МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КУБГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**Отчет**

**по лабораторной работе №5 по курсу**

**«НЕЙРОСЕТЕВЫЕ И НЕЧЕТКИЕ МОДЕЛИ»**

Работу выполнил

Студент 49 группы

Иванова В. А

Преподаватель:

Крамаренко А. А.

Краснодар

2024

**Цель работы:** нейронные сети в пакете MATLAB.

**Ход работы:**

**Задание №1.** Необходимо реализовать различные подходы к аппроксимации функций с применением аппарата нечетких множеств (нечеткого логического вывода) и искусственных нейронных сетей в качестве универсальность аппроксиматора. Выполнить аппроксимацию функции согласно варианту (вариант №3).

Рассматриваемая функция:

на области , где *k* – номер варианта согласно заданию.

Через команду nntool запускаем Neural Network/Data Manager, в котором мы можем выбрать нужную структуру NN и алгоритм для обучения сети.

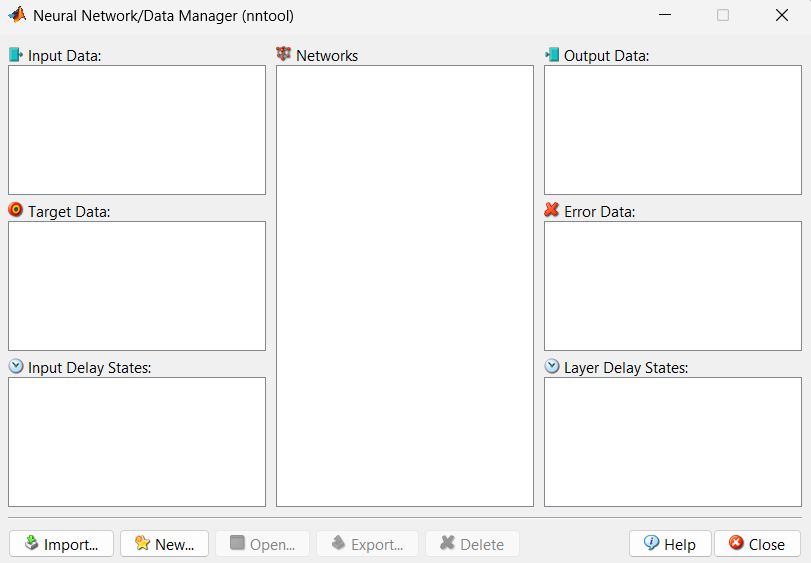


Рисунок 1 – Окно-менеджер NNTool

Переходим к созданию нейронной сети по указанной задаче. Получаем обучающую выборку выданной функции

определенной в области .

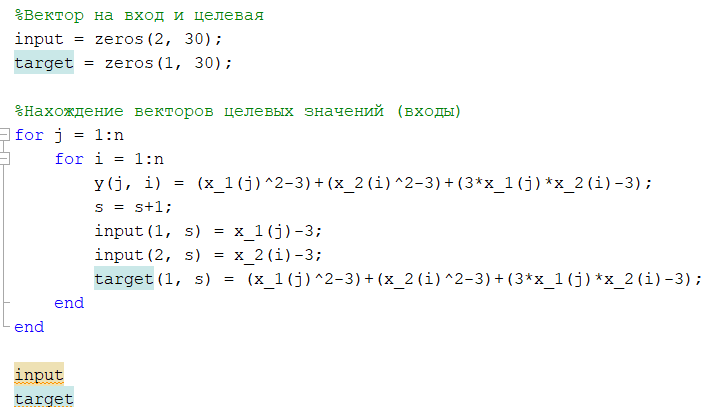


Рисунок 2 – Блок кода для вычисления целевых значений

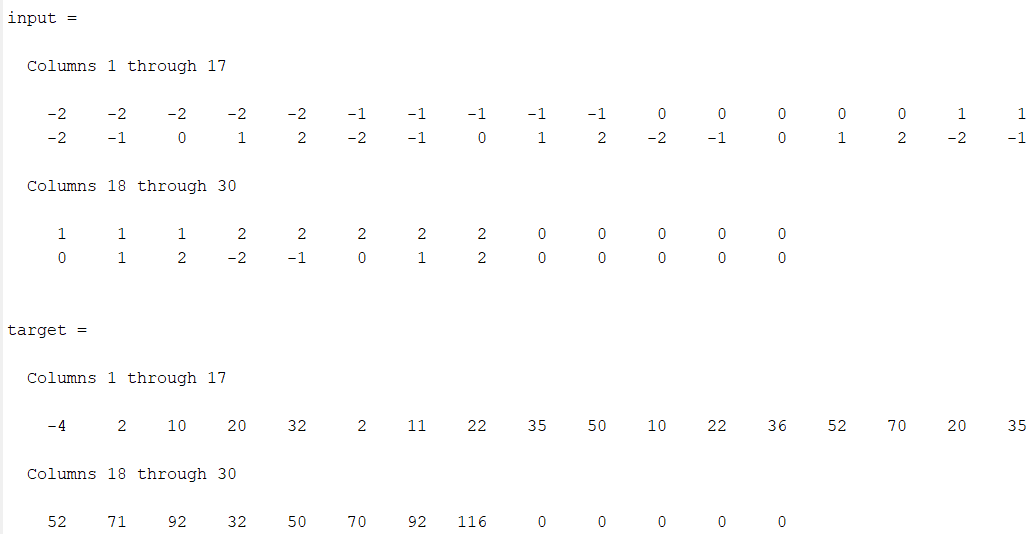


Рисунок 3 – Вывод полученных значений

Переходим к окну NNTool. Создаем новую сеть, в Network выбираем персептрон (Feed-Forward backprop) с 30 сигмоидными нейронами (TANSIG) скрытого слоя и одним линейным (PURELIN) нейроном выходного слоя.

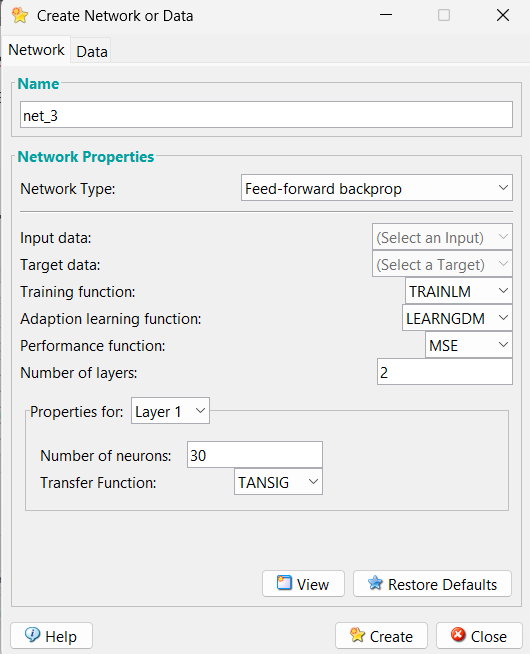


Рисунок 4 – Настройки для создания сети

Во вкладке Data добавляем входные и целевые значения, которые получили ранее, и сохраняем.

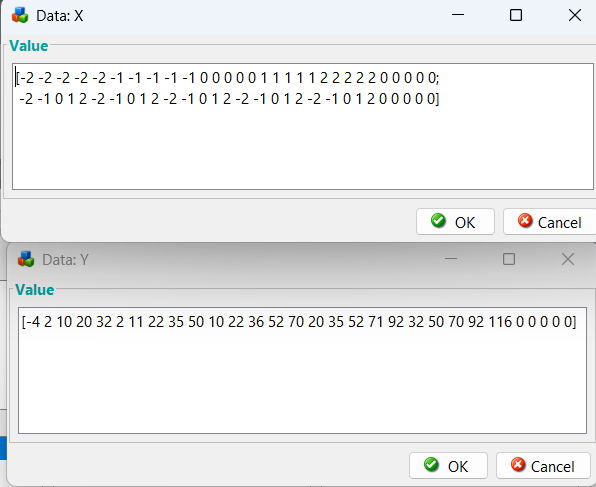


Рисунок 5 – Запись необходимых значений

Добавляем в окне настройки выше значения X и Y, выбираем View и открываем вид сети.

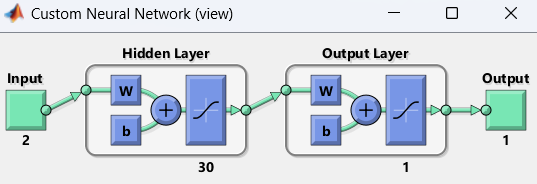


Рисунок 6 – Вид созданной сети

Переходим к этапу обучения созданной сети. Для этого используем алгоритм обратного распространения ошибки с минимизацией функцией ошибки по методу Левенберга-Маркардта (функция TRAINLM). В функции ошибки берем MSE. В окне Network выбираем на входные параметры, целевые X и Y соответственно. Для выходных значений и ошибок будут автоматически созданы network1\_outputs и network1\_errors.

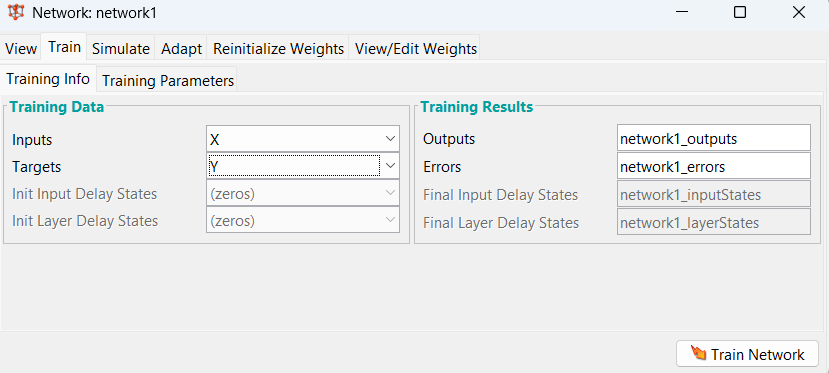


Рисунок 7 – Выбор необходимых параметров для обучения сети

После выбора Train Network переходим к окну с отображением сети и получаемым параметрам.

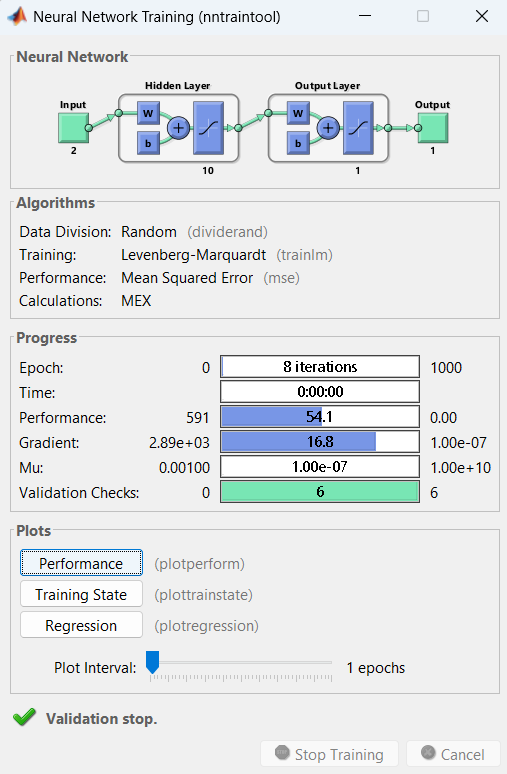


Рисунок 8 – Результат обучения сети

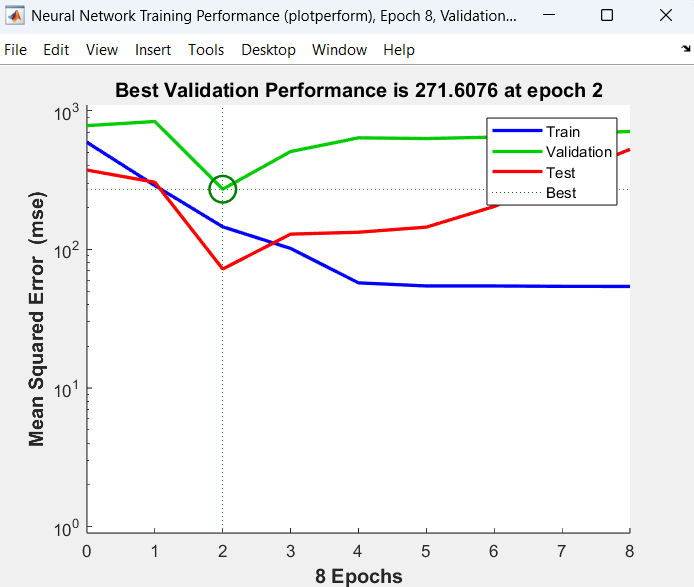


Рисунок 9 – Отображение обучения сети

Теперь необходимо проверить приближение функции. Переходим к Network → Simulate, указываем вектор входных значений и нажимаем на Simulate Network.

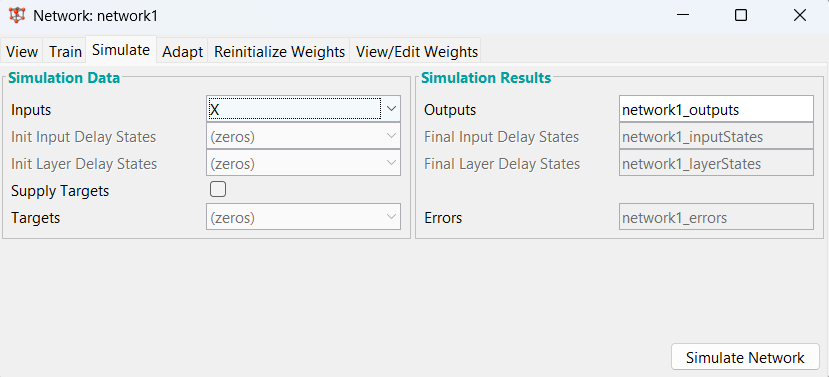


Рисунок 10 – Проверка работы сети

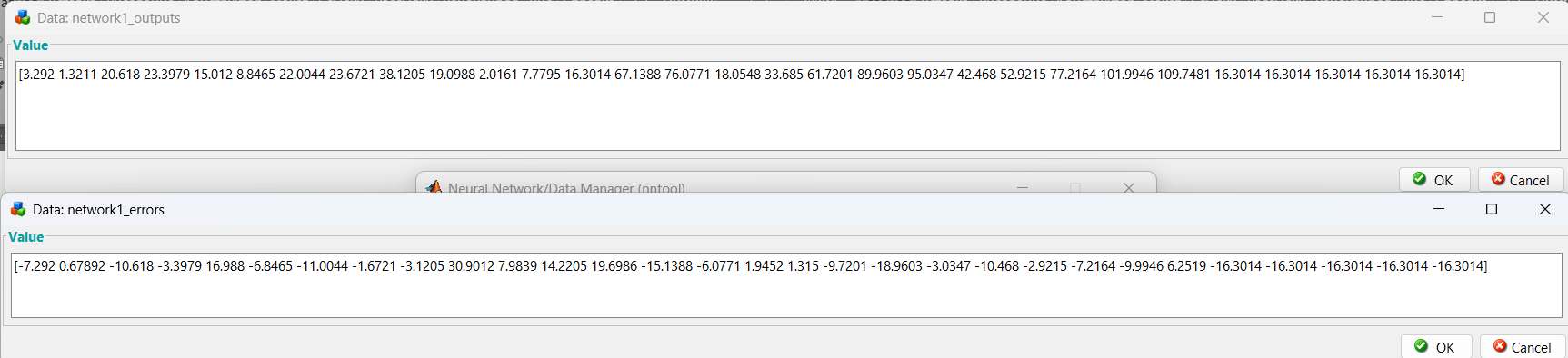


Рисунок 11 – Выходные значения и ошибки

Переходим к оценке погрешности аппроксимации, вычисляемой по следующей формуле:

Задаем массивы выходных и целевых значений, переходим в цикл, где вычисляем их разность и возводим в квадрат.

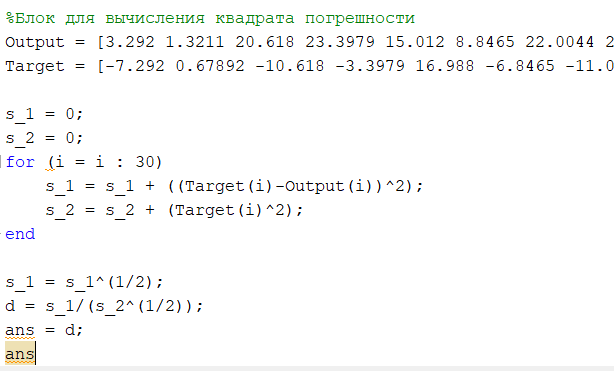


Рисунок 12 – Блок кода для вычисления оценки погрешности

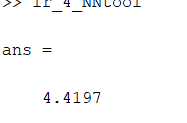


Рисунок 13 – Получение оценки погрешности

Полученный результат может указывать на нестабильность обучаемой модели.

**Вывод:** ознакомились с графическим интерфейсом пользователя NNTool, создали и обучили модель, нашли оценку погрешности согласно варианту.