Московский авиационный институт
(Национальный исследовательский университет)
Факультет прикладной математики и физики
Кафедра вычислительной математики и программирования

Лабораторная работа № 2

по курсу «Нейроинформатика»

Тема: Линейная нейронная сеть. Правило обучения Уидроу-Хоффа.

Студент: Ваньков Д. А.
Группа: 8О-407Б-17
Преподаватель: Аносова Н.П.

Москва, 2021

Постановка задачи

Исследование свойств линейной нейронной сети и алгоритмов её обучения, применение в задачах аппроксимации и фильтрации.

1. Использовать нейронную сеть с задержками для аппроксимации функции. В качестве метода обучения использовать адаптацию.
2. Использовать нейронную сеть с задержками для аппроксимации функции и выполнении много-шагового прогноза.
3. Использовать нейронную сеть в качестве адаптивного фильтра для подавления помех. Для настройки весовых коэффициентов использовать метод наименьших квадратов.

Условие

7.	$x = \sin(t^2 - 6t + 3), \quad t \in [0, 6],\ h = 0.025$ $x = \sin(\sin(t)t^2), \quad t \in [0, 3.5],\ h = 0.01$	$y = \frac{1}{4}\sin(\sin(t)t^2 - \pi)$

Ход работы

1.

График функции ошибки

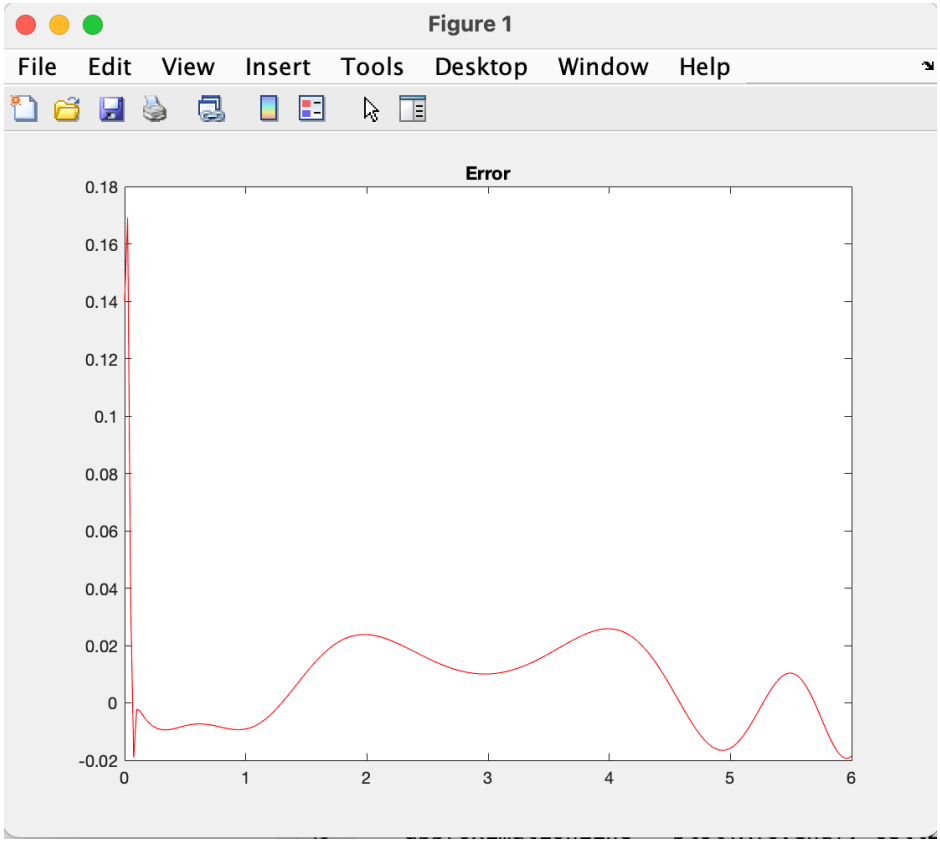
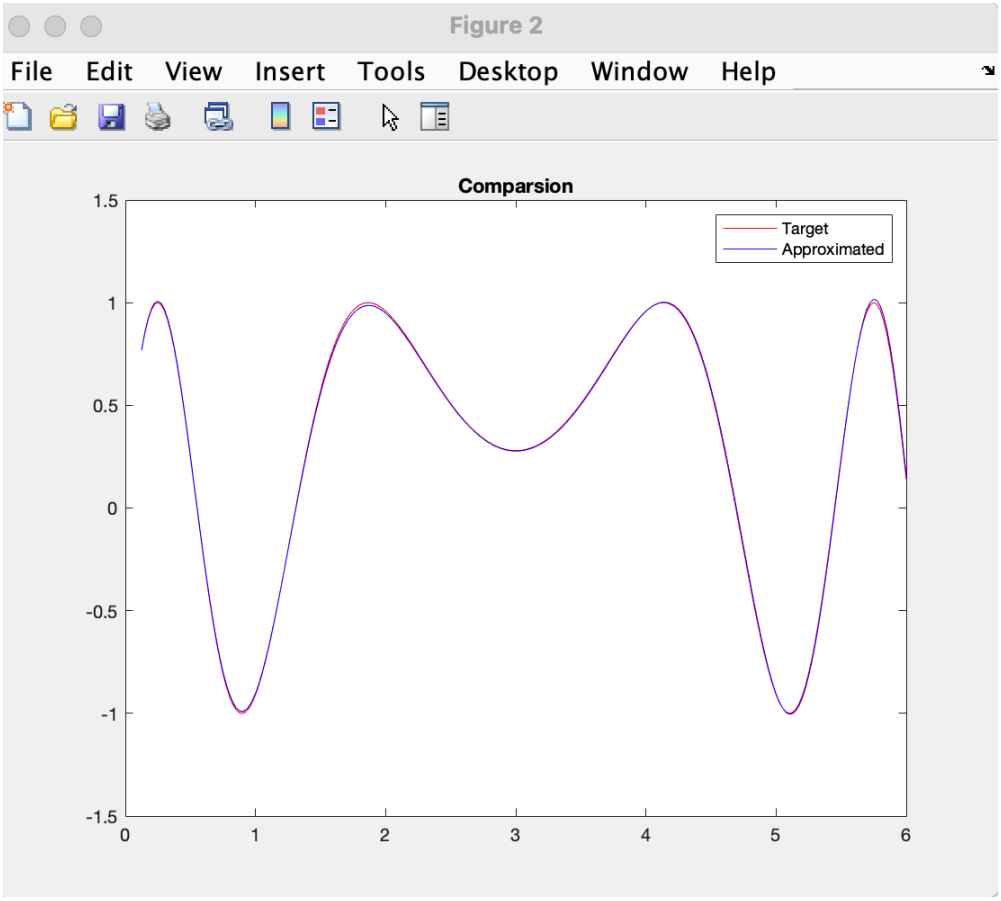


График сравнения эталонных значений и предсказанных




```
X = 0:h:6;
Y = sin((X.^2) - 6 * X + 3);

Pn = con2seq(Y);

% 1.2
% Инициализация задержки и скорости обучения
delays = [1 2 3 4 5];
net = newlin([-1 1], 1, delays, 0.01);

% 1.3
% Инициализация сети
net.inputweights{1,1}.initFcn = 'rands';
net.biases{1}.initFcn = 'rands';
net = init(net);

% 1.4
% Адаптация на цикле из 50ти шагов
Pi = con2seq(Y(1:5));
P = Pn(6:end);
T = Pn(6:end);

for i = 1:50
    [net, Y, E] = adapt(net, P, T, Pi);
end

% Ошибка обучения
err = sqrt(mse(E));
display(err);

% Инициализация графика ошибки
figure

E = cell2mat(Pn) - cell2mat(sim(net, Pn));
plot(X, E, 'r');
title('Error');

%1.5
% Эталонные значения
figure
referenceLine = plot(X(6:end), cell2mat(T), 'r');
hold on;

% Предсказанные значения
approximationLine = plot(X(6:end), cell2mat(Y), 'b');
```

2.

Структура сети

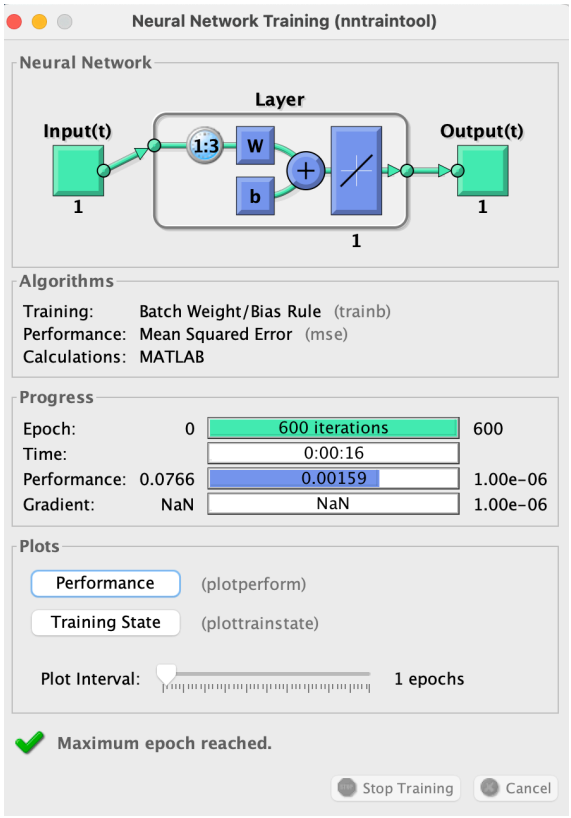


График функции ошибки

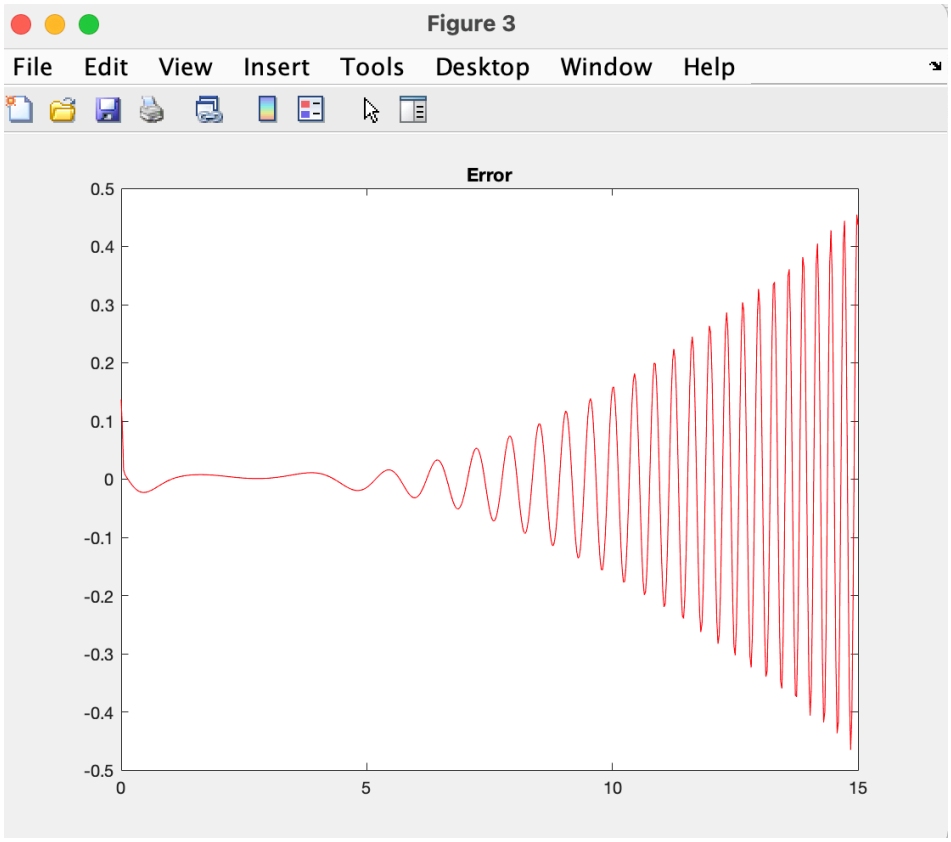
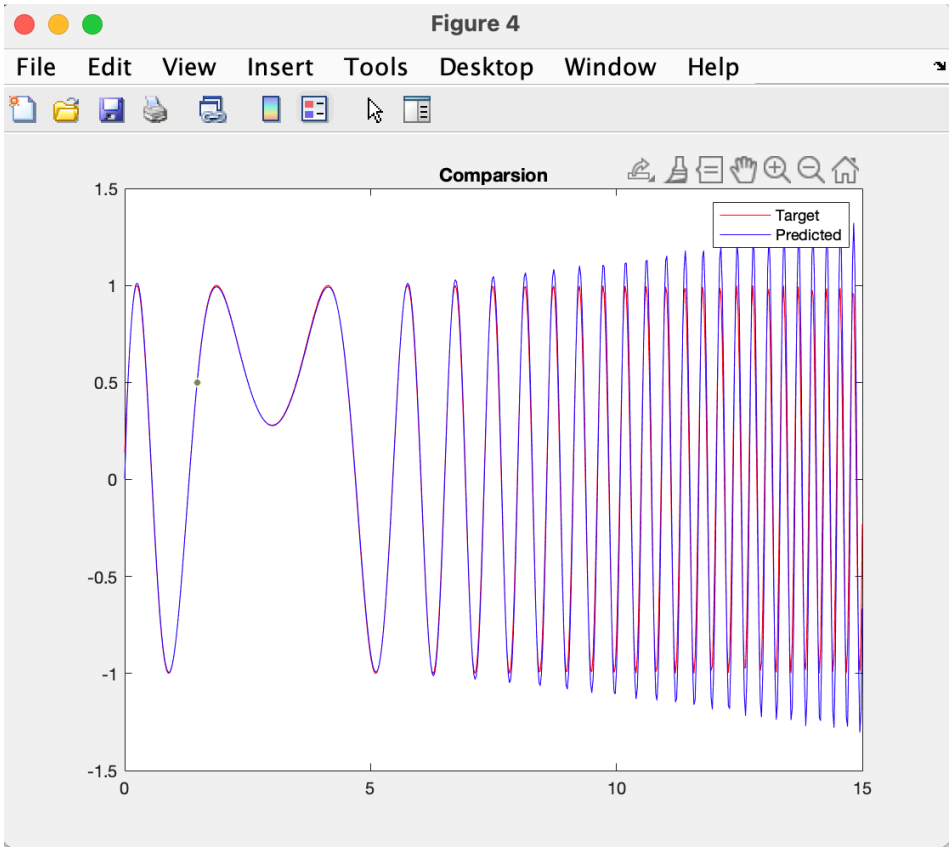


График сравнения эталонных значений и предсказанных




```
% 2.1
% Задание входной последовательности
h = 0.025;

X = 0:h:6;
Y = sin((X.^2) - 6 * X + 3);

Pn = con2seq(Y);

% 2.2
% Инициализация задержки и скорости обучения
delays = [1 2 3];
net = newlin([-1 1], 1, delays, maxlinlr(cell2mat(Pn),'bias'));

% 2.3
% Инициализация сети
net.inputweights{1,1}.initFcn = 'rands';
net.biases{1}.initFcn = 'rands';
net = init(net);

% 2.4
% Обучение на 600 эпохах
Pi = con2seq(Y(1:3));
P = Pn(4:end);
T = Pn(4:end);

net.trainParam.epochs = 600;
net.trainParam.goal = 0.000001;
net = adapt(net, P, T, Pi);
net = train(net, P, T);
E = cell2mat(Pn) - cell2mat(sim(net, Pn));
```

```
figure
plot(X, E, 'r');
title('Error');

figure
referenceLine = plot(X, Y, 'r');
hold on;

% 2.6
predictionLine = plot(X, cell2mat(sim(net, Pn)), 'b');

legend([referenceLine,predictionLine],'Target', 'Predicted');
title('Comparsion')
hold off;

% 2.7
X = 0:h:15;
Y = sin((X.^2) - 6 * X + 3);
Pn = con2seq(Y);

E = cell2mat(Pn) - cell2mat(sim(net, Pn));

figure
plot(X, E, 'r');
title('Error');

figure
referenceLine = plot(X, Y, 'r');
hold on;

predictionLine = plot(X, cell2mat(sim(net, Pn)), 'b');

legend([referenceLine,predictionLine],'Target', 'Predicted');
title('Comparsion')
hold off;
```

Таблица 1.

Функция создания сети	newlin
Входной слой	1
Скрытый слой	0
Выходной слой	1
Активационные функции	purelin
Динамика задержки	1, 2, 3
Число примеров в подмножествах	238
Метод обучения	train
Параметры обучения	lr = 0.01
Метод инициализации сети	rands
Критерий окончания обучения	epochs = 600, goal = 1E-6
Причина окончания обучения	Достигнуто максимальное значение количества эпох
Число эпох обучения	600

3.

1×2 string array

"Weights: " "-0.125 -7.9596e-15 4.4522e-15 -3.1238e…"

1×2 string array

"Biases: " "1.3046e-18"

График функции ошибки

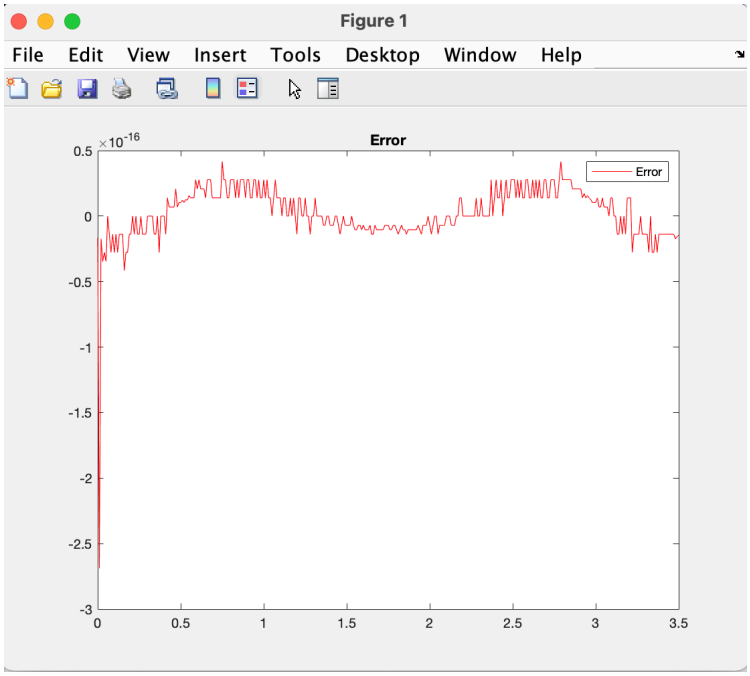
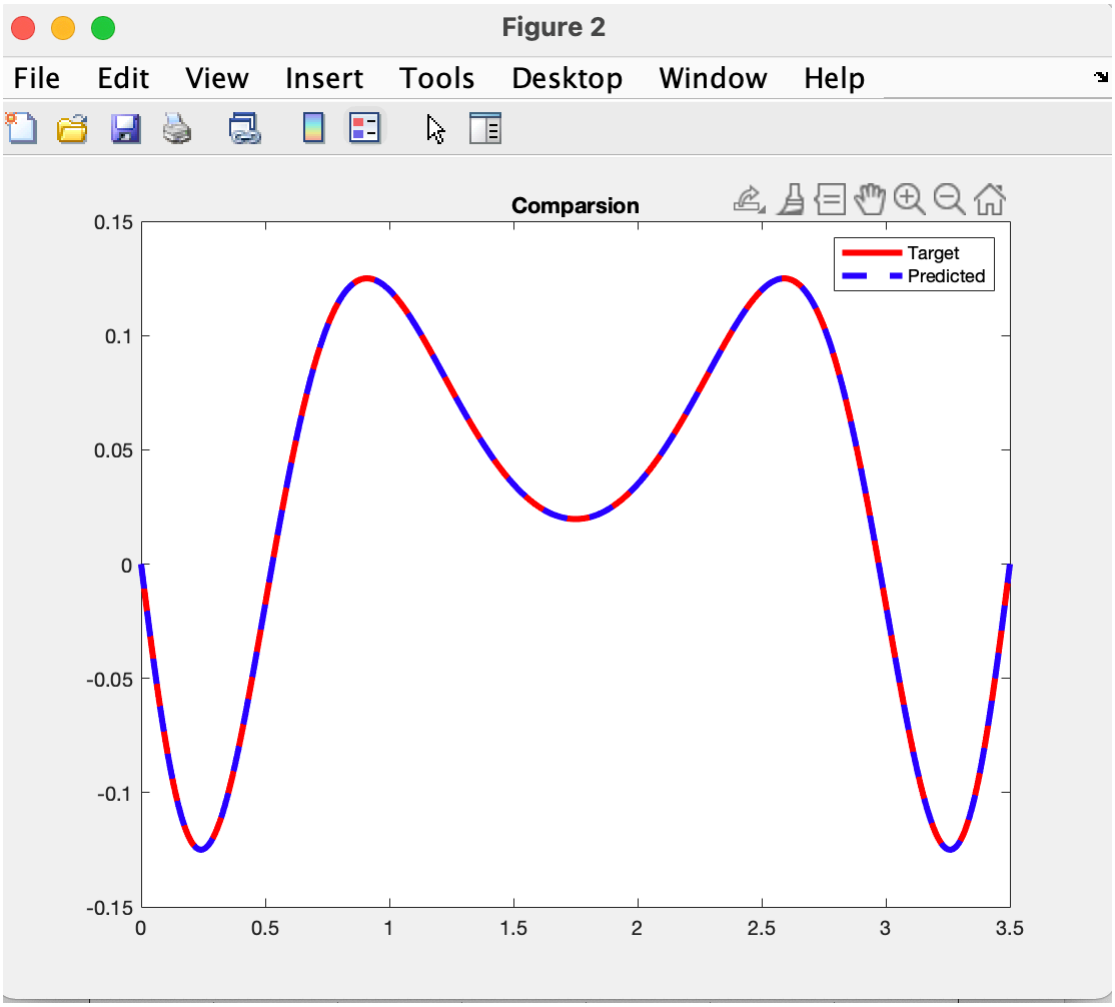


График сравнения эталонных значений и предсказанных




```matlab
% 3.1
% Задание входной последовательности
h = 0.01;
X = 0:h:3.5;

% Входное множество
x = sin(-2 * X.^2 + 7 * X);

% Эталонный выход
Y = 0.125 * sin(-2 * X.^2 + 7 * X - pi);

% 3.2
% Расширение входного множества с глубиной погружения
D = 4;
Q = length(x);
P = zeros(D, Q);

for i=1:D
    P(i, i:Q) = x(1:Q - i + 1);
end

% 3.3
% Инициализация сети
net = newlind(P, Y);
```

```matlab
% Вывод полученных весов и смещения
display(["Weights: ", num2str(net.IW{1,1})]);
display(["Biases: ", num2str(net.b{1})]);

T = sim(net, P);

% Погрешность
E = Y - T;

figure
err = plot(X, E, 'r');
title('Error');
legend(err,'Error');

% Эталонные значения
figure
referenceLine = plot(X, Y, 'r');
set(referenceLine, 'linewidth', 3);
hold on;

% Предсказанные значения
approximationLine = plot(X, T, '--b');
set(approximationLine, 'linewidth', 3);
legend([referenceLine,approximationLine],'Target', 'Predicted');
title('Comparsion')
hold off;
```