

Современные проблемы информатики и вычислительной техники

Лабораторная работа №3

Задание

1. Сформулировать задачу прогнозирования из области информатики вычислительной техники, для решения которой было бы обосновано применение НС и ННС.
2. Сформировать обучающую выборку для НС и ННС.
3. Построить НС и, экспериментируя с количеством нейронов во входных и скрытых слоях, функциями активации, методами обучения, добиться наилучшего результата по прогнозированию.
4. Визуализировать полученную структуру НС в системе MATLAB.
5. Построить ННС и, экспериментируя с методами обучения, количеством функций принадлежности во входном слое, добиться наилучшего результата прогнозирования. Количество входов взять таким же, как и в п. 3 индивидуального задания.
6. Визуализировать полученную структуру ННС в системе MATLAB. Построить систему нечеткого логического вывода для полученной ННС.
7. Сравнить полученные с помощью НС и ННС результаты (численно) и сделать выводы.

Ответ

Задача прогнозирования

Анализ и прогнозирование количества загрузок npm пакета Redux

Формирование обучающей выборки

npm предоставляет открытое api, позволяющие получить статистику скачивания пакета по дням за определённый период. Нужно сделать GET запрос по пути со схожей структурой <https://api.npmjs.org/downloads/range/2023-04-01:2023-04-30/redux>. (здесь получают данные для пакета redux за апрель 2023 года). Сформированы файлы training.dat (статистика с 2023-04-

01 по 2023-04-30), testing.dat (статистика с 2023-08-01 по 2023-08-29) и checking.dat (статистика за 2023-08-30).

Построение НС, эксперименты

Листинг 1: Построение и тренировка НС

```
clear;clc;
fileId = fopen('training.dat', 'r');
formatSpec = '%d';
trainingData = fscanf(fileId, formatSpec);
fclose(fileId);

% Использование первых 29 значений для обучения
input = trainingData(1:end-1);
% Использование последнего значения как цели обучения
output = trainingData(end);

% Преобразование входных и выходных данных в формат, подходящий для
сети
inputFormatted = reshape(input, [29, 1]);
outputFormatted = output;

% Создание и обучение сети
network = newff(minmax(inputFormatted), [10, 1], {'logsig',
'purelin'});
network = train(network, inputFormatted, outputFormatted);

% Предсказание для тестового набора данных
prediction = network(inputFormatted);
```

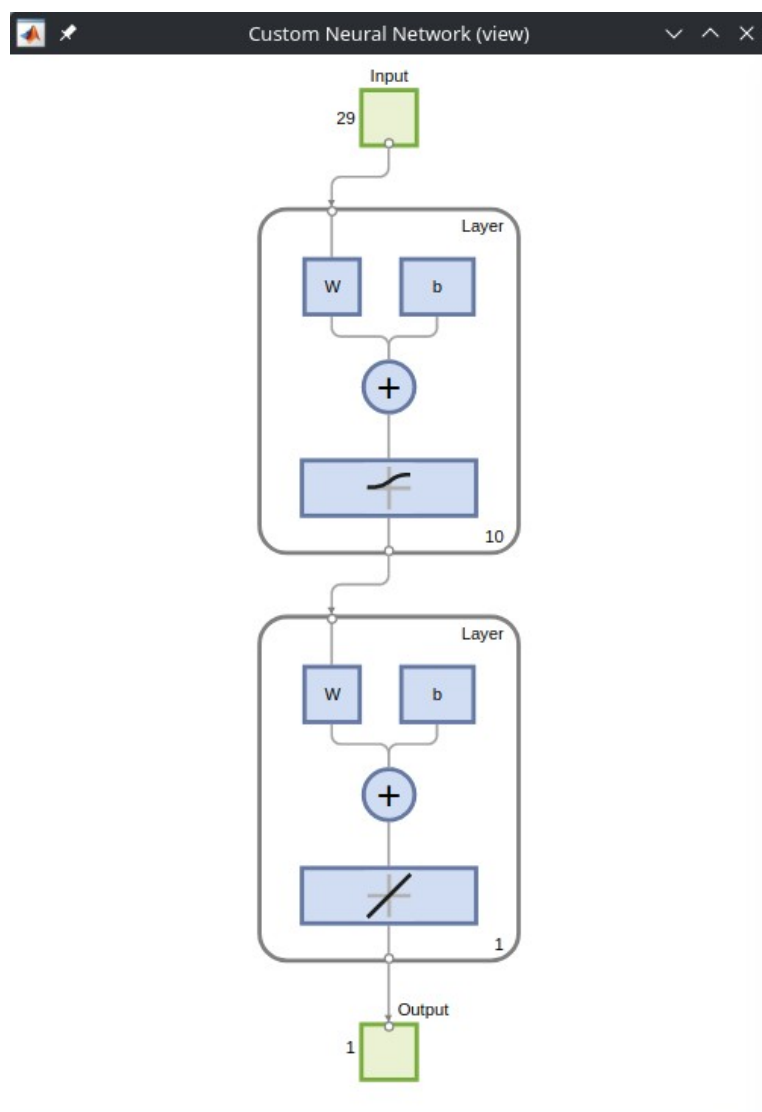


Рисунок 1: Схема НС

На такой маленькой выборке обучение получилось плохим, из-за природы данных. Немного переделал скрипт, увеличил количество слоёв.

Листинг 2. Обновленный скрипт.

```
clear;clc;
trainingFileId = fopen('training.dat', 'r');
formatSpec = '%d';
trainingData = fscanf(trainingFileId, formatSpec);
fclose(trainingFileId);
testingFileId = fopen('testing.dat', 'r');
testingData = fscanf(testingFileId, formatSpec);
fclose(testingFileId);
checkingFileId = fopen('checking.dat', 'r');
checkingData = fscanf(checkingFileId, formatSpec);
```

```
fclose(checkingFileId);

INPUT_SIZE = 13;
OUTPUT_SIZE = 1;

% Чтение данных
fileId = fopen('training.dat', 'r');
data = fscanf(fileId, '%d');
fclose(fileId);

% Создание новой матрицы
numRows = length(data) - INPUT_SIZE; % Количество строк в новой матрице
matrix = zeros(numRows, INPUT_SIZE + OUTPUT_SIZE);

for i = 1:numRows
    matrix(i, :) = data(i:i+INPUT_SIZE);
end

% Запись в файл
fileId = fopen('training-matrix.dat', 'w');
for i = 1:size(matrix, 1)
    fprintf(fileId, '%d ', matrix(i, :));
    fprintf(fileId, '\n');
end
fclose(fileId);

% Открытие файла
fileId = fopen('training-matrix.dat', 'r');

% Чтение и разделение данных
input = [];
output = [];
while ~feof(fileId)
    line = fgetl(fileId);
    nums = str2num(line);
    input = [input; nums(1:end-1)];
    output = [output; nums(end)];
end

fclose(fileId);

% Открытие файла
fileId = fopen('testing.dat', 'r');

% Чтение и разделение данных
```

```
testingInput = [];  
testingOutput = [];  
while ~feof(fileId)  
    line = fgetl(fileId);  
    nums = str2num(line);  
    testingInput = [testingInput; nums(1:end-1)];  
    testingOutput = [testingOutput; nums(end)];  
end  
  
fclose(fileId);  
  
% Теперь input и output содержат входные и выходные данные для обучения сети  
  
% Обучение нейронной сети  
network = newff(minmax(input'), [INPUT_SIZE, floor(INPUT_SIZE / 2), floor(INPUT_SIZE / 3 + 1), 1], {'logsig'  
'logsig' 'logsig', 'purelin'});  
network.trainParam.lr = 0.01;  
network.performFcn = 'mse';  
network.trainParam.epochs = 100;  
network.trainParam.goal = 1e-6;  
network = train(network, input', output');  
  
testSample = testingInput(1,:);  
  
% Проверка размера  
if size(testSample, 1) ~= network.inputs{1}.size  
    error('Размер входных данных не соответствует размеру входного слоя сети');  
end  
  
% Получение предсказания  
result = network(testSample)  
  
result =  
  
    1.6306e+06  
  
ans =  
  
    1604144
```

В целом, точность предсказания неплохая.

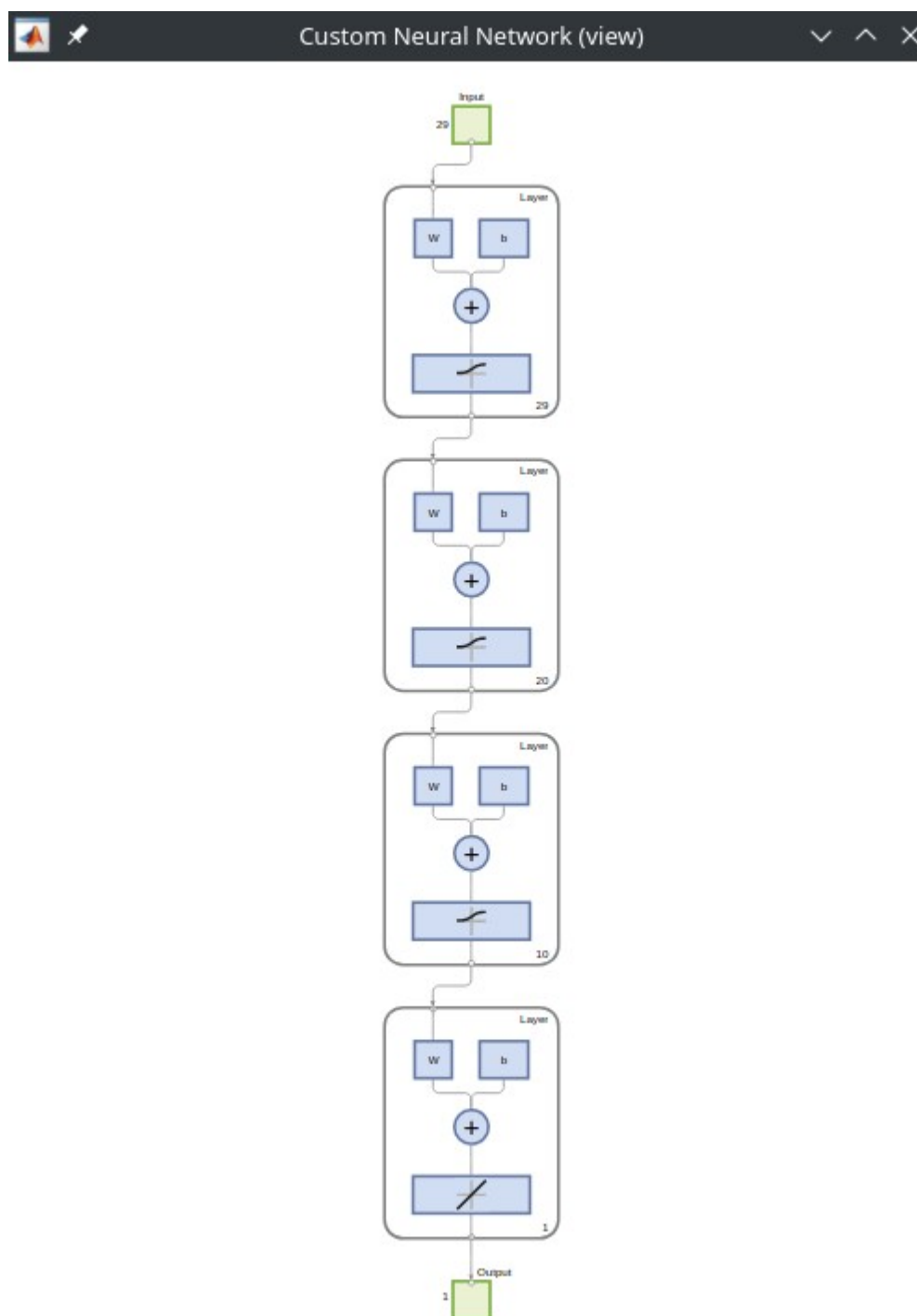


Рисунок 2: 4 слоя в НС

Построение ННС, эксперименты

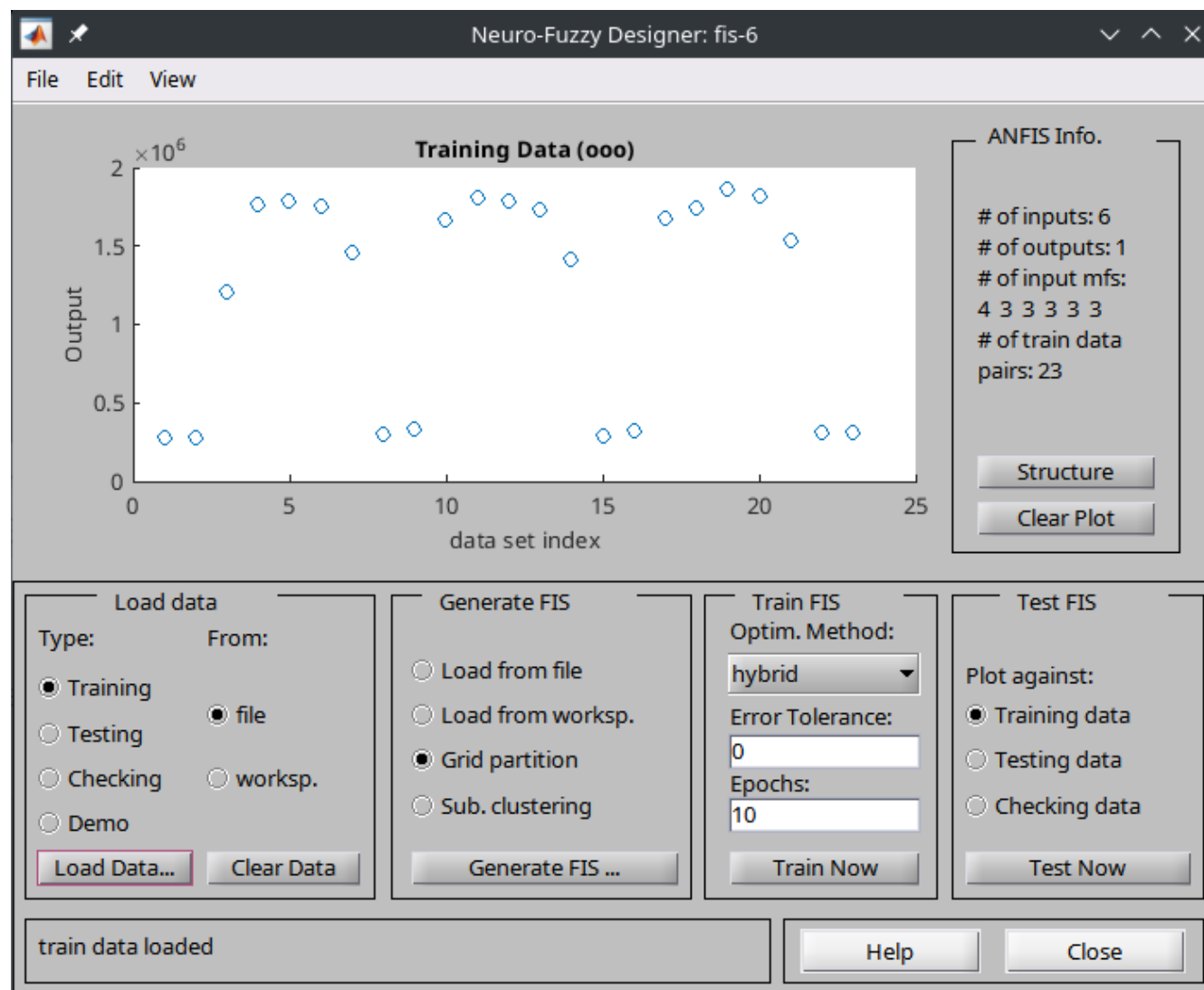


Рисунок 3: Визуализация тренировочных данных

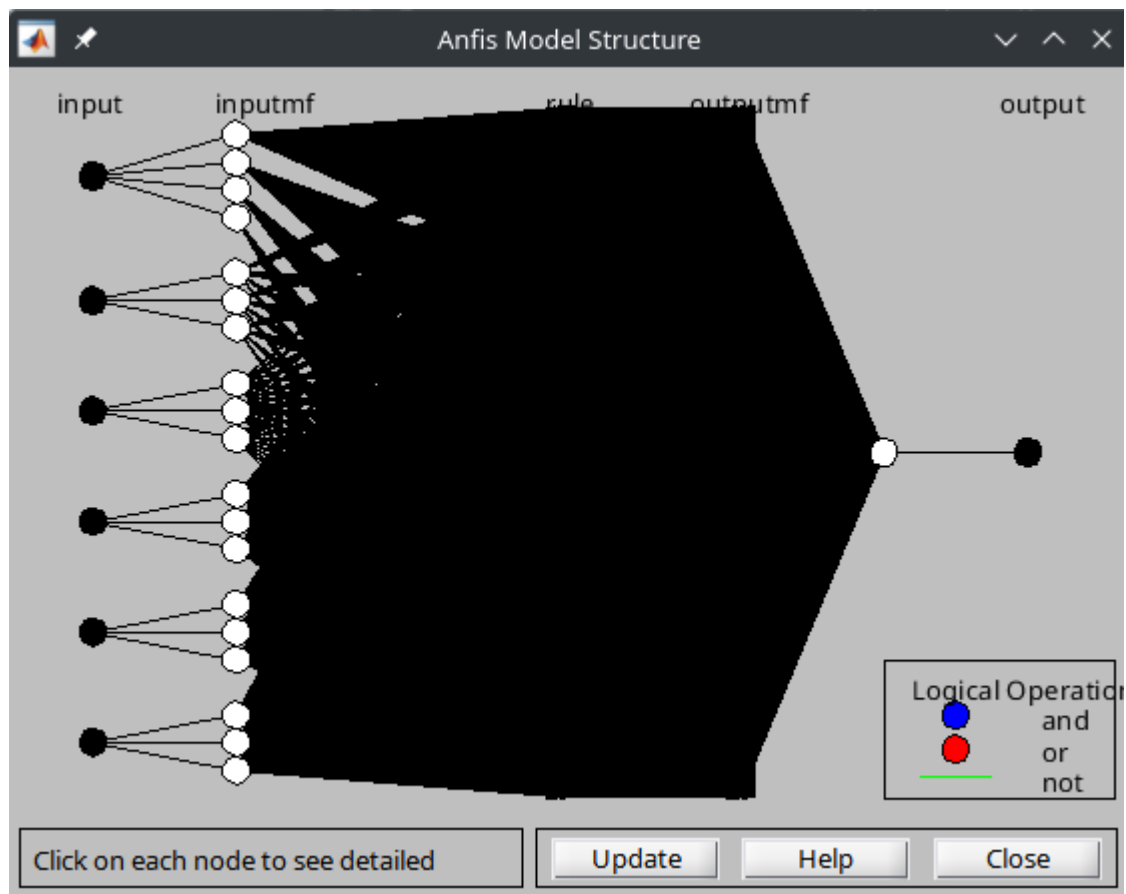
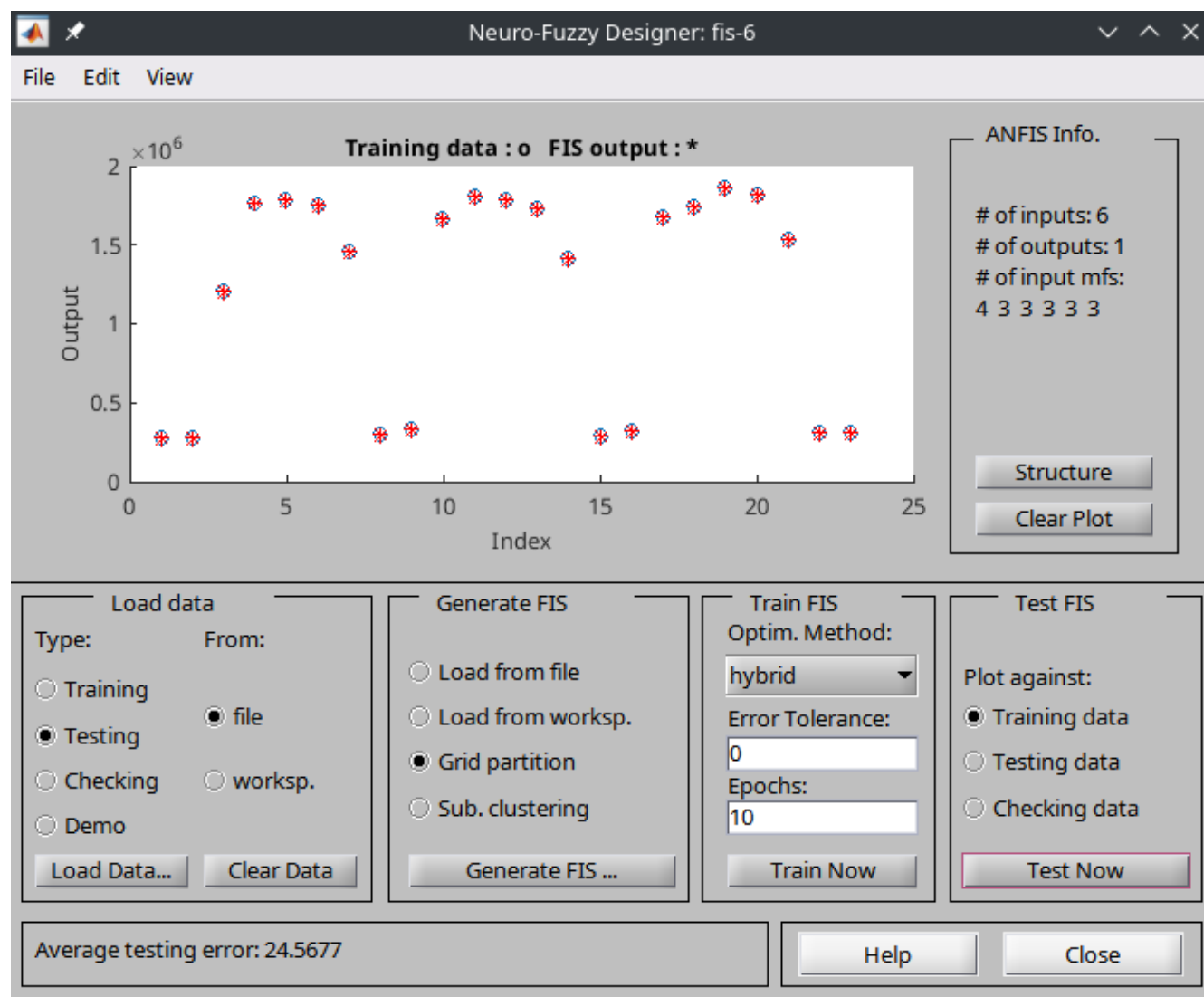
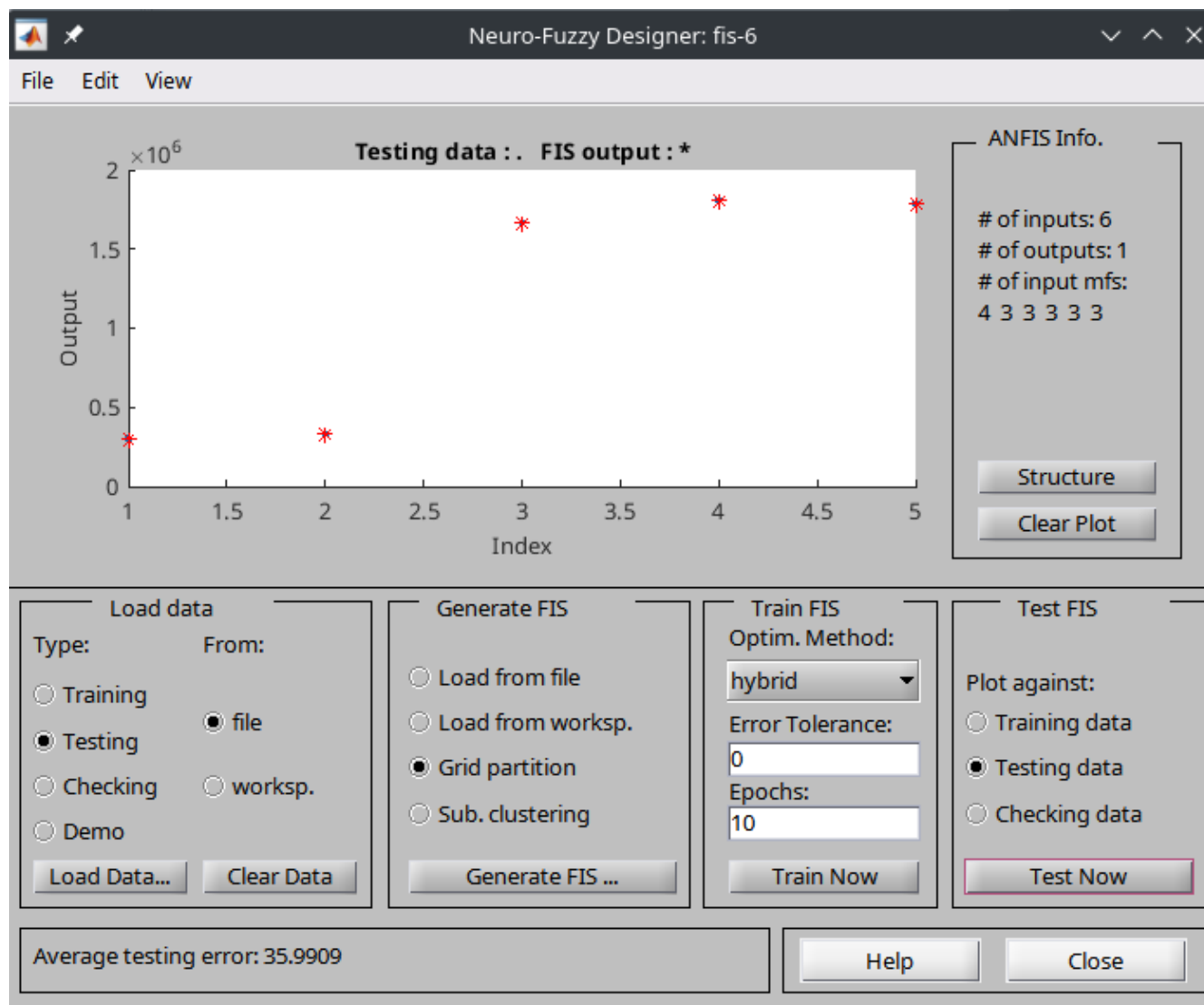


Рисунок 4: Структура ННС

Запустить с большими N у меня не получилось (получилось только для модели 6-1 [предсказание на следующий день по 6 предыдущим дням]) и небольшой обучающей выборке (20 строк). Иначе у меня крашился матлаб. Для модели 29-1 вообще не получилось запустить, матлаб поругался на превышенный размер.





В целом, ошибка тут меньше чем в обычной НС.

Выводы

Сформулирована задача прогнозирования из области информатики и вычислительной техники, для решения которой обосновано применение НС и ННС. Сформирована обучающая выборка для НС и ННС. Построена НС, в ходе экспериментов уменьшена ошибка прогнозирования. Визуализирована структура построенной НС. Построенна ННС, в ходе экспериментов уменьшена ошибка прогнозирования. Визуализирована полученная структура ННС.