

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

Лабораторная работа № 7
по курсу «Нейроинформатика»
Тема: Автоассоциативные сети с узким горлом

Студент: Куликов А.В.
Группа: М80-408Б-17
Преподаватель: Аносова Н.П.
Дата: 25 декабря 2020
Оценка:

Цель работы: исследование свойств автоассоциативных сетей с узким горлом, алгоритмов обучения, а также применение сетей для выполнения линейного и нелинейного анализа главных компонент набора данных.

Основные этапы работы:

1. Использовать автоассоциативную сеть с узким горлом для отображения набора данных, выделяя первую главную компоненту данных.
2. Использовать автоассоциативную сеть с узким горлом для аппроксимации кривой на плоскости, выделяя первую нелинейную главную компоненту данных.
3. Применить автоассоциативную сеть с узким горлом для аппроксимации пространственной кривой, выделяя старшие нелинейные главные компоненты данных.

Оборудование:

Процессор: AMD Ryzen 5 Mobile 3550H

Объем оперативной памяти: 8 Гб

Программное обеспечение:

Python 3.8.5, MATLAB r2020

Сценарий выполнения работы:

Задание №1

```
clear;
clc;

% Создание обучающего множества
t = 0 : 0.025 : 2 * pi;

a = 0.7;
b = 0.7;
alpha = -pi / 6;
x0 = 0;
y0 = -0.1;

x = ellipse(t, a, b, alpha, y0, x0);
x_seq = con2seq(x);

% Создание и конфигурация сети
net = feedforwardnet(1, 'trainlm');
net.layers{1}.transferFcn = 'purelin';

net = configure(net, x_seq, x_seq);
net = init(net);

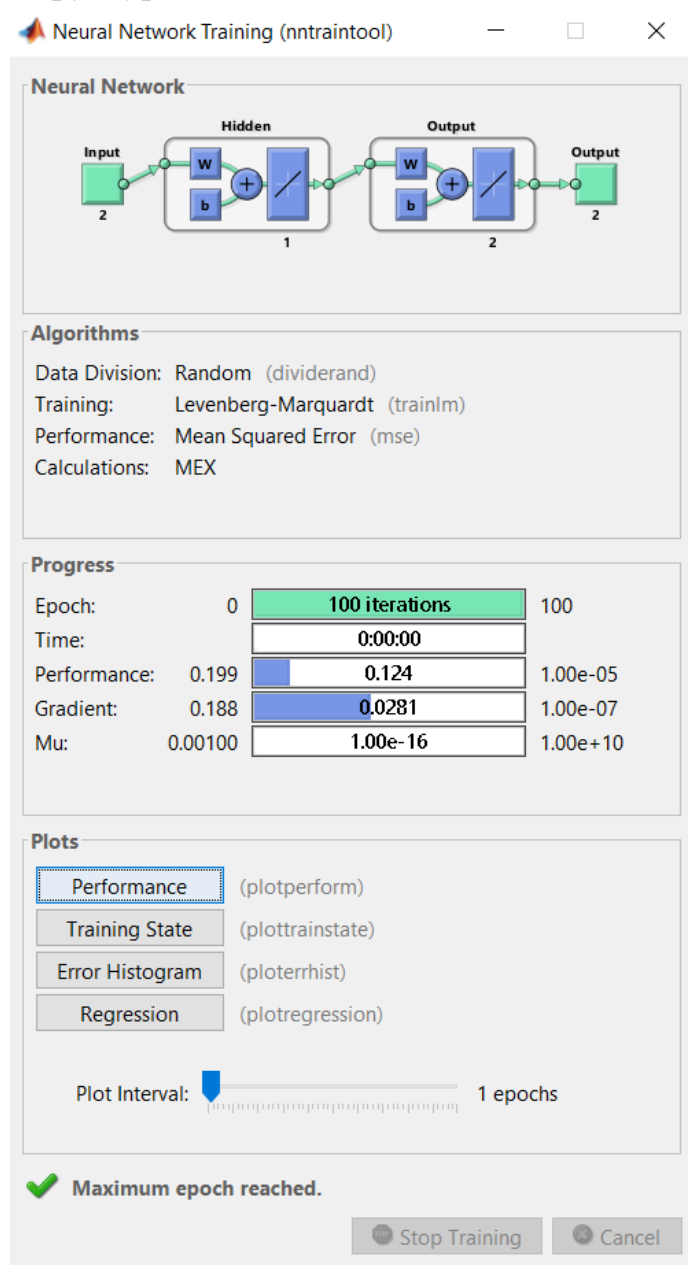
net.trainParam.epochs = 100;
net.trainParam.goal = 1.0e-5;

% Обучение сети
net = train(net, x_seq, x_seq);
```

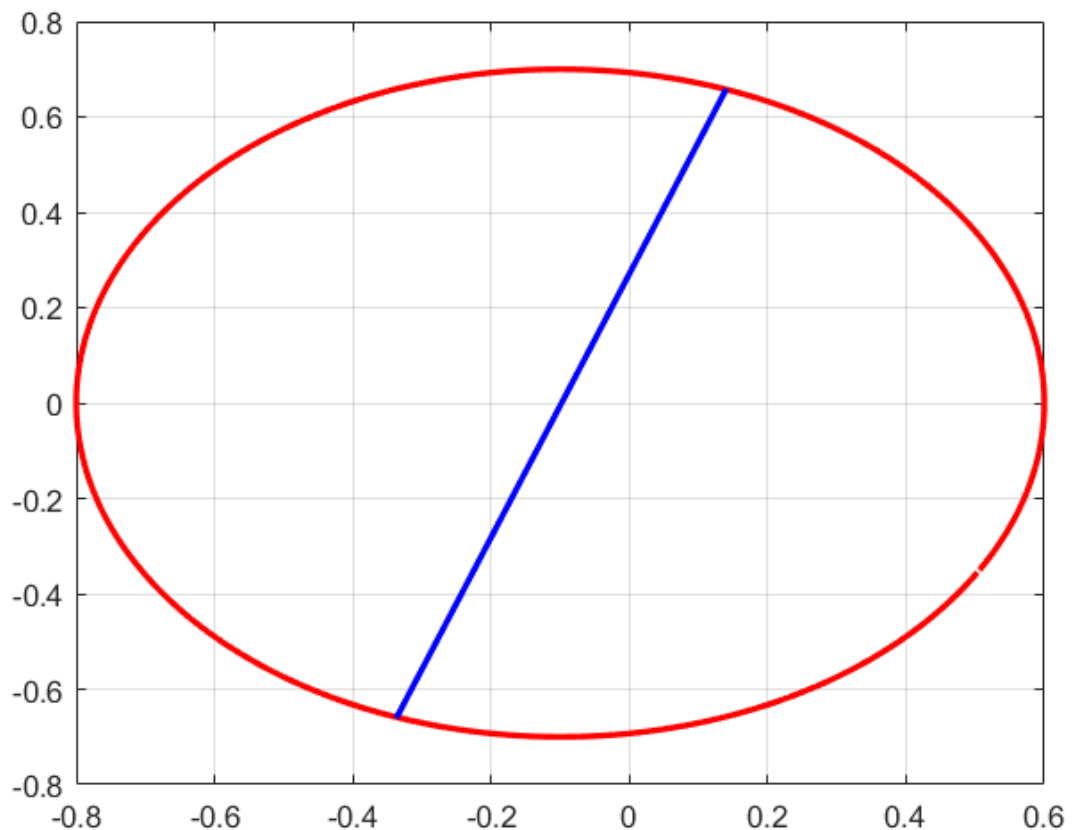
```
% Расчет выхода сети
y_pred_seq = sim(net, x_seq);
result = cell2mat(y_pred_seq);

% Отображение обучающего множества и выхода сети
plot(x(1, :), x(2, :), '-r', 'LineWidth', 2);
hold on;
grid on;
plot(result(1, :), result(2, :), '-b', 'LineWidth', 2);
```

Структура сети



Результат обучения сети и выход сети для тестового множества.



Красная линия – обучающее множество. Синяя – выход сети для тестового множества.

Задание №2

```
clear;
clc;

% Создание обучающего множества
phi = 0 : 0.025 : 2 * pi;
r = 2 * phi;

x = [r .* cos(phi); r .* sin(phi)];
x_seq = con2seq(x);

% Создание и конфигурация сети
net = feedforwardnet([10 1 10], 'trainlm');
net.layers{1}.transferFcn = 'tansig';
net.layers{2}.transferFcn = 'tansig';
net.layers{3}.transferFcn = 'tansig';
net.layers{4}.transferFcn = 'purelin';

net = configure(net, x_seq, x_seq);
net = init(net);

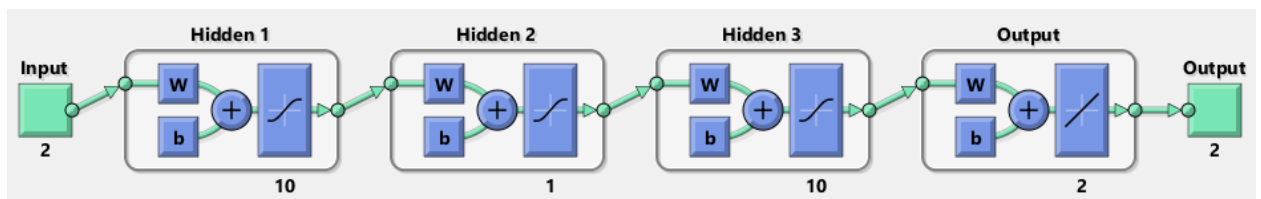
net.trainParam.epochs = 2000;
net.trainParam.goal = 1.0e-5;
```

```
% Обучение сети
net = train(net, x_seq, x_seq);

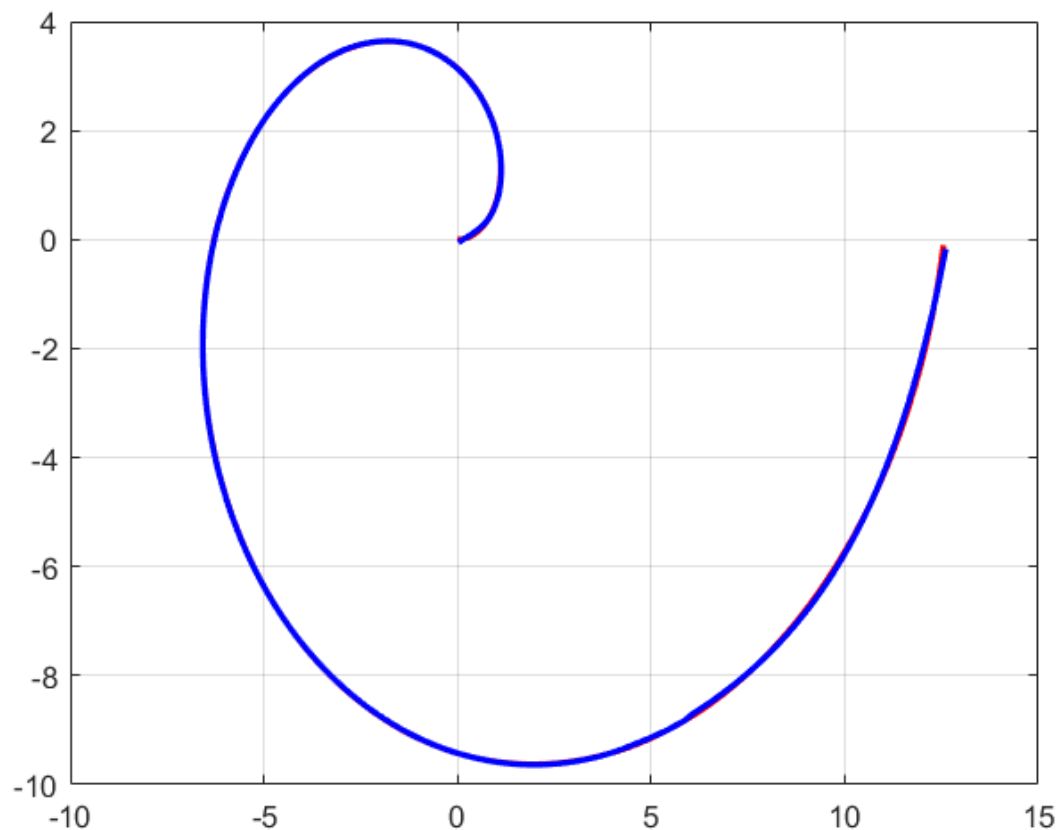
% Расчет выхода сети
y_pred_seq = sim(net, x_seq);
result = cell2mat(y_pred_seq);

% Отображение обучающего множества и выхода сети
plot(x(1, :), x(2, :), '-r', 'LineWidth', 2);
hold on;
grid on;
plot(result(1, :), result(2, :), '-b', 'LineWidth', 2);
```

Структура сети



Выход сети:



Красная линия – обучающее множество. Синяя – выход сети для тестового множества.

Задание №3

```
clear;
clc;

% Создание обучающего множества
phi = 0 : 0.025 : 2 * pi;
r = 2 * phi;

x = [r .* cos(phi); r .* sin(phi); phi];
x_seq = con2seq(x);

% Создание и конфигурация сети
net = feedforwardnet([10 2 10], 'trainlm');
net.layers{1}.transferFcn = 'tansig';
net.layers{2}.transferFcn = 'tansig';
net.layers{3}.transferFcn = 'tansig';
net.layers{4}.transferFcn = 'purelin';

net = configure(net, x_seq, x_seq);
net = init(net);

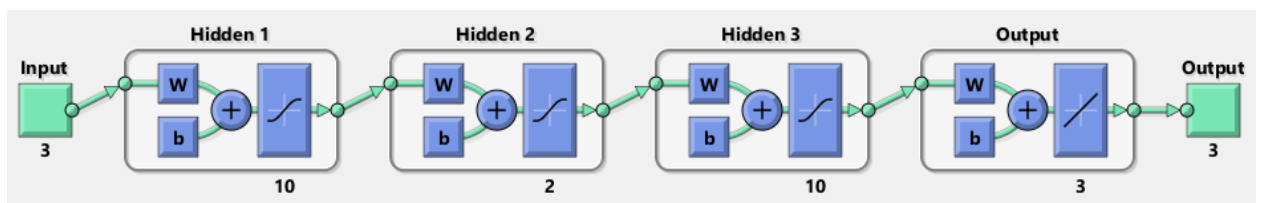
net.trainParam.epochs = 1000;
net.trainParam.goal = 1.0e-5;

% Обучение сети
net = train(net, x_seq, x_seq);

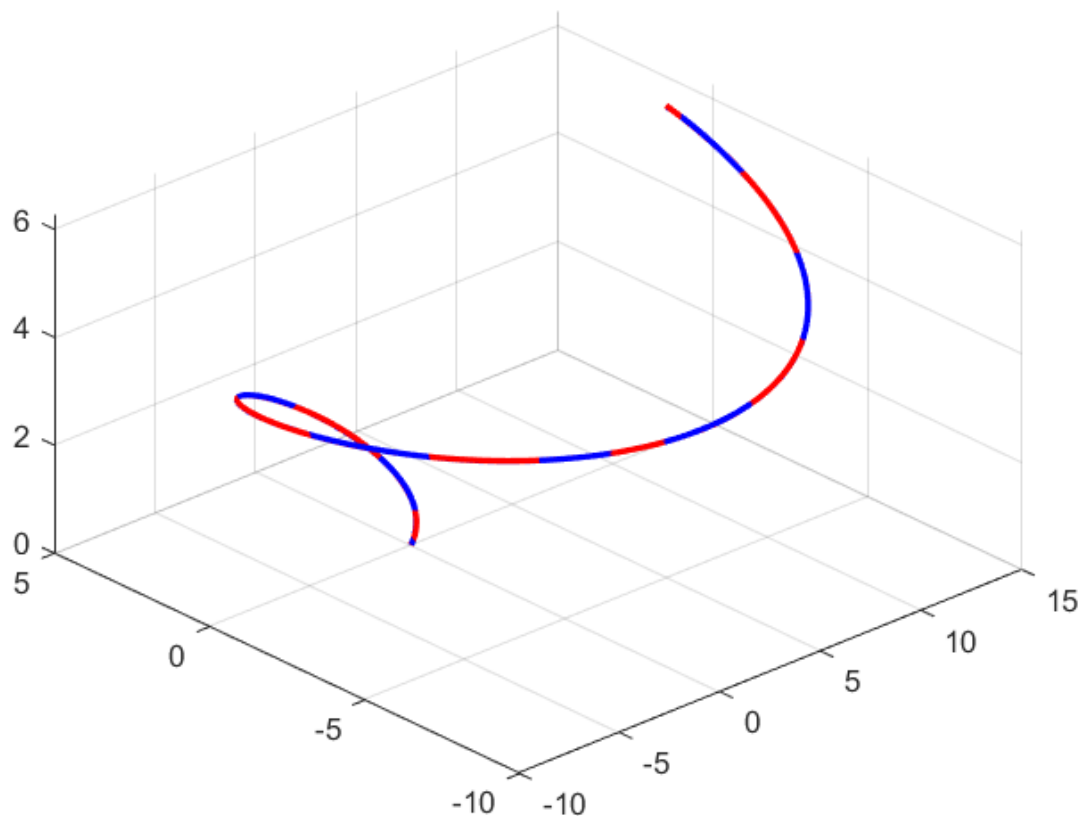
% Расчет выхода сети
y_pred_seq = sim(net, x_seq);
result = cell2mat(y_pred_seq);

% Отображение обучающего множества и выхода сети
plot3(x(1, :), x(2, :), x(3, :), '-r', 'LineWidth', 2);
hold on;
grid on;
plot3(result(1, :), result(2, :), result(3, :), '-b', 'LineWidth', 2);
```

Структура сети



Выход сети



Выводы:

Автоассоциативная сеть -- многослойная нейронная сеть прямого распространения сигнала, обученная выдавать входные данные на выходе. В обучении автоассоциативной сети "учителем" является сама входная информация. Такая сеть имеет один или более скрытых слоев меньшей размерности, которые выделяют наиболее значимые признаки во входной информации. Автоассоциативные сети с таким слоем оказываются полезны при решении задач обработки данных высокой размерности, так как позволяют сократить объем данных.

Т.о. автоассоциативные сети с узким горлом позволяют решать задачу понижения размерности данных с минимальной потерей информации, что в свою очередь может применяться в алгоритмах сжатия данных.