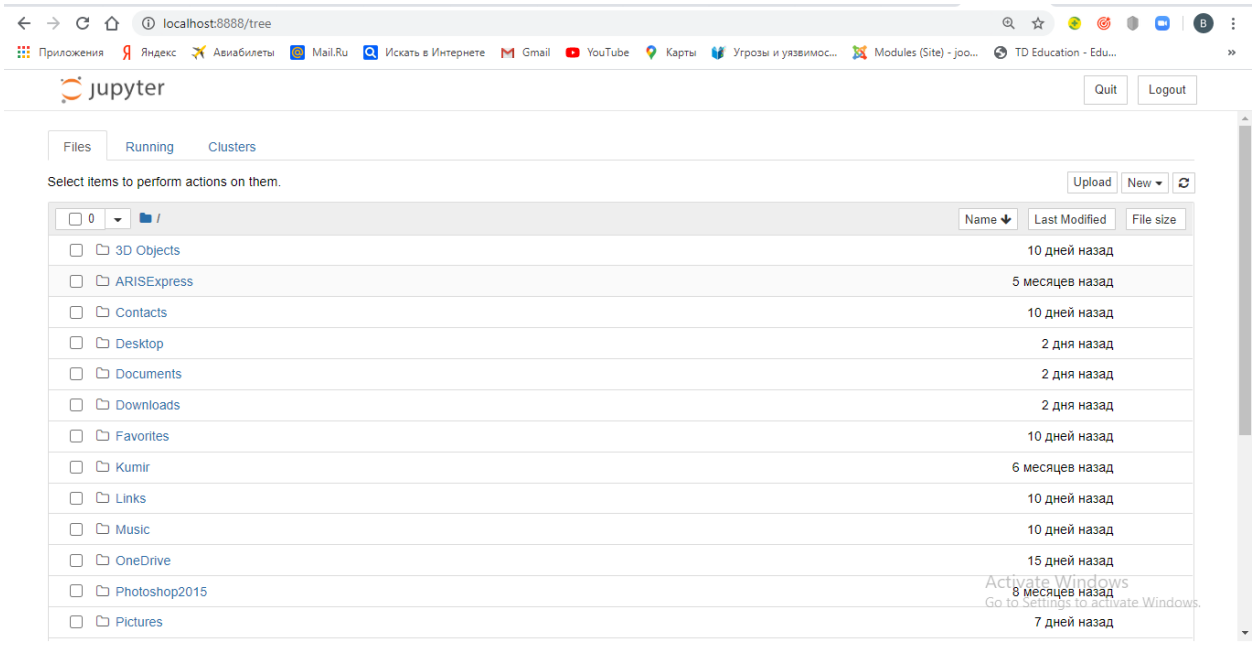

Рис.19

Здесь создали папку python. А также создали minus файл расширением ipynb.

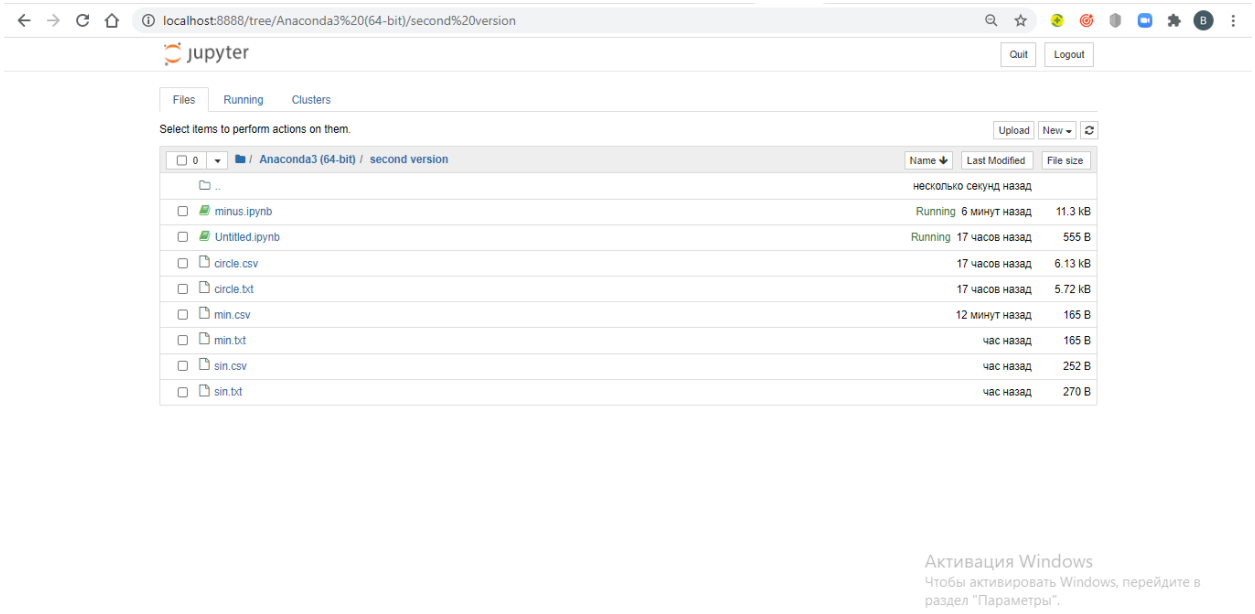

Рис.20

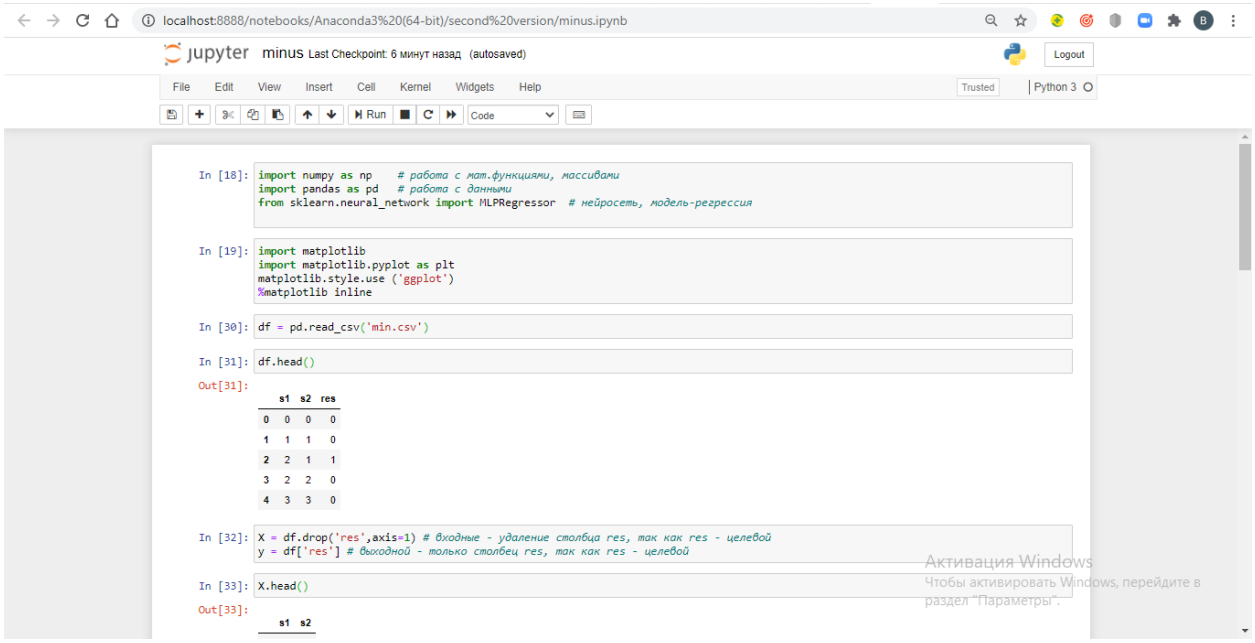

Рис.21

Проимпортировали библиотеки pandas(для работы с данными) и numpy (библиотека для работы с массивами, математическими функциями).

Загрузили файл “минус” в ipynb файл. После мы получили результаты, разделение столбцов:

1. Входные, удаление столбца
2. Выходные, только столбец “результат”

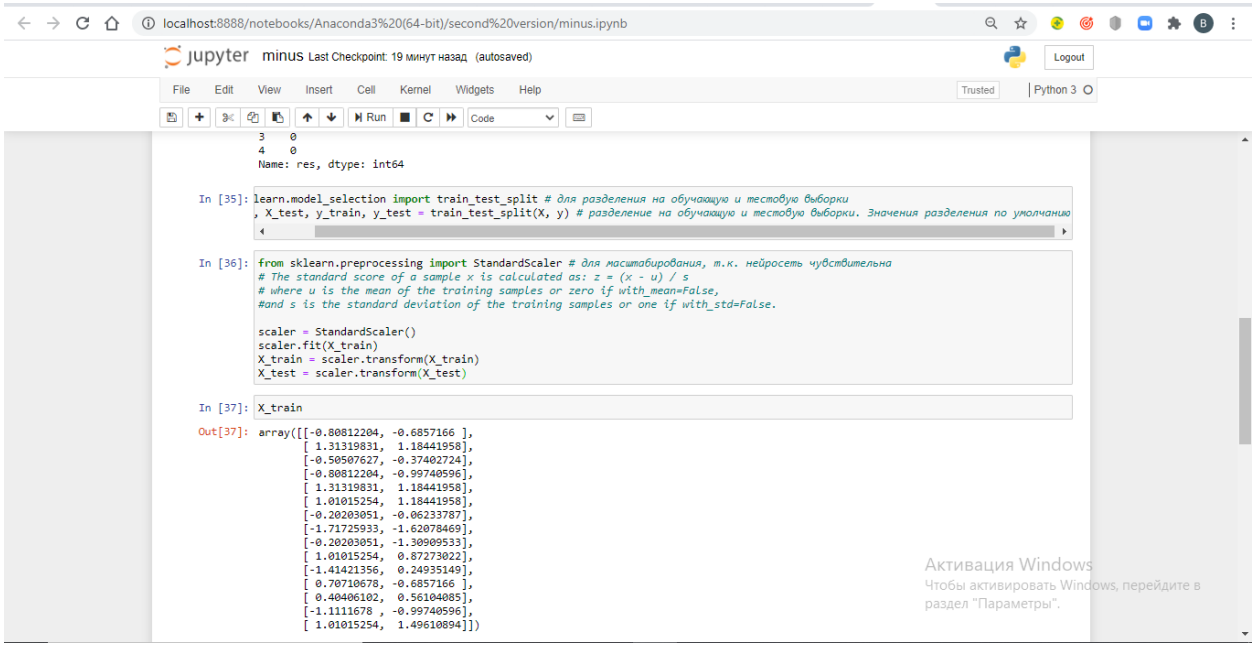

Рис.22

После загрузили модуль – обучающую(80%) и тестовую(20%) используя стандартный нормализатор, проверяем на чувствительность.

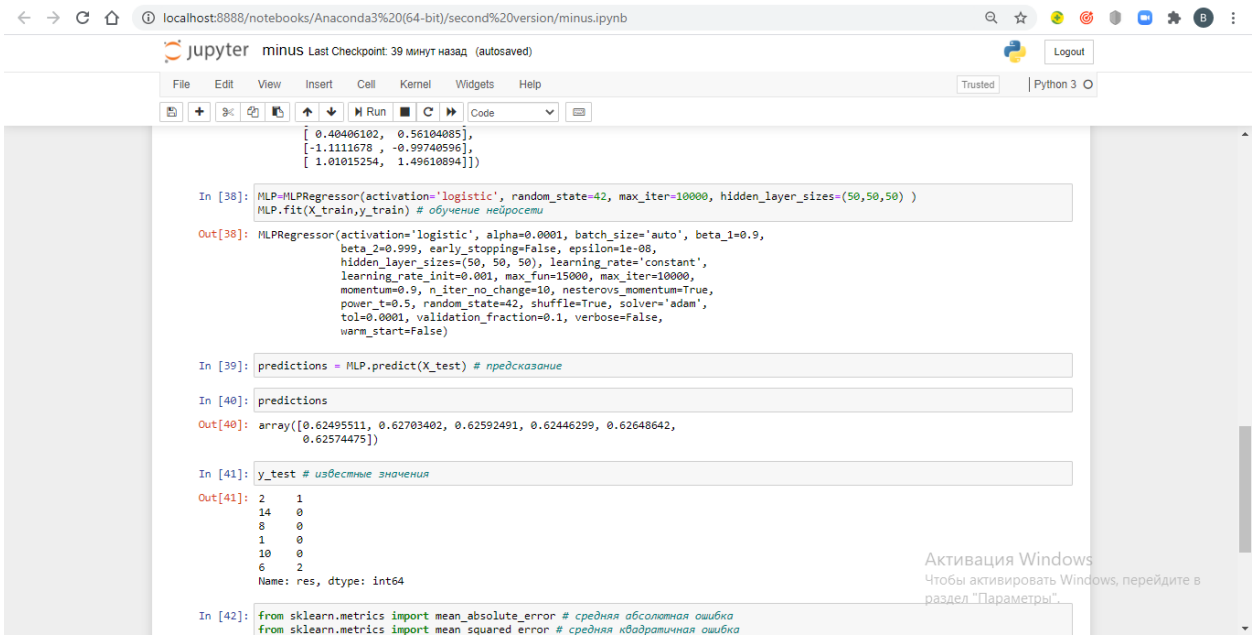

Рис.23

Создание нейронной сети. Здесь random_state=42 является случайным генератор чисел который закодирован до 42, если вы не установите семя, оно будет отличаться каждый раз. И установлен слои- 3 слоя, в каждом слое 50нейронов. Максимальное количество итераций 10000, это озночает что будет 10000 эпох для того чтобы обучать нейронную сеть по алгоритму.

После попытались предсказать на основе наших данных. На выходе получились данные Out[40]. После этого пытаемся оценить ошибки, средняя абсолютная и квадратичная ошибка. Сравниваем данные.

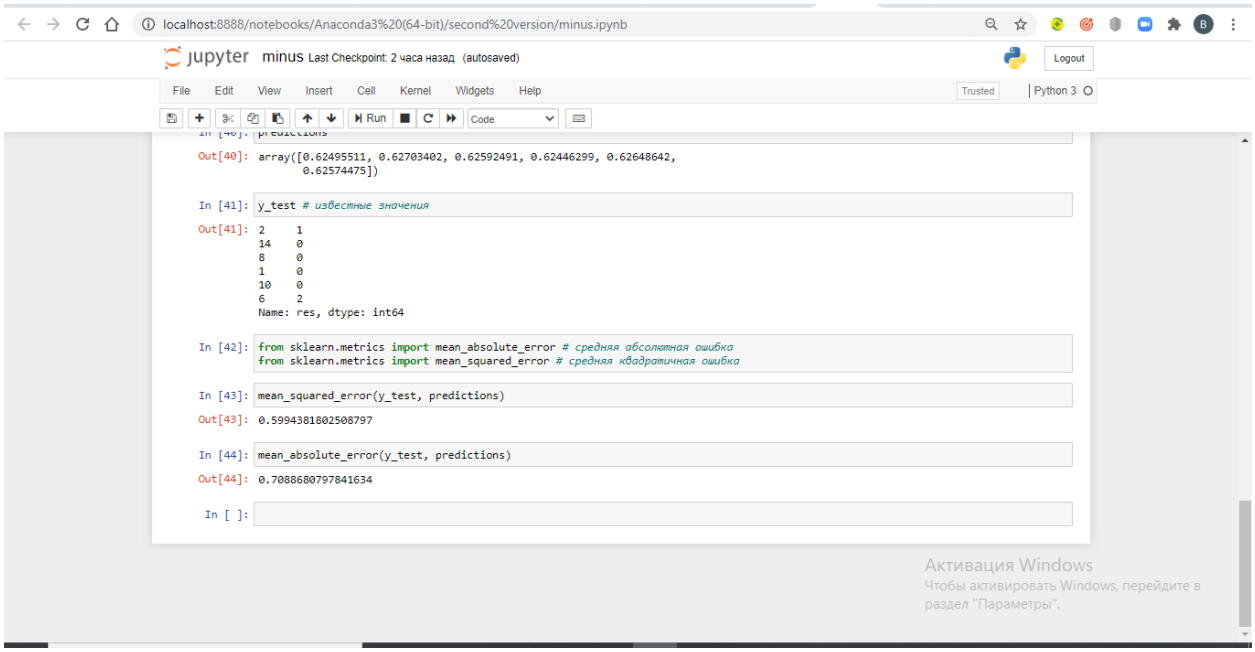

Рис.24

С помощью mean_squared_error(y_test, predictions) – провереям на среднюю абсолютную ошибку которая равна 0.5994381802508797.

Потом проверяем на среднеквадратичную ошибку - mean_absolute_error(y_test, predictions), которая равна 0.7088680797841634.

Из примера видно, что результаты меньше нуля, это означает что программа работает правильно.

Библиографический список

1. Созыкин А.В. Обзор методов обучения глубоких нейронных сетей. Институт математики и механики им. Н.Н.Красовского УрО РАН, Уральский федеральный университет, 2017.
2. Васильев А.П. , Абрамов А.Х. Искусственный интеллект на основе нейронных сетей. Якутск: Северо-Восточный федеральный университет имени М.К. Аммосова. Колледж инфраструктурных технологий.
3. Федотов В.В. Применения нейронных сетей в телекоммуникационных сетях связи // Известия вузов. Северо-Кавказский регион Естественные науки. Приложение. 2004. № 5.
4. Потапкин К. О. Искусственные нейронные сети. Нейронная сеть Хопфилда. МГУ им. Н. П. Огарёва.
5. Малыгина Ю. П. Нейронные сети: особенности, тенденции, перспективы развития // Молодой исследователь Дона. 2018. №5(14)
6. Маркова С.В., Жигалов К.Ю. Применение нейронной сети для создания системы распознавания изображений // Фундаментальные исследования. 2017. № 8-1. С. 60-64
7. Степанов П. П. Искусственные нейронные сети // Молодой ученый. 2017. № 4 (138). С. 185-187.
8. Фаустова К.И. Нейронные сети: применение сегодня и перспективы развития. //Территория науки. 2017. № 4
9. Larry M. Manevitz. Document Classification on Neural Networks Using Only Positive Examples. Department of Computer Science, University of Haifa, Haifa, Israel.
10. Boehm O., Hardoon D.R., Manevitz L.M. Classifying cognitive states of brain activity via one-class neural networks with feature selection by genetic algorithms // International Journal of Machine Learning and Cybernetics. 2 (3). С. 125
11. Schmidhuber J. Deep learning in neural networks: An overview //Neural networks. 2015. Т. 61. С. 85-117.
12. https://www.sas.com/ru_ru/insights/articles/analytics/what-is-artificial-intelligence.html
13. http://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D1%81%D0%B5%D1%82%D0%B8_(%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5_%D1%81%D0%B5%D1%82%D0%B8)
14. http://statsoft.ru/home/textbook/modules/stneunet.html
15. https://tproger.ru/translations/scikit-learn-in-python/