Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Вятский государственный университет»

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Исследование многослойного персептрона с обучением по методу с обратным распространением ошибки

Отчёт

### Лабораторная работа № 2 по дисциплине

«Системы обработки знаний»

Выполнил студент группы ИВТб-41\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Седов М.Д./

## Проверил доцент кафедры ЭВМ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Ростовцев В.С./

Киров 2021

**Цель:**

Ознакомиться с основными командами создания, обучения и применения многослойных нейронных сетей в Neural Network Toolbox для аппроксимации заданной функции.

**Задание:**

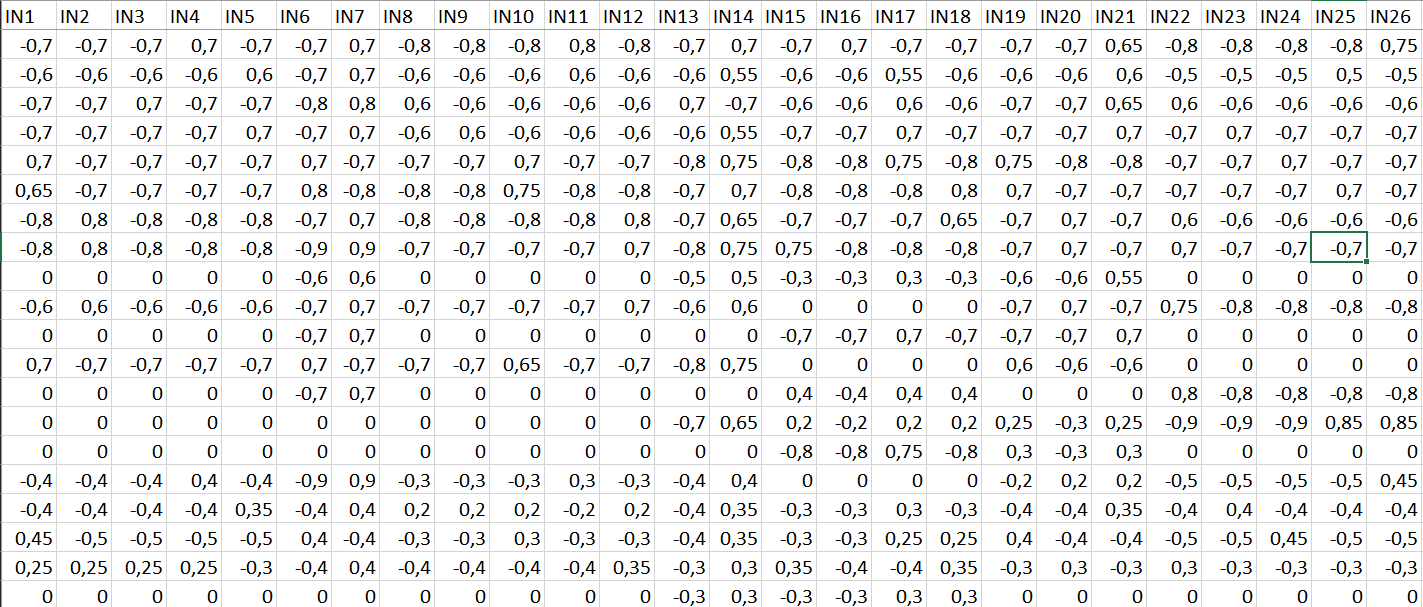
1. Ознакомиться с демонстрационными программами персептрона. Создать в рабочем пространстве MATLAB каскадную сеть с прямой передачей данных и исследовать ее структуру с оценкой качества обучения mse.

2. Обучить сеть в заданной число циклов (до 100 циклов).

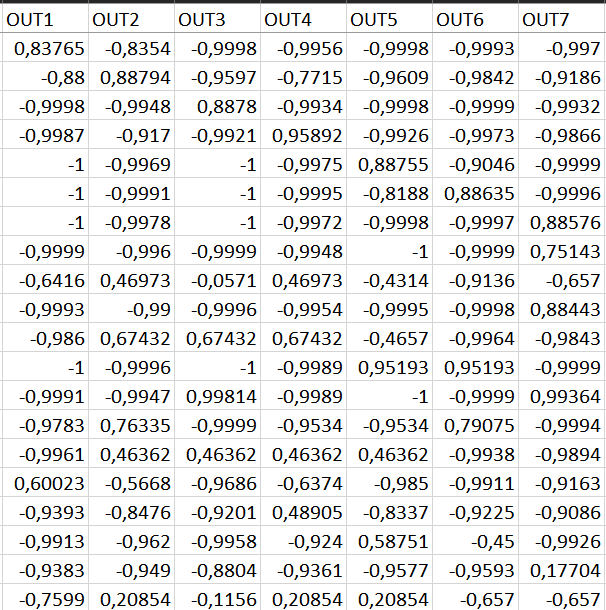
3. Построить графики зависимостей обучения нейросетей от количества циклов.

**Данные для обучения:**

Входные:



Выходные:



**Обучение НС:**

* traingd – функция обучения НС с использованием алгоритма градиентного спуска GD.
* trainlm – функция обучения НС с использованием алгоритма Левенберга-Марквардта LM.
* trainbr – функция обучения НС с использованием алгоритма Левенберга-Марквардта, дополненная регуляцией по Байесу BR.

Обучение проводилось в 100 итераций.

Аппроксимация функции будет выполняться на различных архитектурах нейронной сети (двухслойной и трехслойной) с оценкой качества MSE.

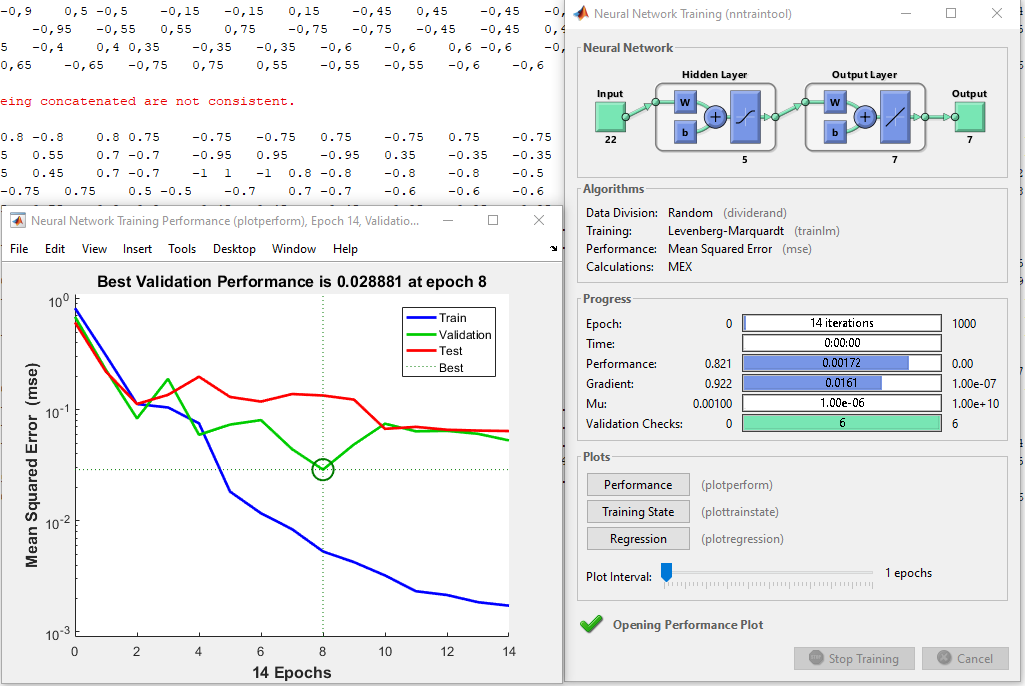
Для аппроксимации будут подобраны различное количество нейронов:

На выходном слое 7 нейронов и линейная функция активации, на скрытых слоях подобраны 5, 12, 20 нейронов и сигмоидальная функция активации.

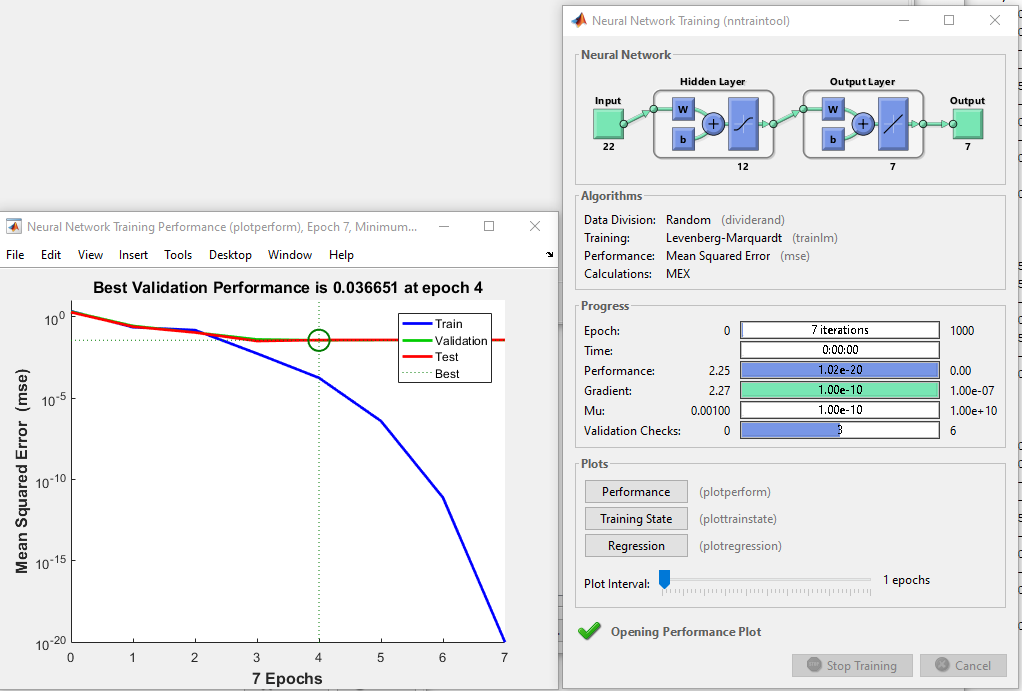
**Результаты обучения:**

На рисунках 1-18 представлены графики зависимости среднеквадратичной ошибки от количества итераций при аппроксимации заданной функции.

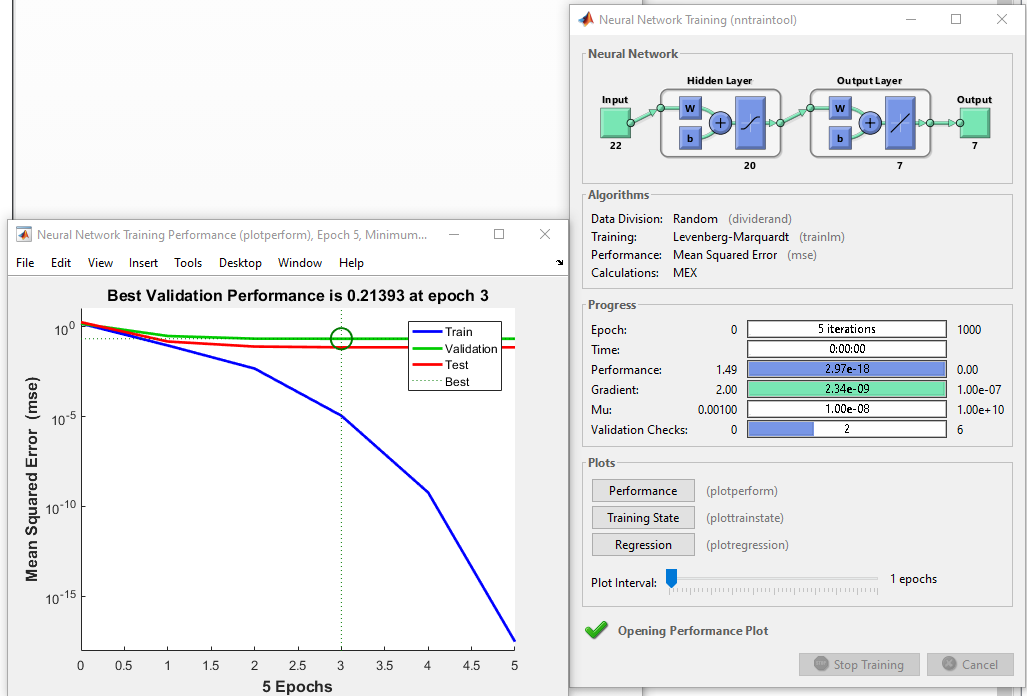
Рисунок 1 - график зависимости среднеквадратичной ошибки от количества итераций (trainlm 5 7) при аппроксимации заданой функции.



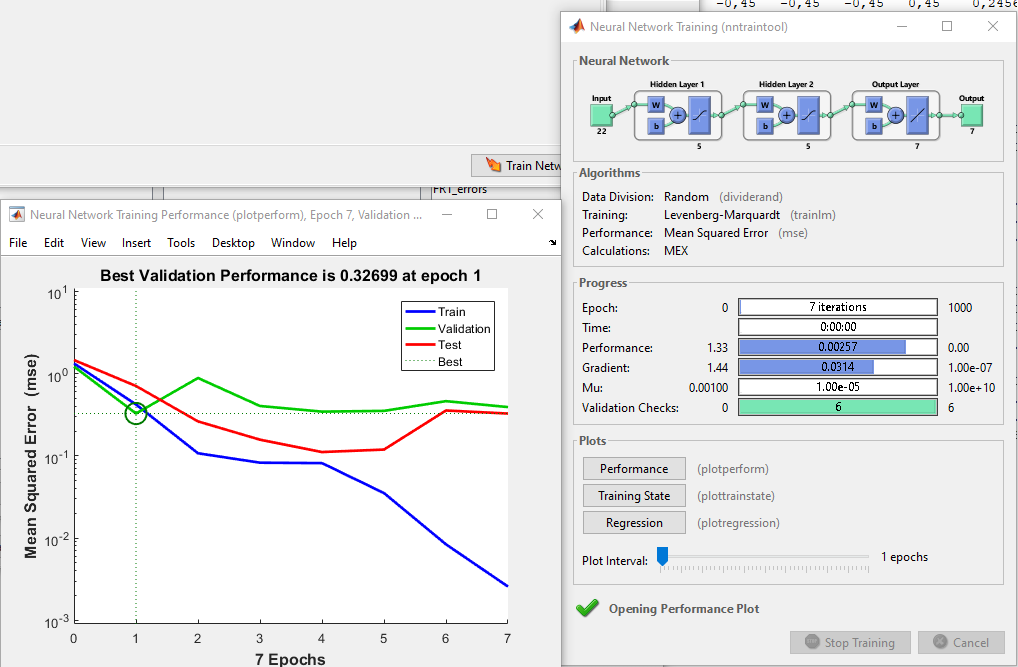
*Рисунок 1 - график зависимости среднеквадратичной ошибки от количества итераций (trainlm 5 7) при аппроксимации заданой функции.*



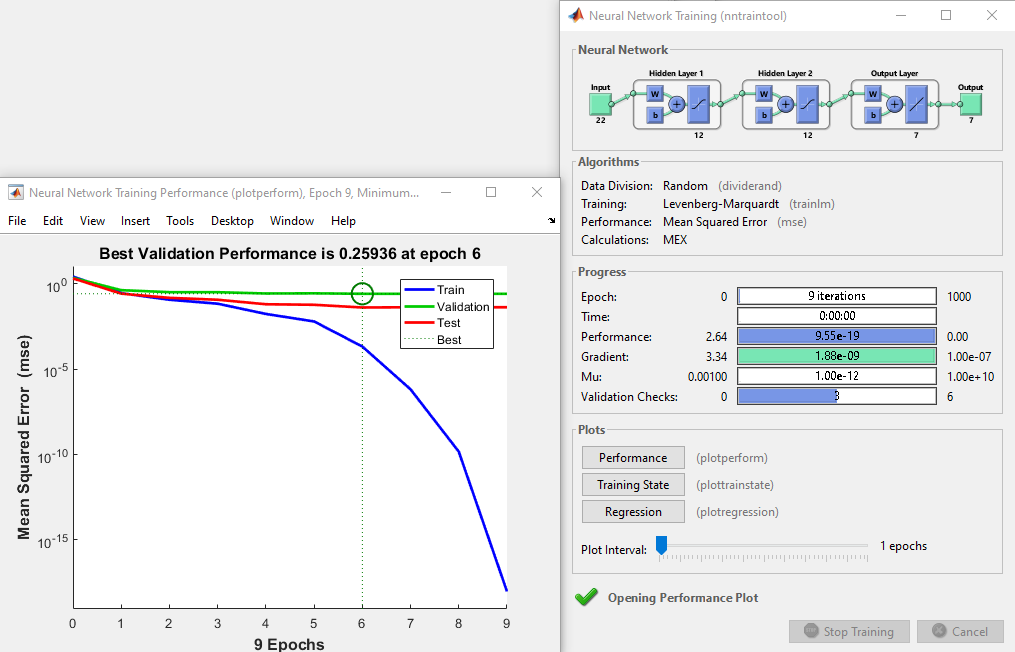
*Рисунок 2 - график зависимости среднеквадратичной ошибки от количества итераций (trainlm 12 7) при аппроксимации заданой функции.*



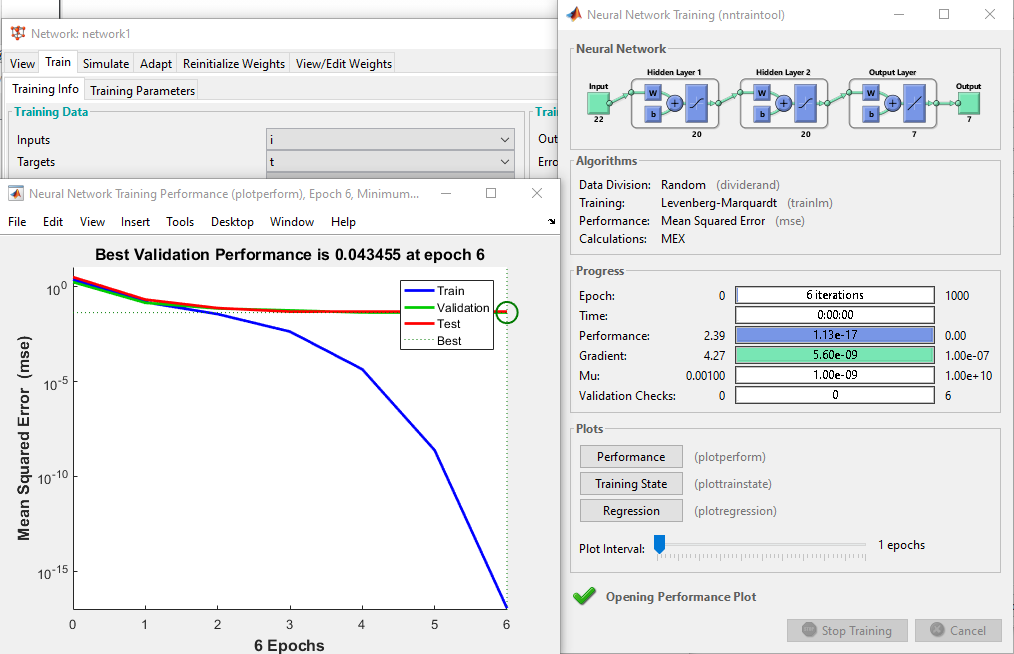
*Рисунок 3 - график зависимости среднеквадратичной ошибки от количества итераций (trainlm 20 7) при аппроксимации заданой функции.*



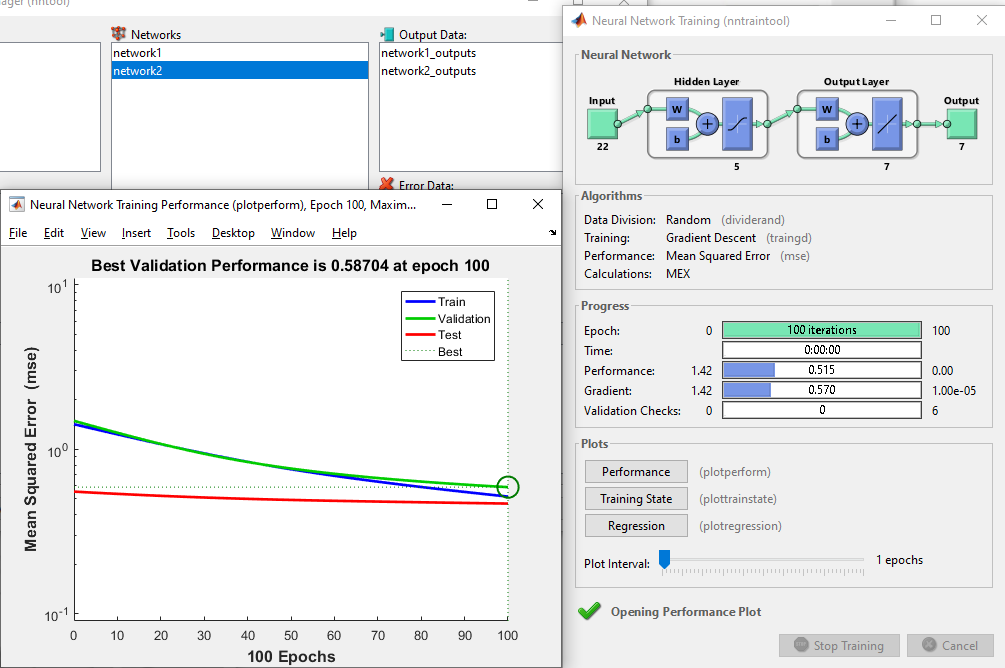
*Рисунок 4 - график зависимости среднеквадратичной ошибки от количества итераций (trainlm 5 5 7) при аппроксимации заданой функции.*



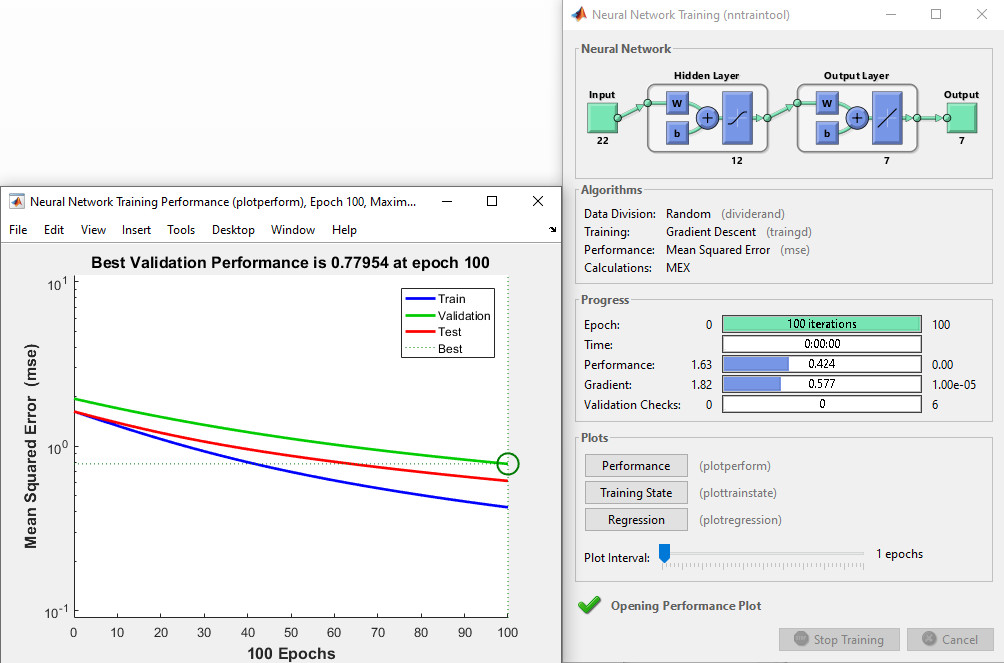
*Рисунок 5 - график зависимости среднеквадратичной ошибки от количества итераций (trainlm 12 12 7) при аппроксимации заданой функции.*



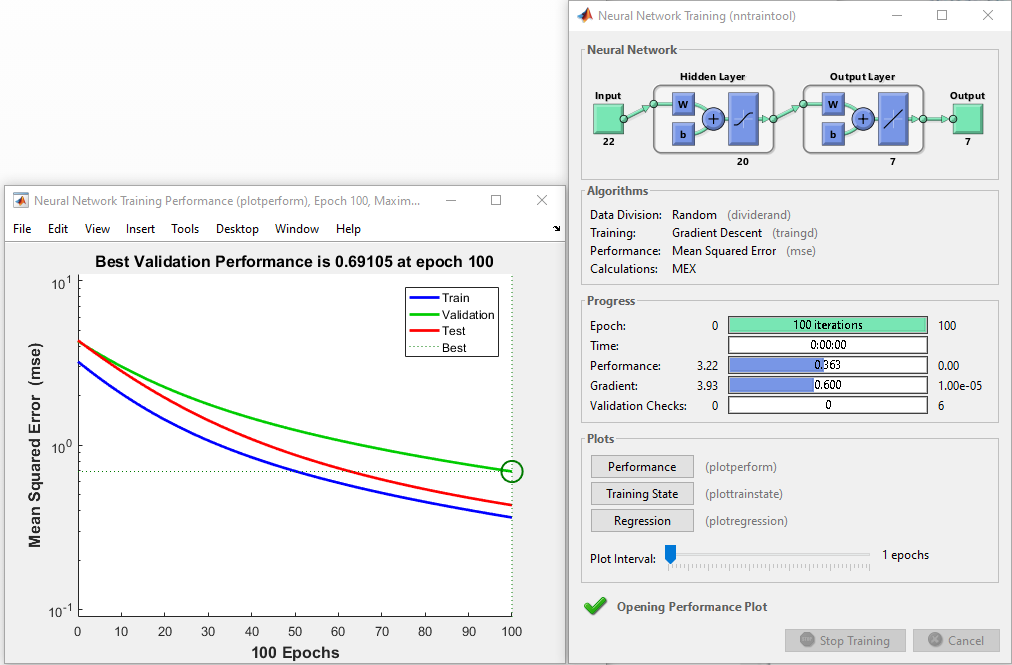
*Рисунок 6 - график зависимости среднеквадратичной ошибки от количества итераций (trainlm 20 20 7) при аппроксимации заданой функции.*



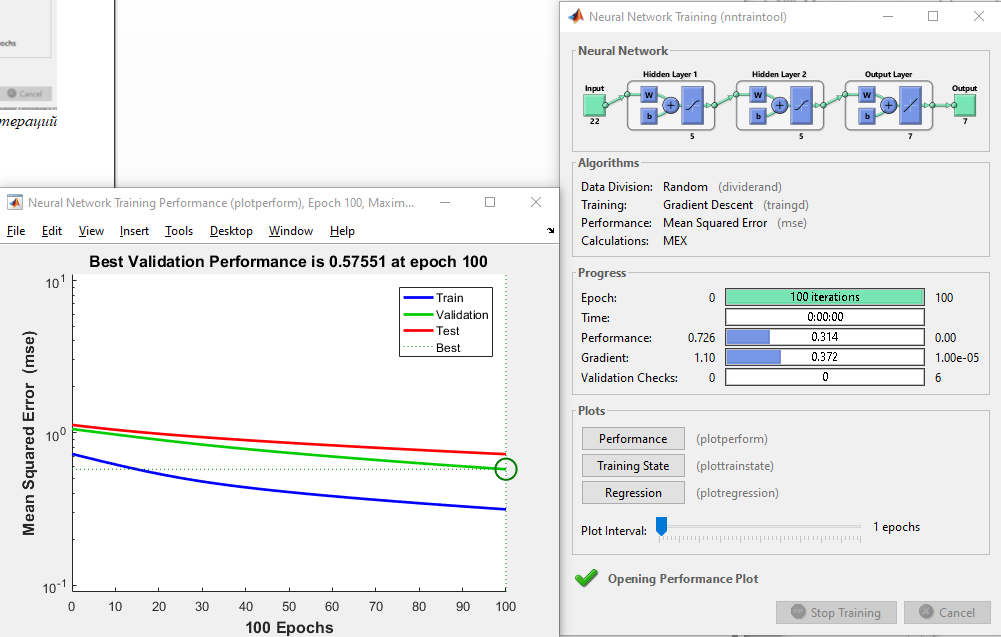
*Рисунок 7 - график зависимости среднеквадратичной ошибки от количества итераций (traingd 5 7) при аппроксимации заданой функции.*



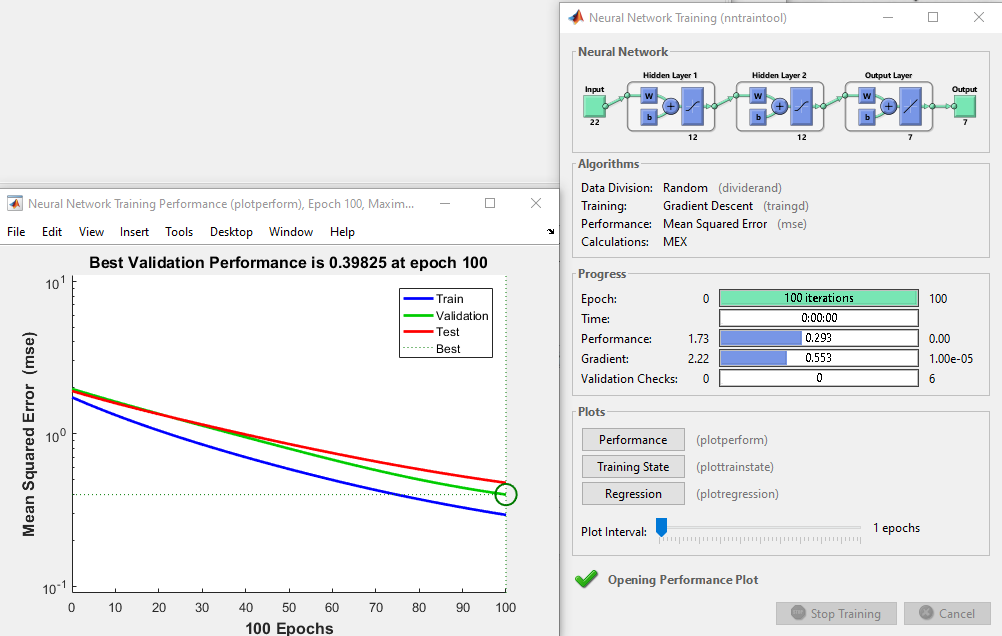
*Рисунок 8 - график зависимости среднеквадратичной ошибки от количества итераций (traingd 12 7) при аппроксимации заданой функции.*



*Рисунок 9 - график зависимости среднеквадратичной ошибки от количества итераций (traingd 20 7) при аппроксимации заданой функции.*



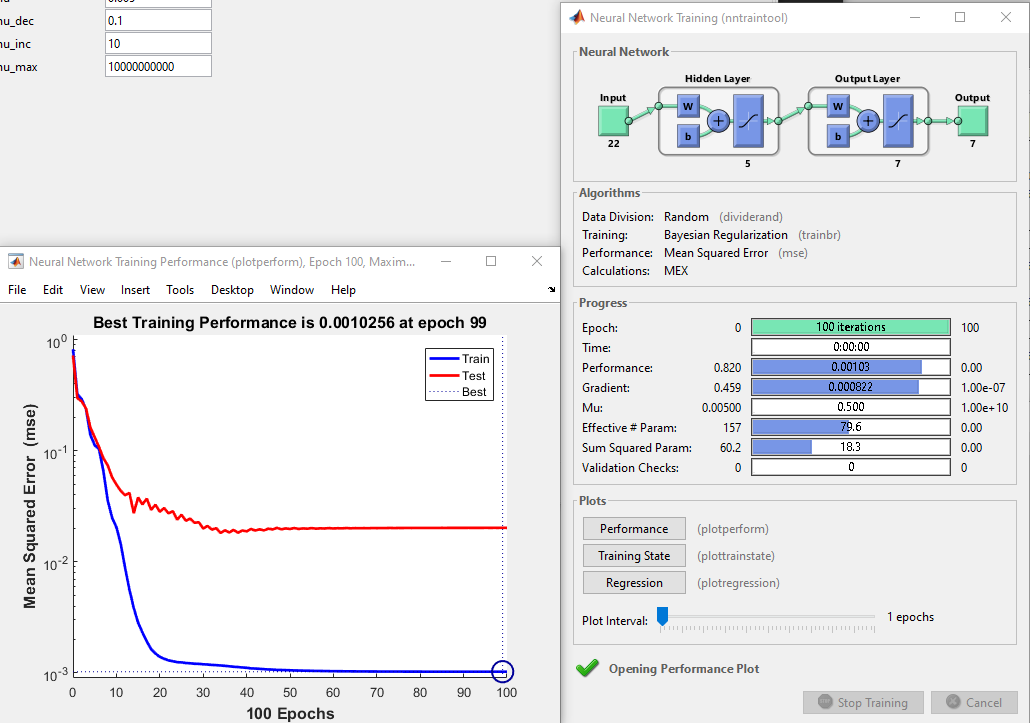
*Рисунок 10 - график зависимости среднеквадратичной ошибки от количества итераций (traingd 5 5 7) при аппроксимации заданой функции.*



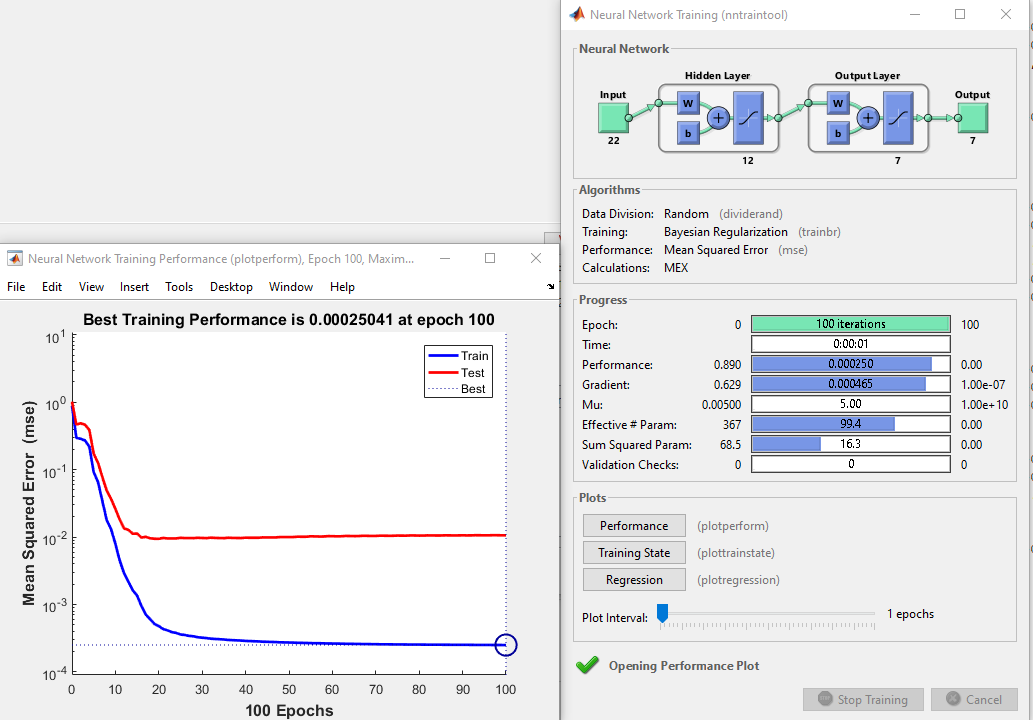
*Рисунок 11 - график зависимости среднеквадратичной ошибки от количества итераций (traingd 12 12 7) при аппроксимации заданой функции.*



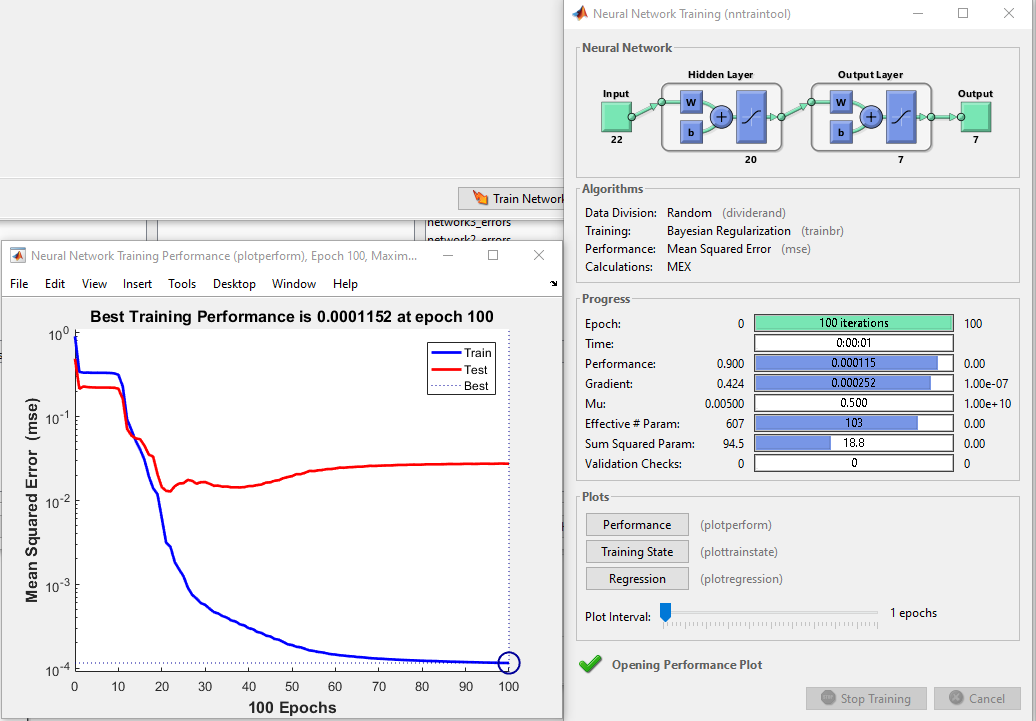
*Рисунок 12 - график зависимости среднеквадратичной ошибки от количества итераций (traingd 20 20 7) при аппроксимации заданой функции.*



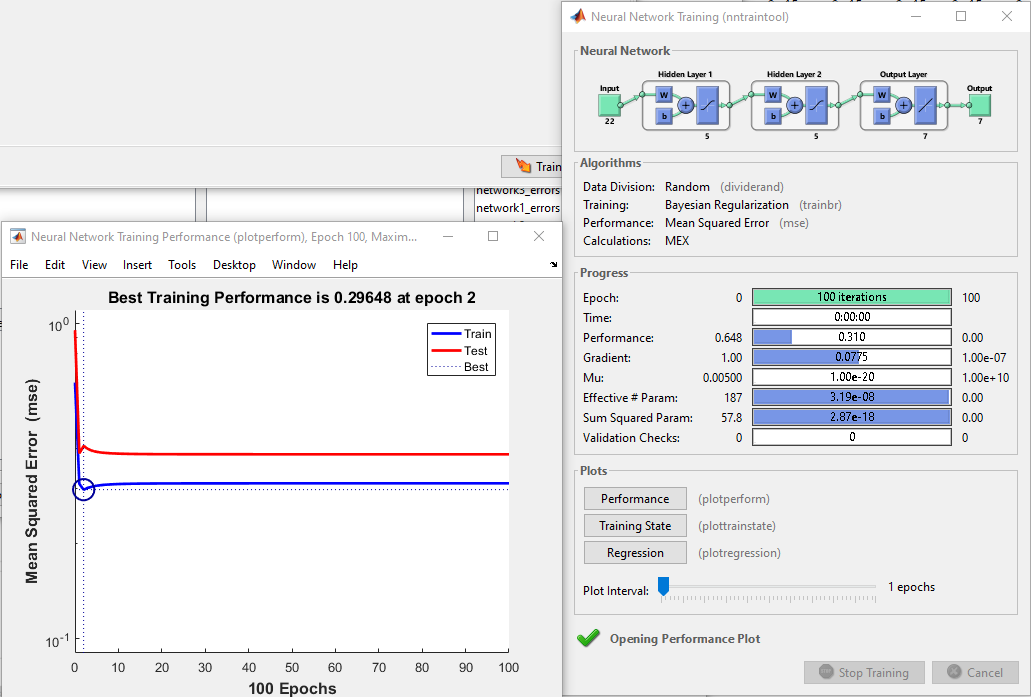
*Рисунок 13 - график зависимости среднеквадратичной ошибки от количества итераций (trainbr 5 7) при аппроксимации заданой функции.*



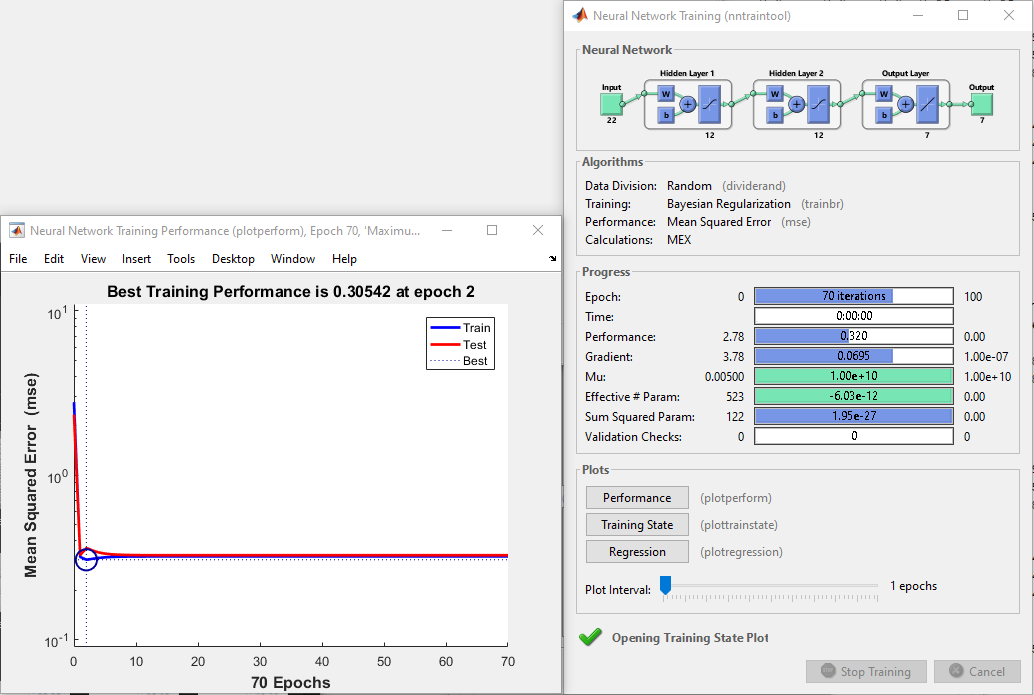
*Рисунок 14 - график зависимости среднеквадратичной ошибки от количества итераций (trainbr 12 7) при аппроксимации заданой функции.*



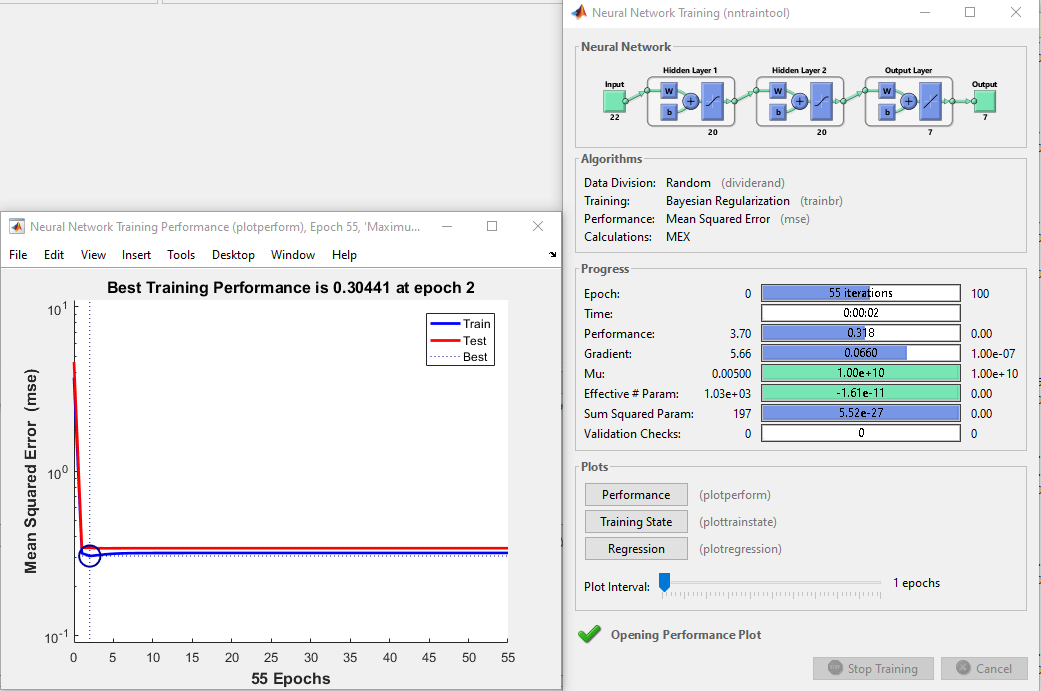
*Рисунок 15 - график зависимости среднеквадратичной ошибки от количества итераций (trainbr 20 7) при аппроксимации заданой функции.*



*Рисунок 16 - график зависимости среднеквадратичной ошибки от количества итераций (trainbr 5 5 7) при аппроксимации заданой функции.*



*Рисунок 17 - график зависимости среднеквадратичной ошибки от количества итераций (trainbr 12 12 7) при аппроксимации заданой функции.*



*Рисунок 18 - график зависимости среднеквадратичной ошибки от количества итераций (trainbr 20 20 7) при аппроксимации заданой функции.*

**4 Выводы**

Наилучшие результаты аппроксимации функции были получены с архитектурой trainbr 20 7 (MSE = 0.0001152). Аппроксимация градиентным спуском дала наихудшие результаты.

Алгоритм градиентного спуска проявил очень медленную работу при обучении НС. Увеличение количества слоев и нейронов в слоях дает улучшение среднеквадратичной ошибки.

Алгоритм Левенберга-Марквардта, дополненная регуляцией по Байесу BR дает неплохой результат при обучении нейронной сети с двумя слоями, но при добавлении третьего слоя наблюдается переобучение НС при любом количестве нейронов в слоях, среднеквадратичная ошибка остается практически неизменной. Увеличение количества нейронов в двухслойной нейронной сети дает улучшение среднеквадратичной ошибки.

Алгоритм Левенберга-Марквардта LM показал наилучшие результаты. Увеличение количества слоев и нейронов в слоях дает улучшение среднеквадратичной ошибки. При использовании 5 нейронов в скрытых слоях как в двухслойной, так и в трехслойной НС наблюдается переобучение сети.