Лабораторная работа 2.

Вариант 23

Москалев Владислав Александрович.

**Github репозиторий с лабораторными -** <https://github.com/worlddeleteRin/neural_sets>

Функциональная связь - 3 + 5x + 2x^2

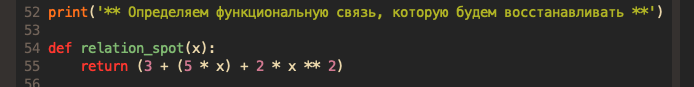
Интервал построения - [-3;0]

Мультипликативная, нормальное, M(X)=1.2 sko = 0.7

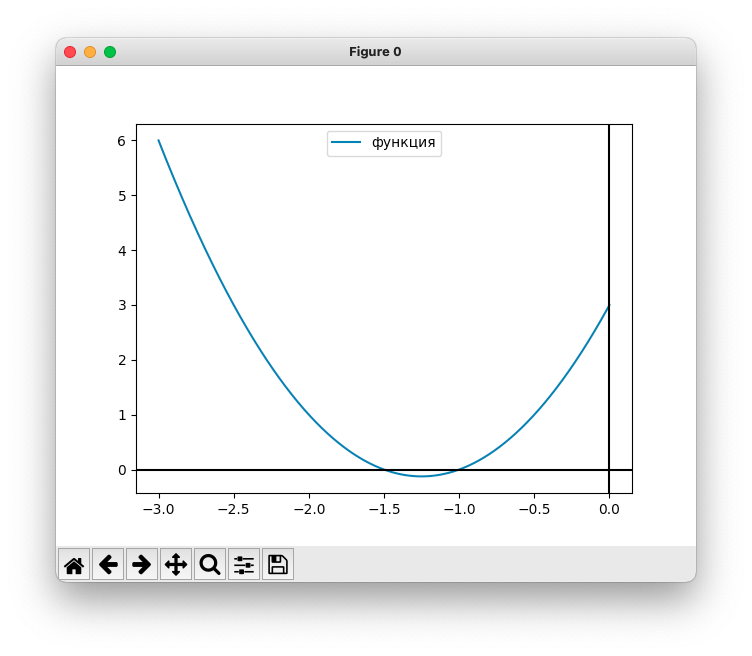
В первую очередь необходимо создать выборку, тоесть начать с разбития интервалов по входной переменной. В результате будет получен следующий массив:



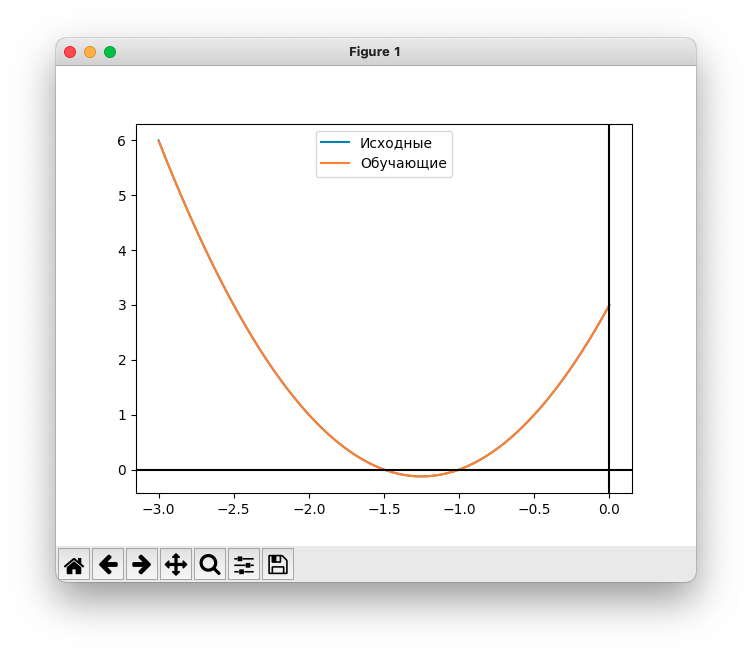
Далее, в зависимости от функциональной связи, которую нужно восстанавливать, создадим соответствующий метод для ее восстановления. В моем случае код метода будет выглядеть следующим образом:



Строим график исходной функциональной зависимости. В моем случае от будет выглядеть следующим образом:

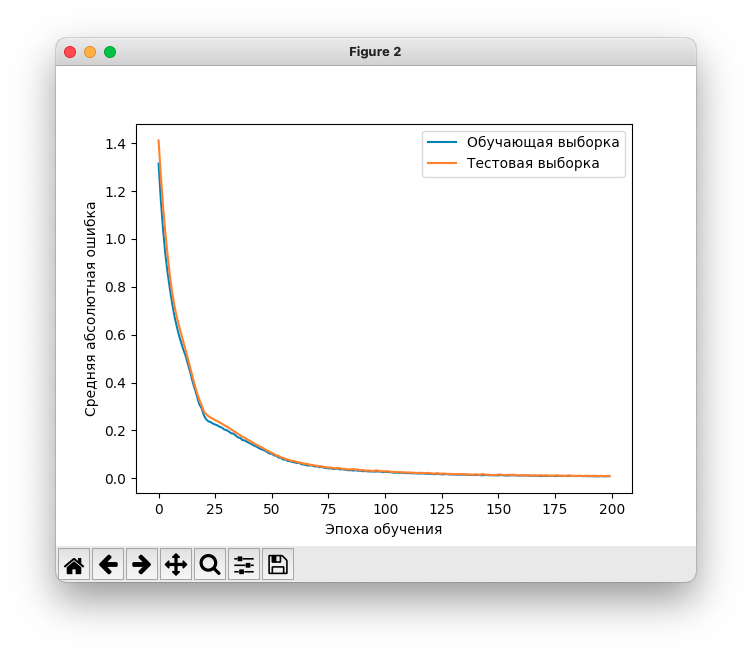


Разбиваем выборку на обучающую и тестовую, для тестовой я использовал 300 записей. Визуализируем обучающую выборку вместе с исходными значениями:

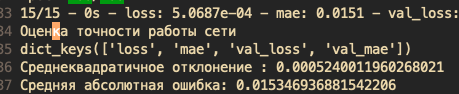


Четко наблюдаем необходимый интервал строго в пределах от -3 до 0. Обучающие данные совпадают с исходными, можно идти дальше.

Нормализуем данные, создаем нейронную сеть и обучаем ее. Проверяем точность работы сети:

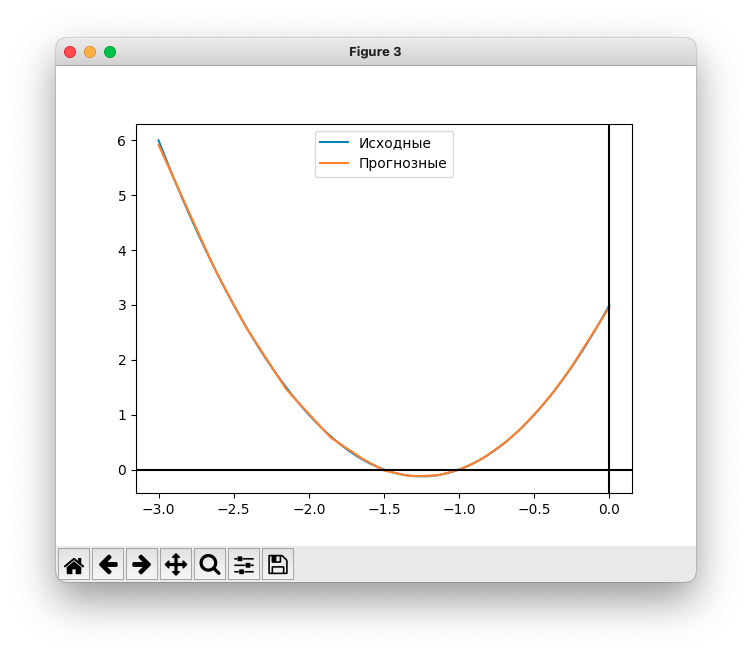


Значения среднеквадратичного отклонения и абсолютной ошибки:



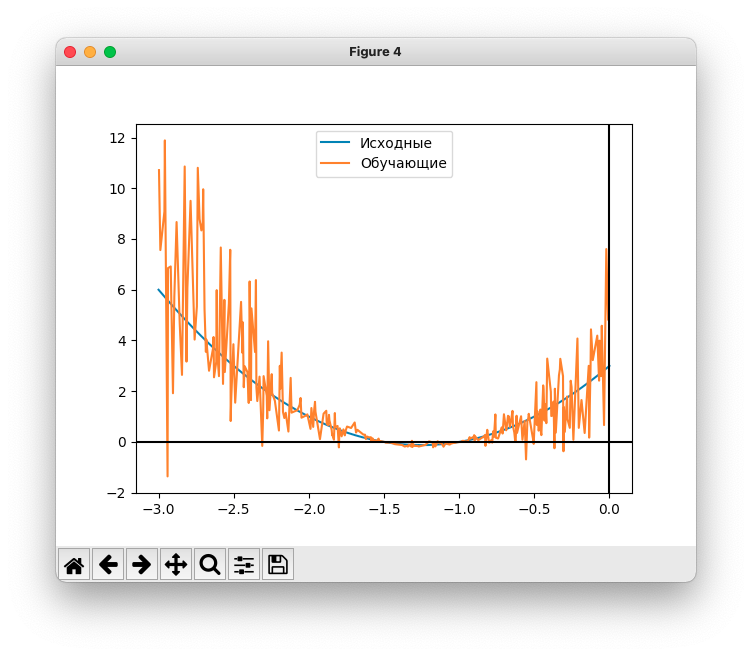
Можно сделать вывод, что на данный момент модель предсказывает с точностью до 99%

Произведем визуализацию работы сети, сопоставив прогнозные значения с исходными:

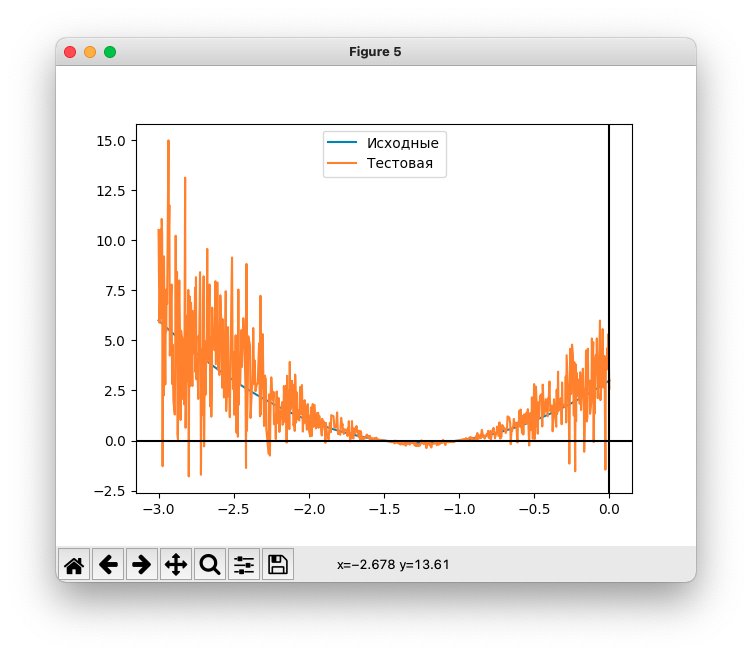


Данный график подтверждает, что обученная модель работает практически безошибочно. Теперь оценим влияние стохастической компоненты на обучаемость нейронной сети. Так как по условию задания стохастическая компонента мультипликативная и распределение нормальное, то, для получения вариативных значений мы будем использовать метод normalvariate пакета random.

Итак, накладываем стохастическую компоненту на обучающую выборку. Для начала построим диаграмму для визуализации полученной обучающей выборки:

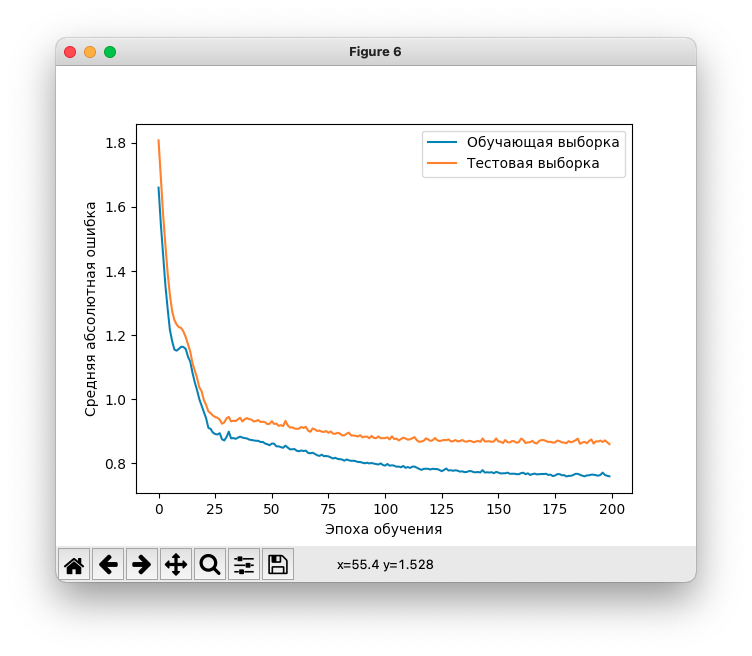
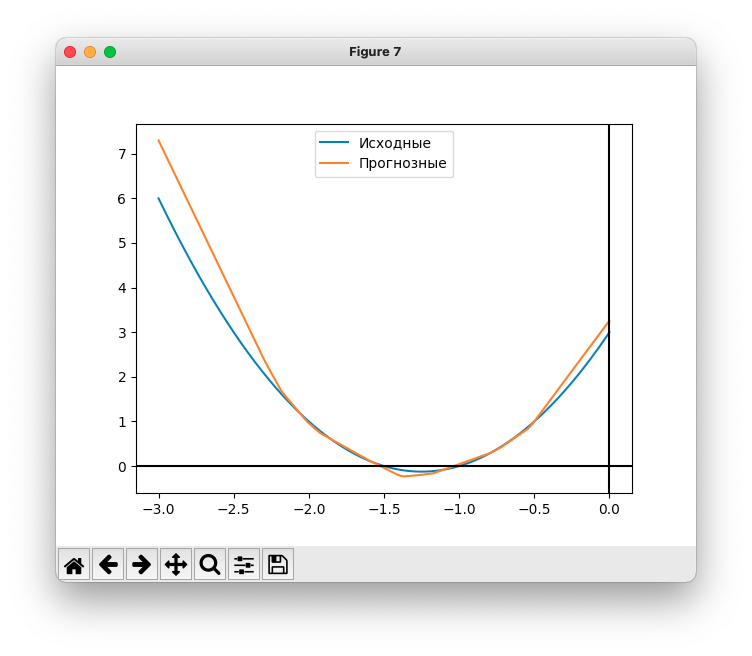


Замечаем довольно сильную амплитуду колебаний значений обучающей выборки от исходной, из-за того, что по условию задания среднеквадратичное отклонение = 0.7. Чем ближе было бы данное значение к 0, тем меньше была бы амплитуда колебаний обучающей выборки от исходных данных. Построим график для тестовой выборки:



Также наблюдаем сильные отклонения значения из-за высокого коэффициента среднеквадратичного отклонения.

Нормализуем обучающую и тестовую выборки, заново обучаем нейронную сеть и строим график средней абсолютной ошибки по эпохам обучения:

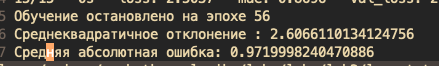


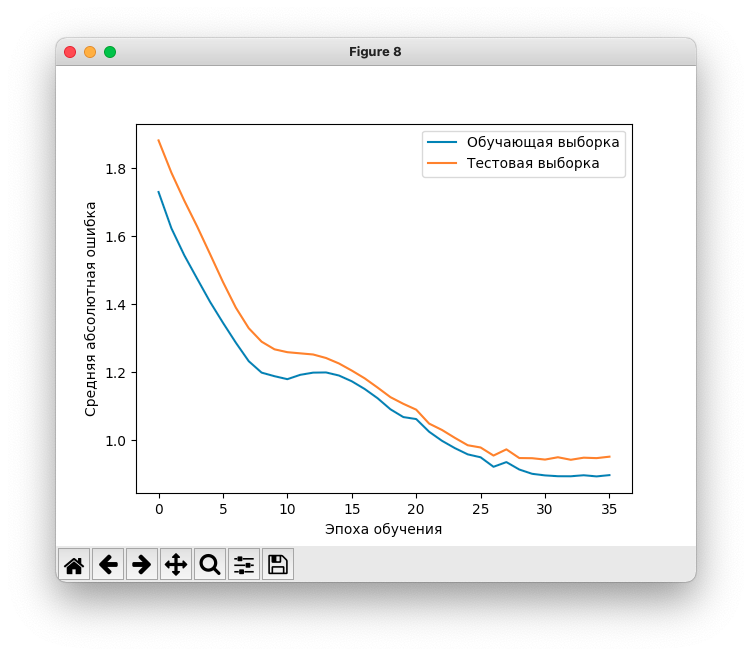
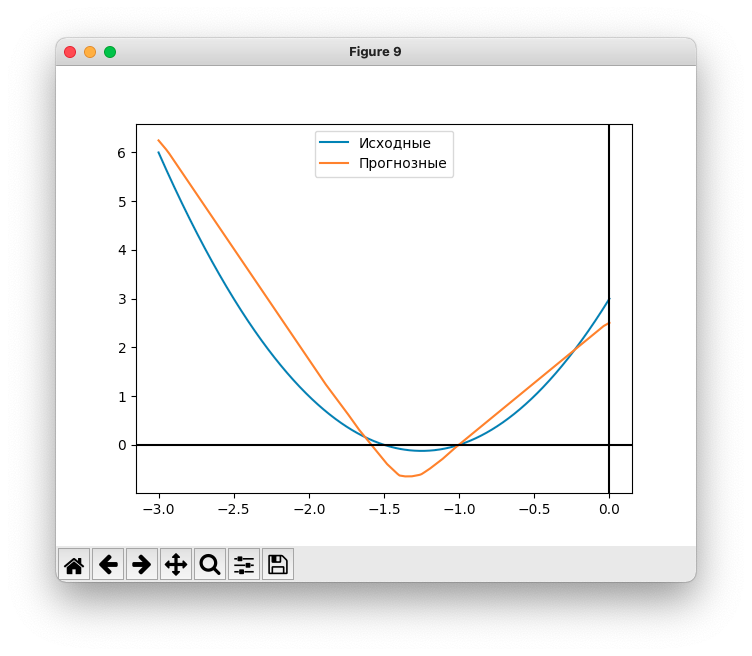
Как можно уже видеть по графику, средняя абсолютная ошибка остается высокой для тестовой и обучающей выборки на протяжении всего обучения, и, не опустилась ниже значения 0.75. Посмотрим на точные цифры:



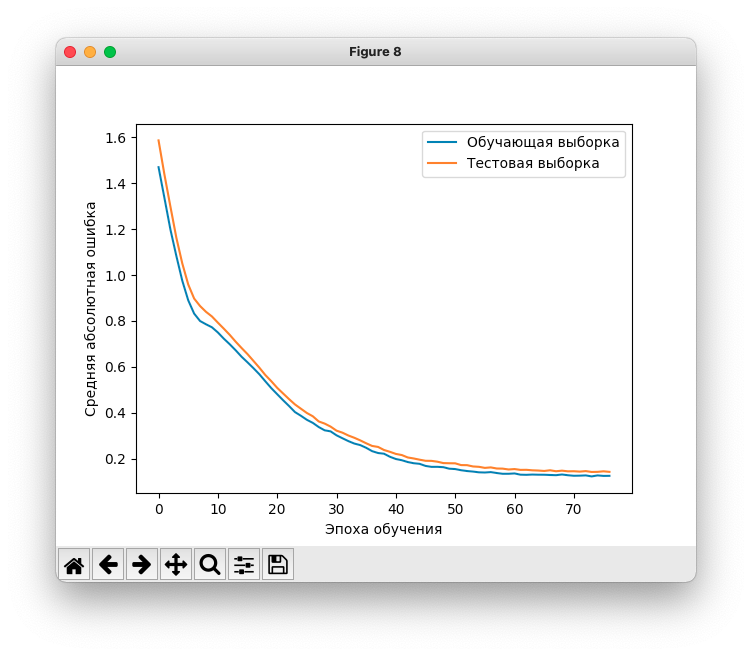
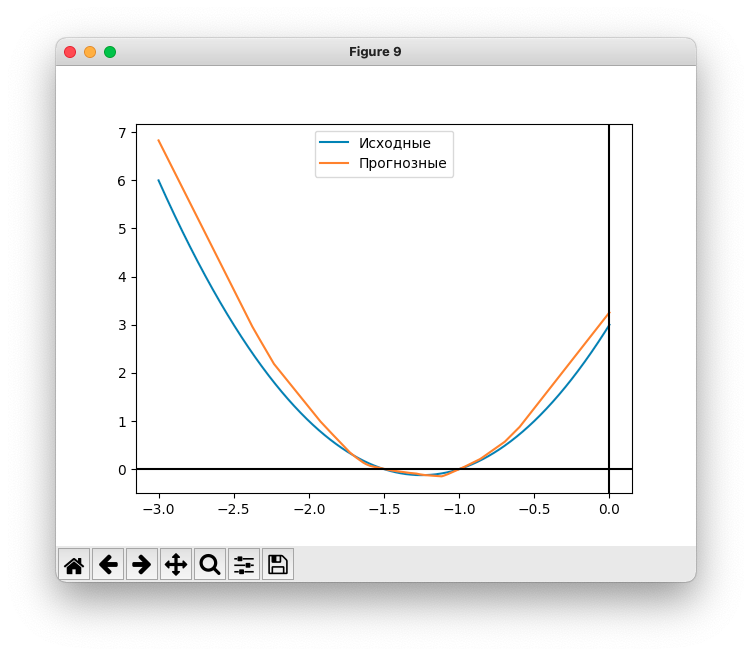
Можно сделать вывод, что модель правильно предсказывает значение только в 17% случаев, что является очень плохим результатом для ее использования.

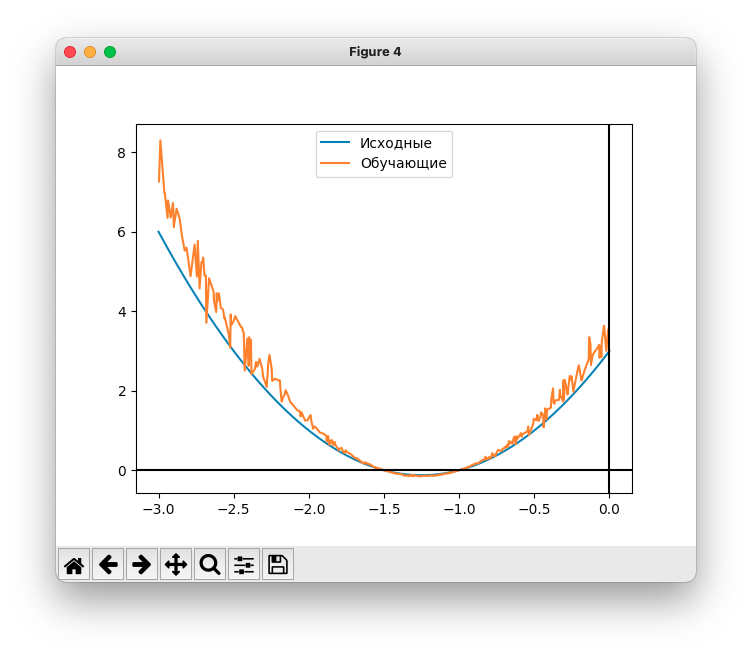
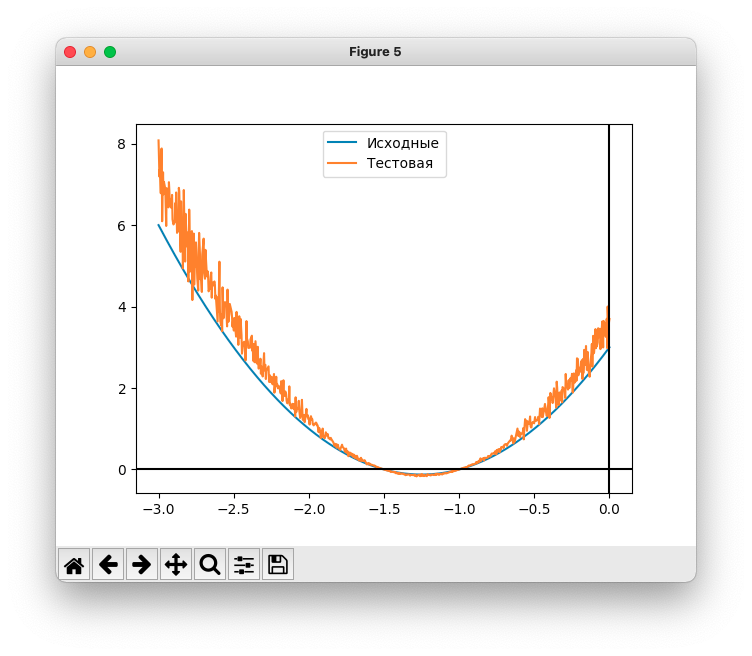
Используем EarlyStopping Callback, смотрим на результаты:

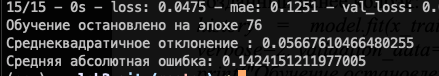


Итак, получаем практически такие же результаты, обучение было остановлено на 56 эпохе, модель все так же ошибается в 97% случаев. Как уже было сказано, такие результаты было получены в связи с тем, что по условию задания среднеквадратичное отклонение равно 0.7, что является большим значением. Попробуем поэкспериментировать, и, изменить это значение (понизить) до 0.1. Смотрим на результаты:







При новом, экспериментальном значении среднеквадратического отклонения модель верно предсказывает значения в 86% случаев, колебания отклонений Обучающих и Тестовых данных наряду с Исходными куда меньше. Уже намного лучше.