**Лабораторная работа № 3**

«Многослойные сети. Алгоритм обратного распространения ошибки »

Целью работы является исследование свойств многослойной нейронной сети прямого распространения и алгоритмов ее обучения, применение сети в задачах классификации и аппроксимации функции.

Ход работы.

**Задание 1**: использовать многослойную нейронную сеть для классификации точек в случае, когда классы не являются линейно разделимыми.

**Задание 2**: использовать многослойную нейронную сеть для аппроксимации функции. Произвести обучение с помощью одного из методов первого порядка.

**Задание 3**: использовать многослойную нейронную сеть для аппроксимации функции. Произвести обучение с помощью одного из методов второго порядка.

Описание алгоритма.

В качестве метода поиска экстремума функции многих переменных первого порядка, был выбран метод **градиентного спуска**.

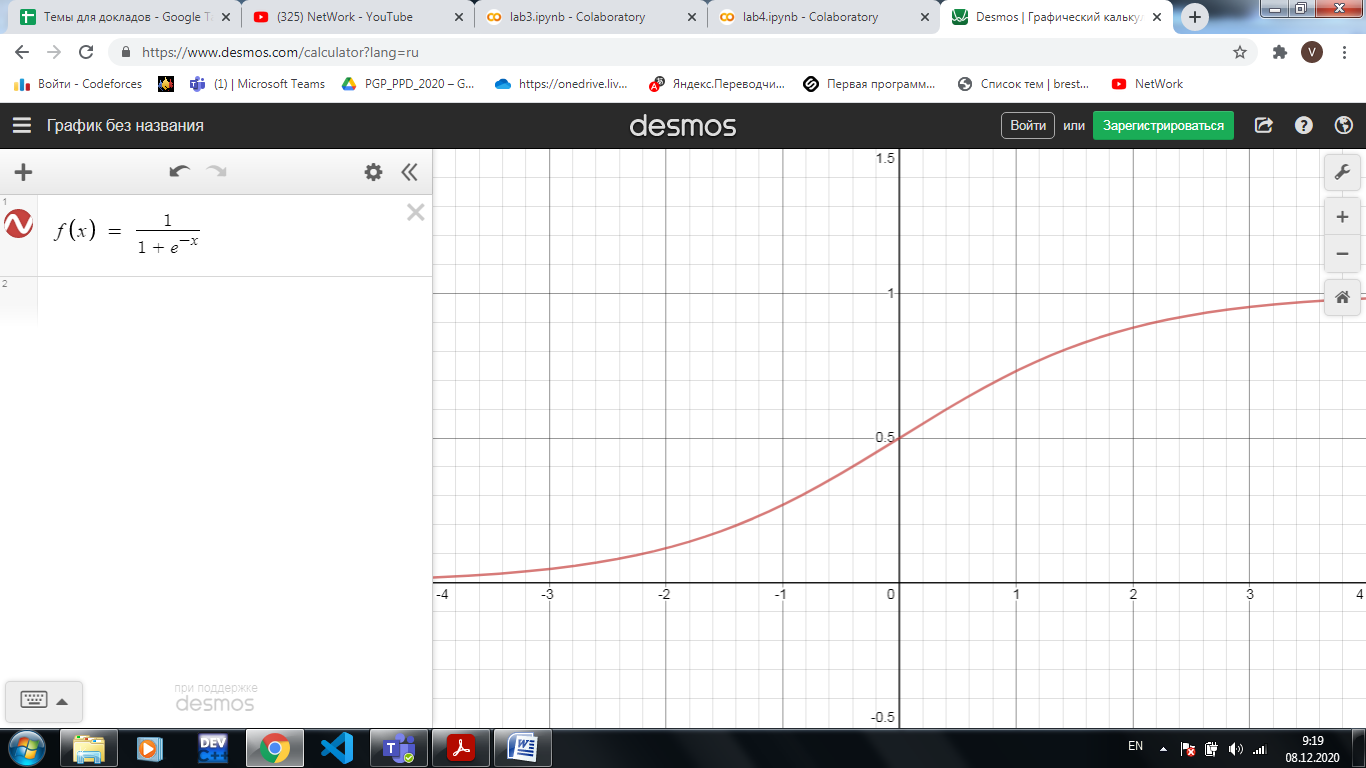
В качестве метода поиска экстремума функции многих переменных второго порядка, был выбран метод **Левенберга-Марквардта**.

***Цепное правило дифференцирования***

«*Производная сложной функции равна произведению ее производной по промежуточному аргументу на производную этого аргумента по независимой переменной*»

Используя цепное правило дифференцирования, можно осуществить вычисление градиента функции ошибки. Именно поэтому очень важно, чтобы активационная функция была дифференцируема и считалась легко.

Например, функция **Sigmoid**.



) = , тогда

=

**Алгоритм обратного распространения ошибки.**

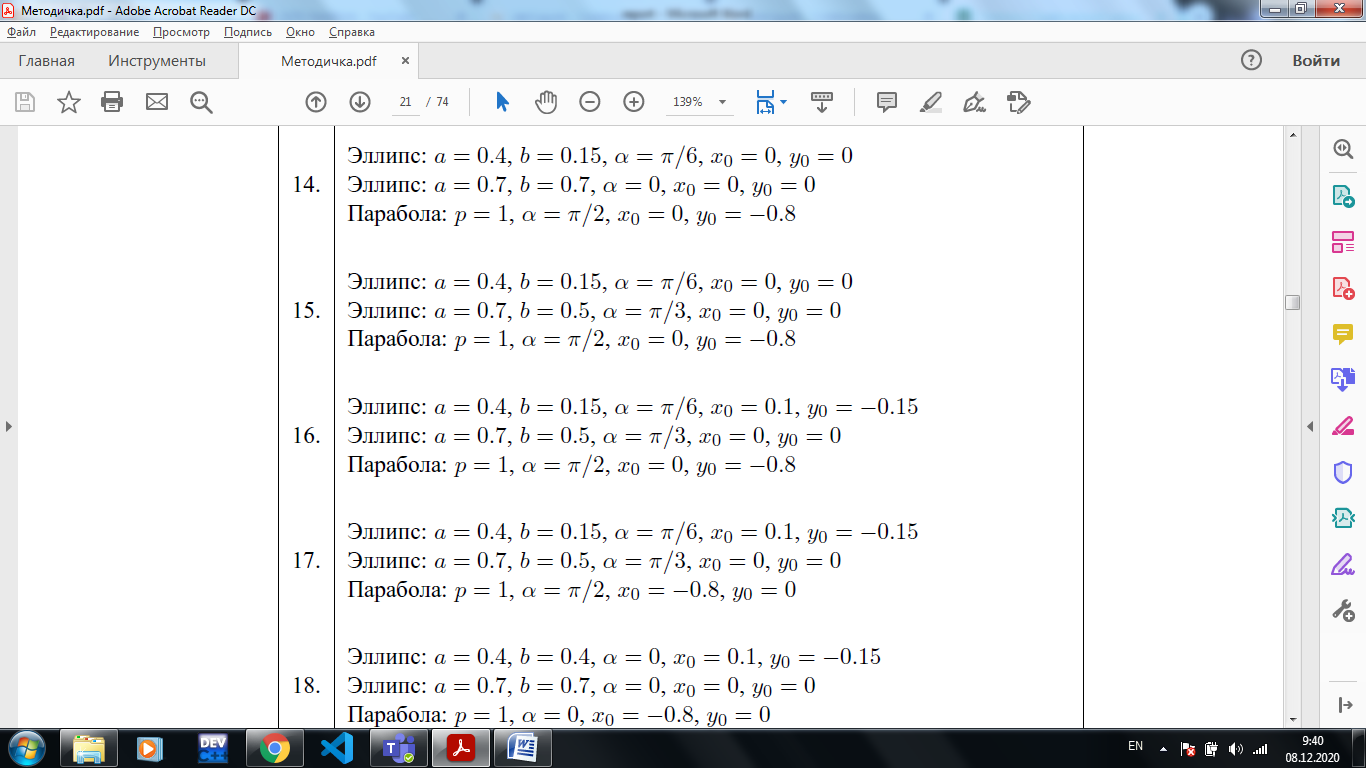
1. Фиксируем веса сети W. Делаем прямой проход алгоритма.
2. Берем пример из обучающего набора и подаём вектор на вход сети.
3. Вычисляем выход сети Y(; W).
4. На основе желаемого выхода для примера и реально полученного выхода Y(; W) вычисляется ошибка сети для данного примера: = -
5. Шаги 2, 3, 4 повторяются для всех примеров из обучающей выборки при фиксированном значении W.
6. Вычисляем суммарную ошибку сети для данной r–й эпохи: =.
7. Веса сети W корректируются так, чтобы уменьшить ошибку .
8. Повторяем 1-7 до тех пор, пока не достигнем максимального числа эпох или пока не будет выполнено условие <=

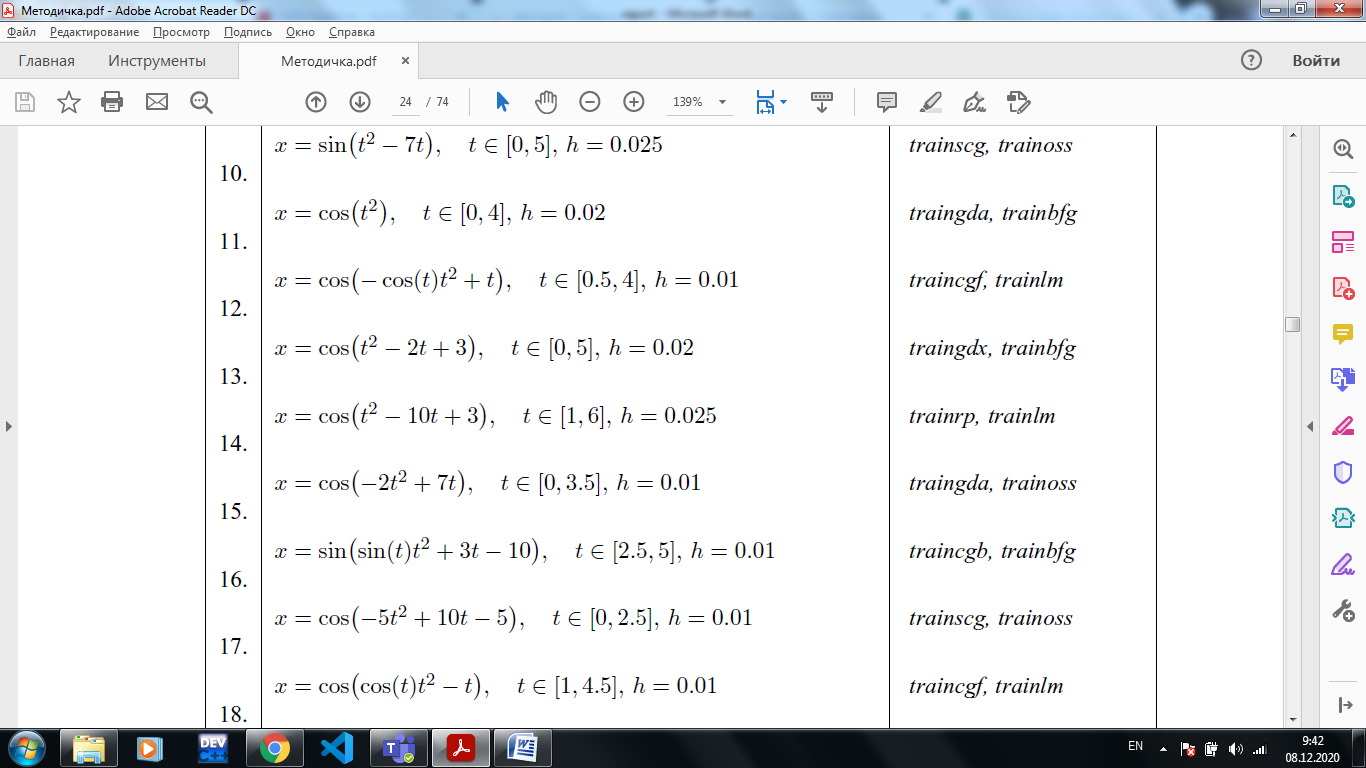
Ошибка сети связана только с выходами сети и весами связей выходного и предшествующего ему скрытого слоя. Как определить ошибку на выходах нейронов скрытых слоев по ошибке в последующем слое?

Ошибку для последнего скрытого слоя можно определить по ошибке выходного слоя. Ошибку для предпоследнего скрытого слоя — по ошибке последнего скрытого слоя и т. д. Ошибка всей сети «распространяется назад», от выходного слоя сети к ее входному слою. Переход «назад» от слоя к слою в алгоритме **Back Propagation** сопровождается не только вычислением ошибки на соответствующем слое сети, но и пересчетом весов данного слоя, уменьшающим его ошибку на последующих эпохах. Веса на всех слоях изменяются одновременно.

Входные данные и результаты.

**Вариант № 15.**

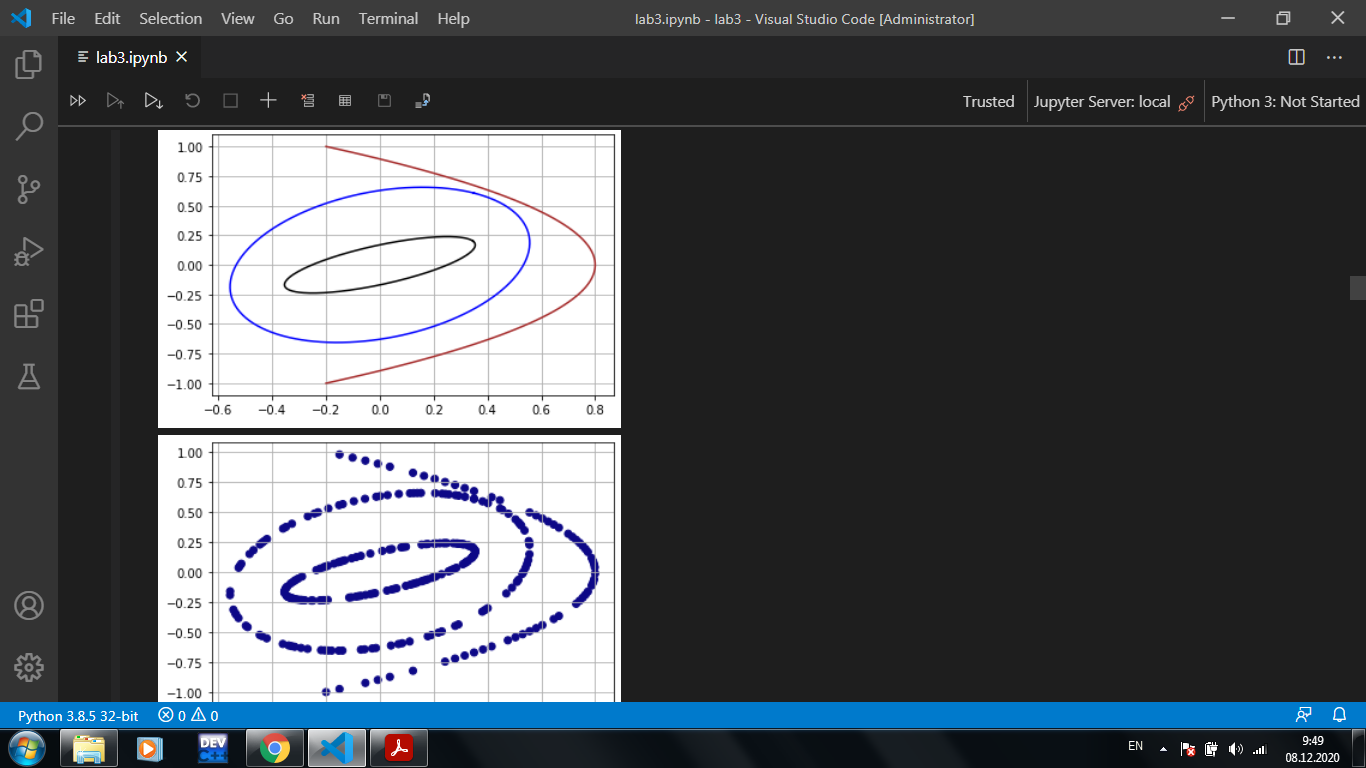


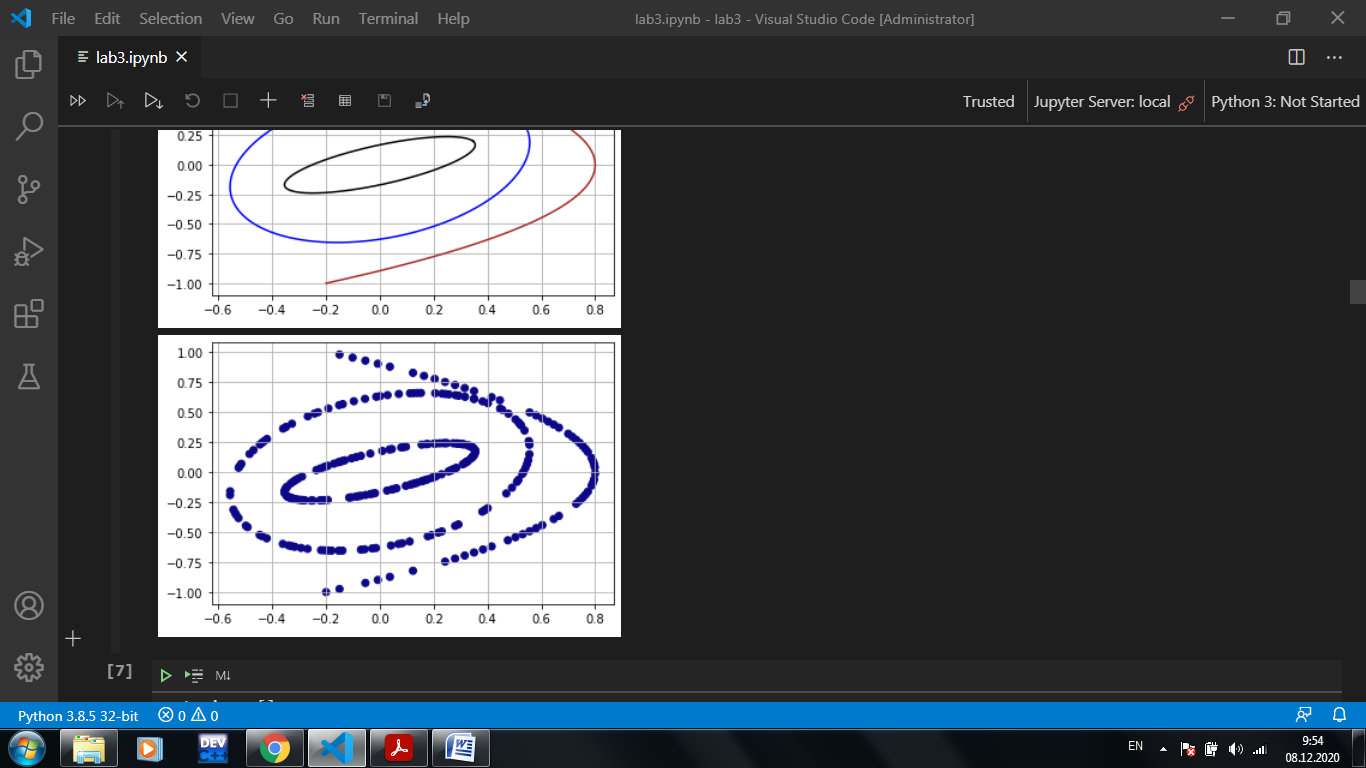


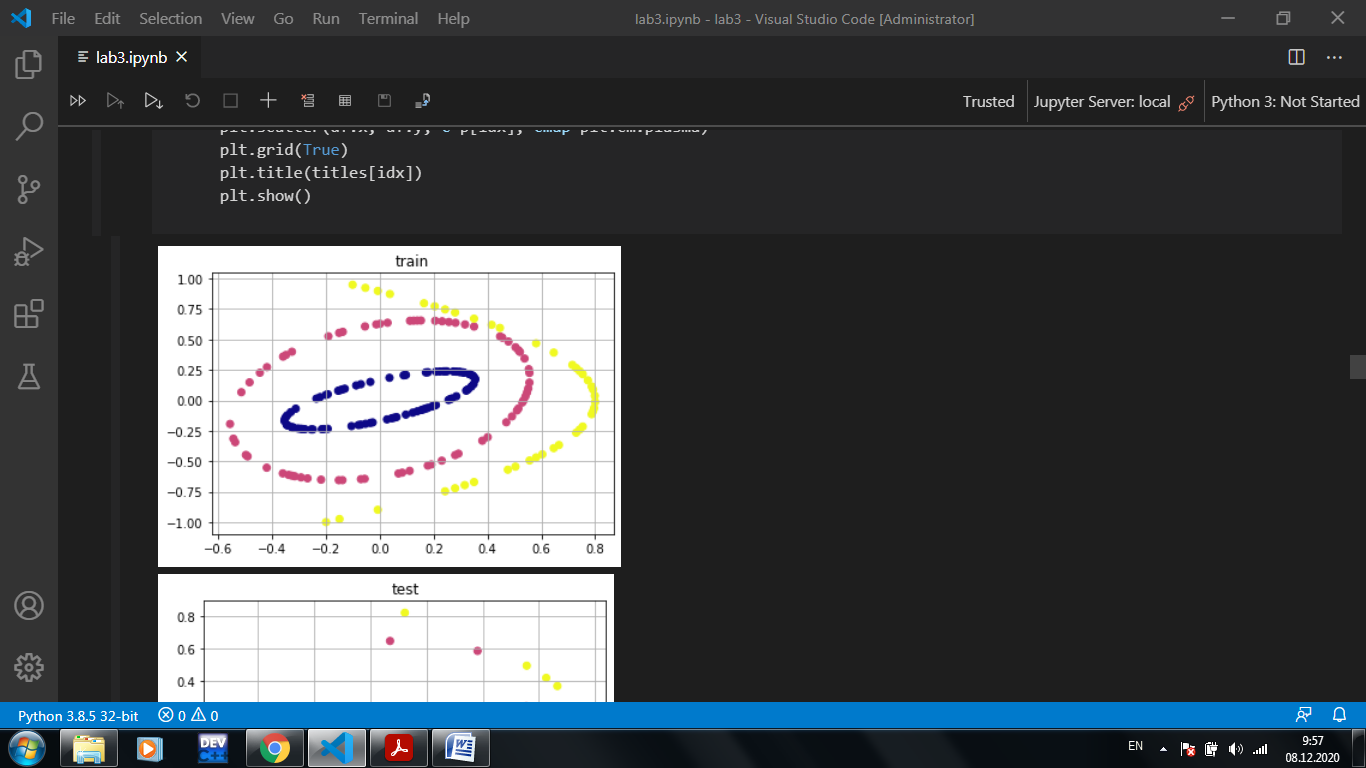
**Traingda** (градиент с адаптивным шагом)

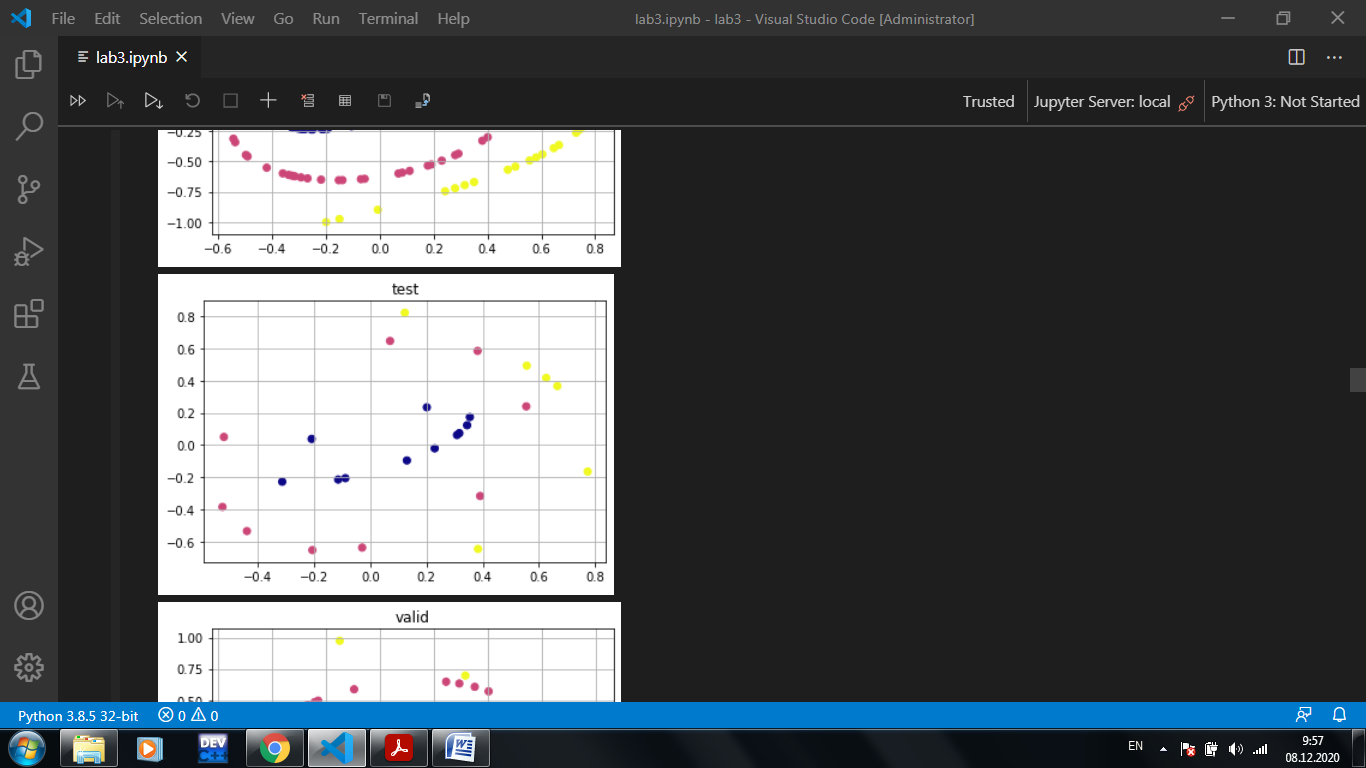
**Trainoss** (одношаговый метод секущих)

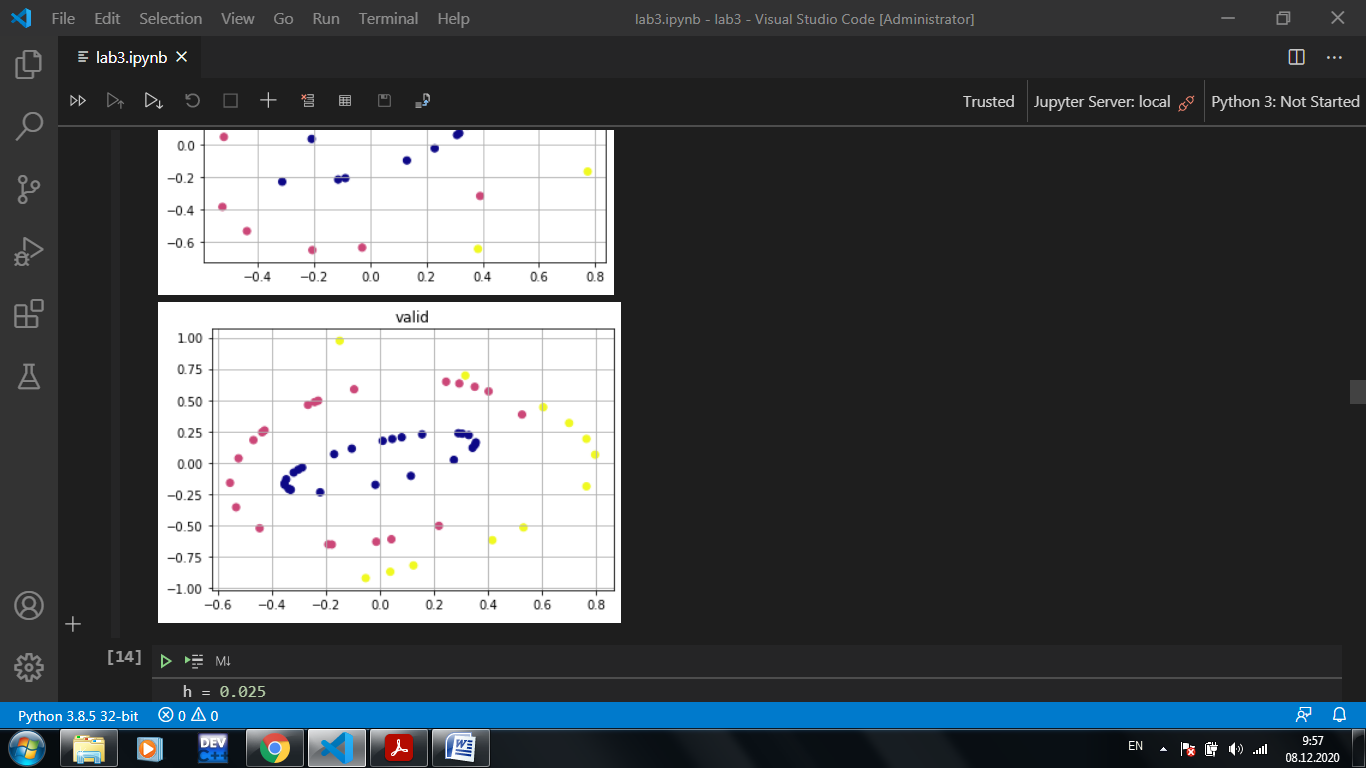
Классификация



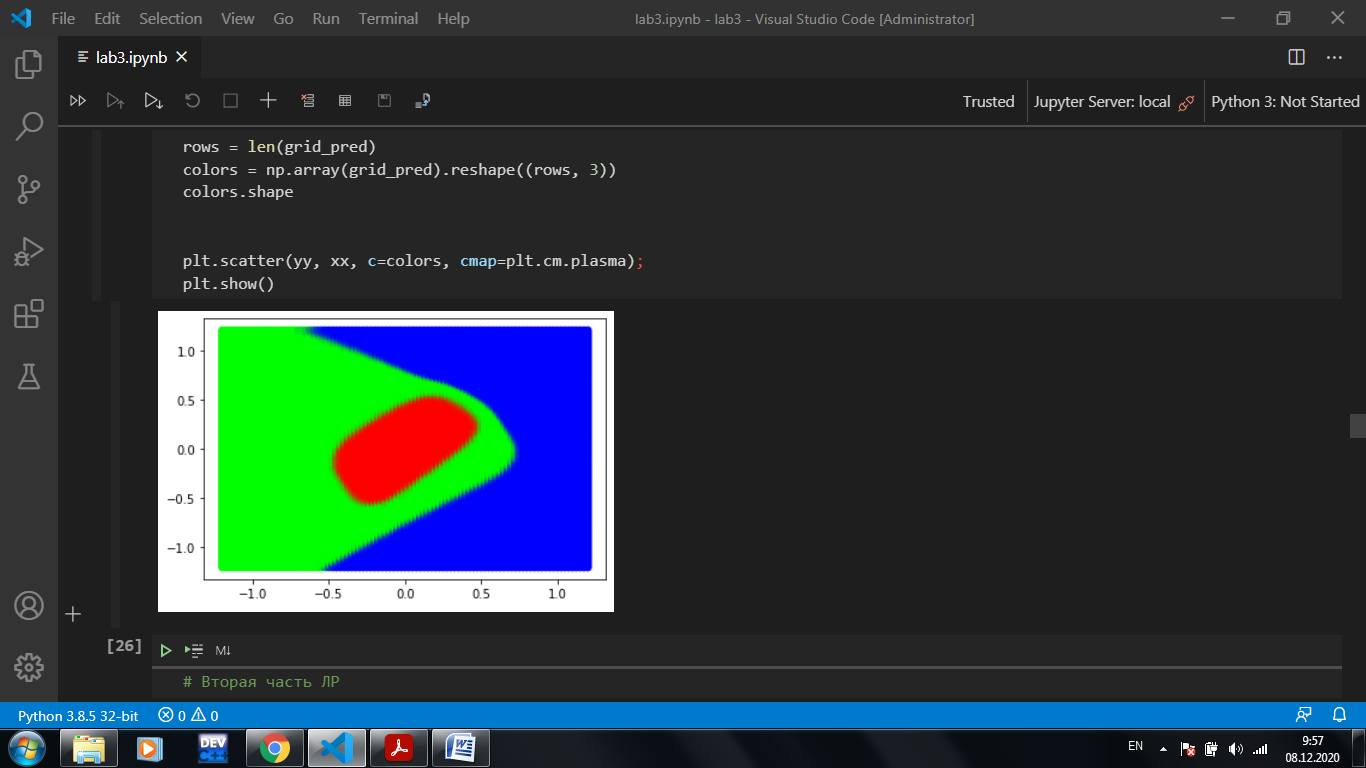




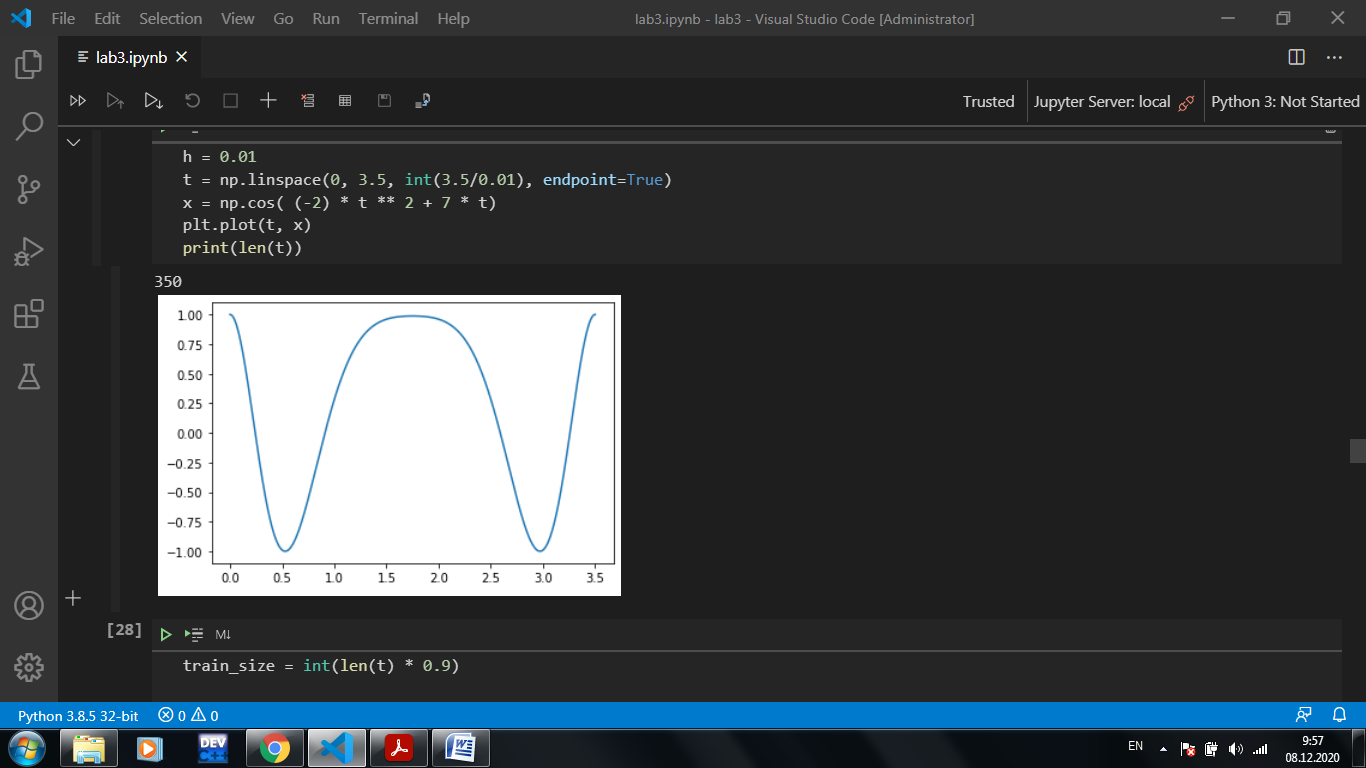


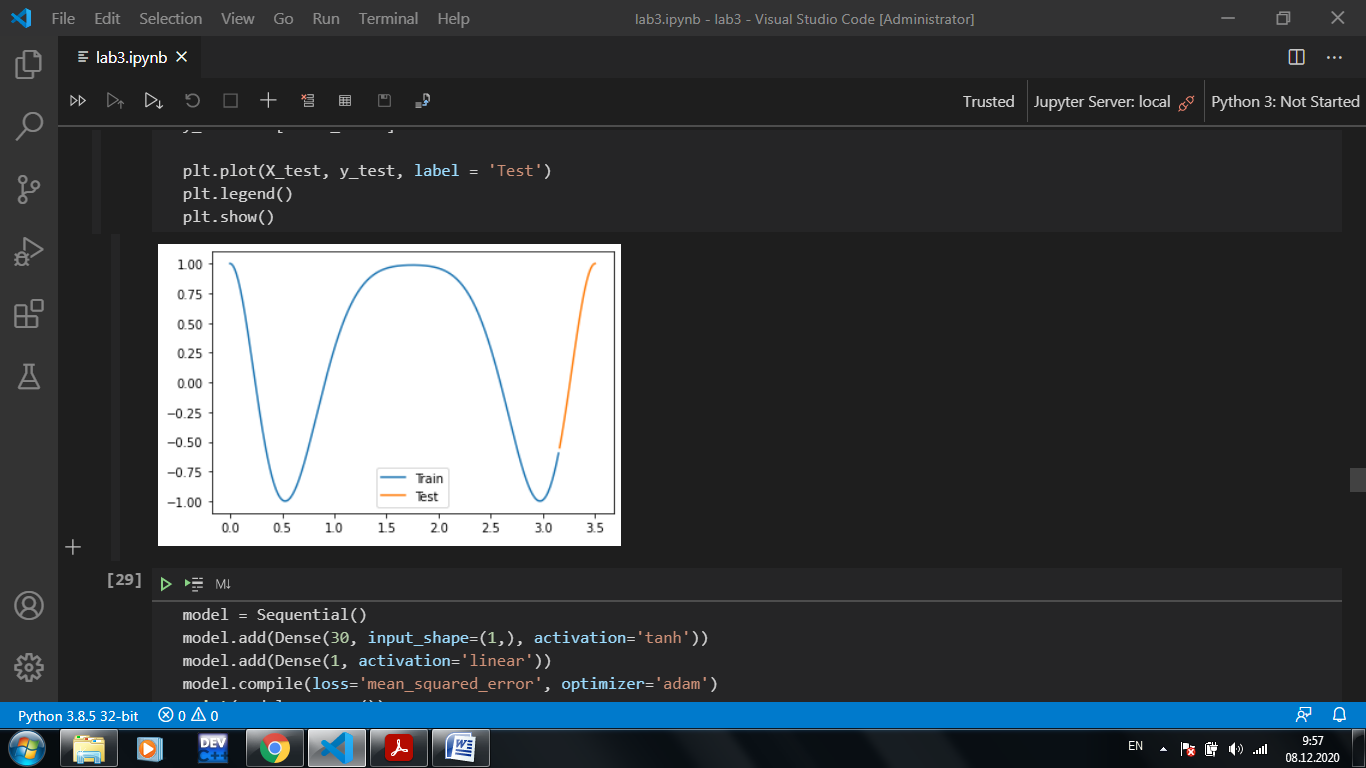


Результат

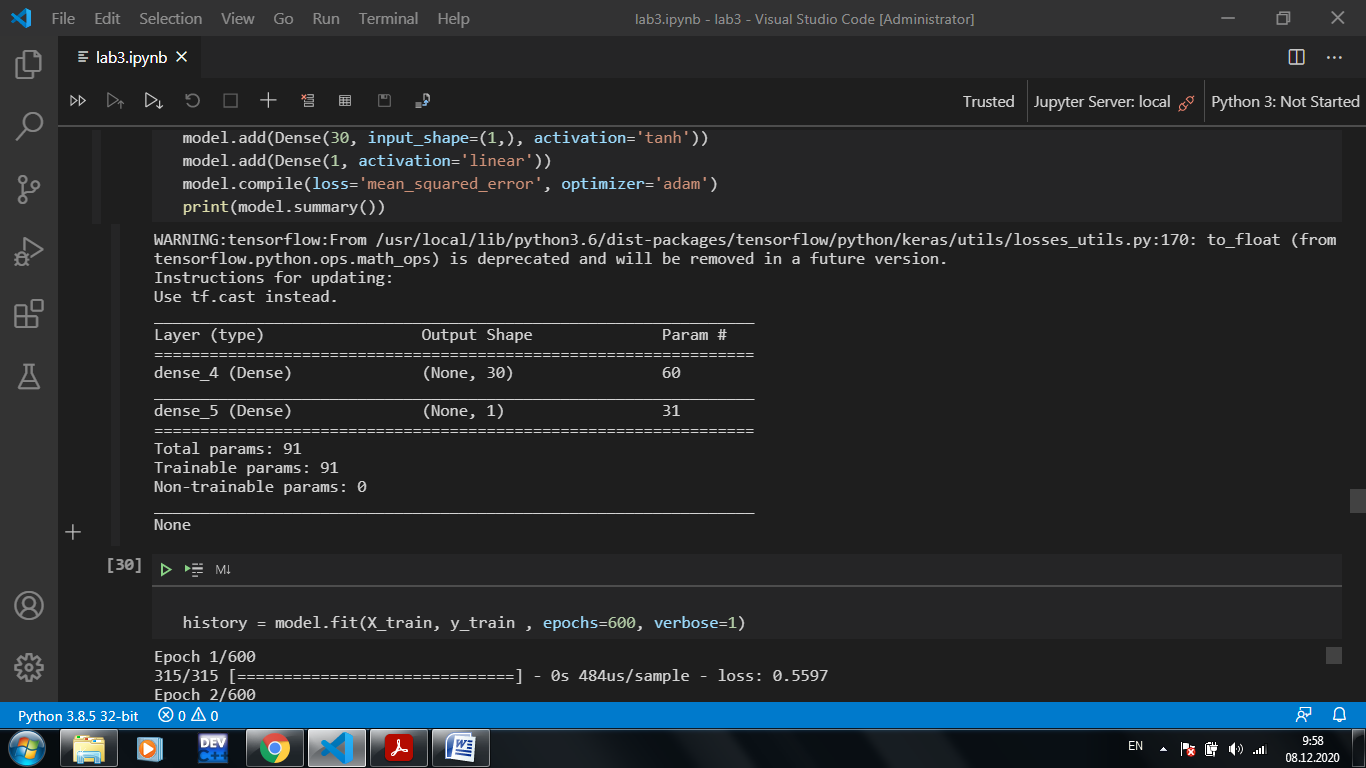


Аппроксимация

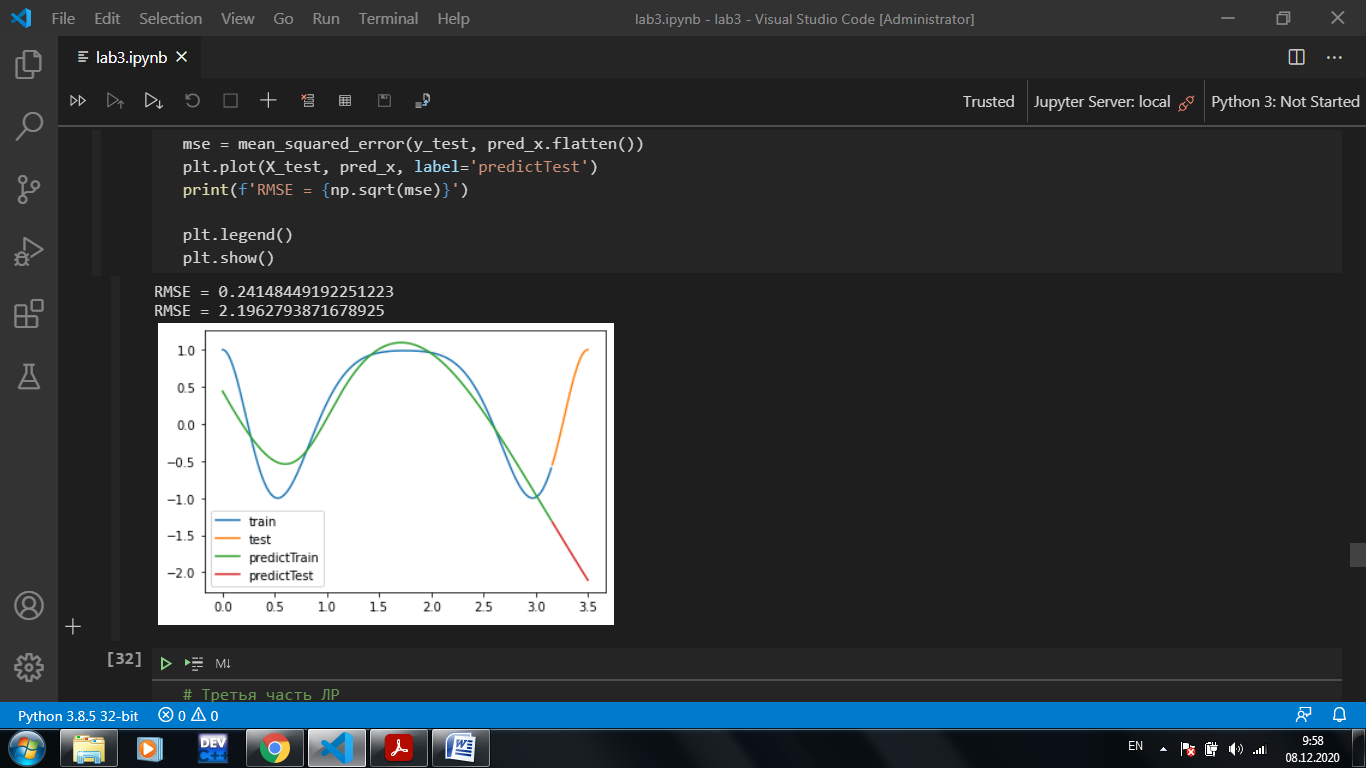


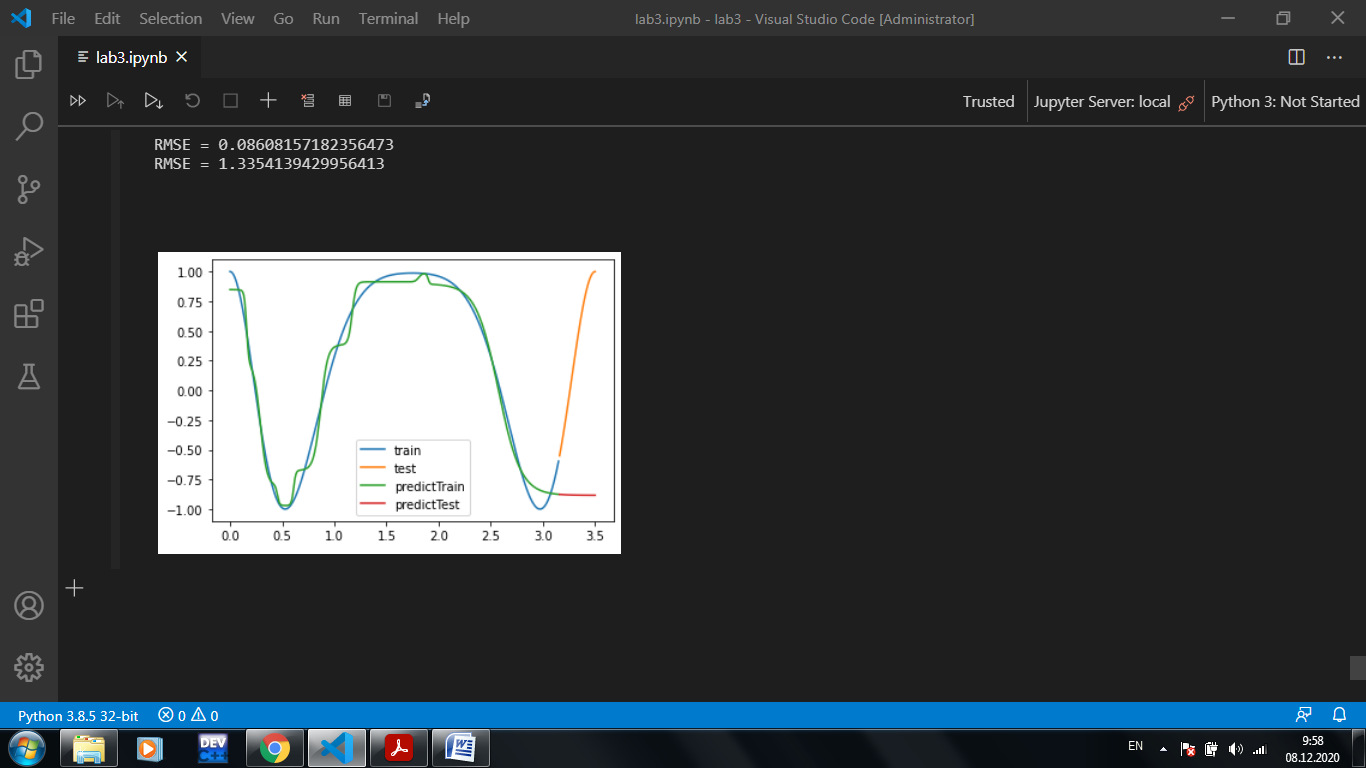


Структура сети



Результат





Исходный код.

Прикреплен в файле lab3.ipynb.

Выводы:

Метод обратного распространения ошибки является мощным инструментом обучения многослойных сетей прямого распространения, но, несмотря на многочисленные успешные применения обратного распространения ошибки, оно не является универсальным решением. Неопределённо долгий процесс обучения - это большой минус. В сложных задачах для обучения сети могут потребоваться дни и не факт, что она обучится с желаемым качеством.

Методы нахождения градиента функции многих переменных можно разделить на две группы:

**Методы первого порядка**

* Они просты как для понимания, так и для программирования
* В каждой эпохе применение этих методов быстрее с точки зрения производительности
* Может очень долго сходиться в конце оптимизации

**Методы второго порядка**

* Высокая устойчивость
* Большие шансы найти глобальный экстремум
* Очень высокая скорость сходимости
* Налагают ограничения на вид целевой функции
* Сложности в реализации.

По результатам лабораторной работы видно, что сеть показывает плохие результаты на том отрезке, где не было обучающих данных. Это естественно, так как нельзя давать сети на вход данные из области, не покрытой обучающей выборкой. Результат будет похож на «угадывание» учеником невыученной домашней работы.