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

Лабораторная работа № 7

по курсу «Нейроинформатика»

Тема: Ассоциативные сети с узким горлом.

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

Москва, 2021

Постановка задачи

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

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. Использовать автоассоциативную сеть с узким горлом для аппроксимации пространственной кривой, выделяя старшие нелинейные главные компоненты данных.

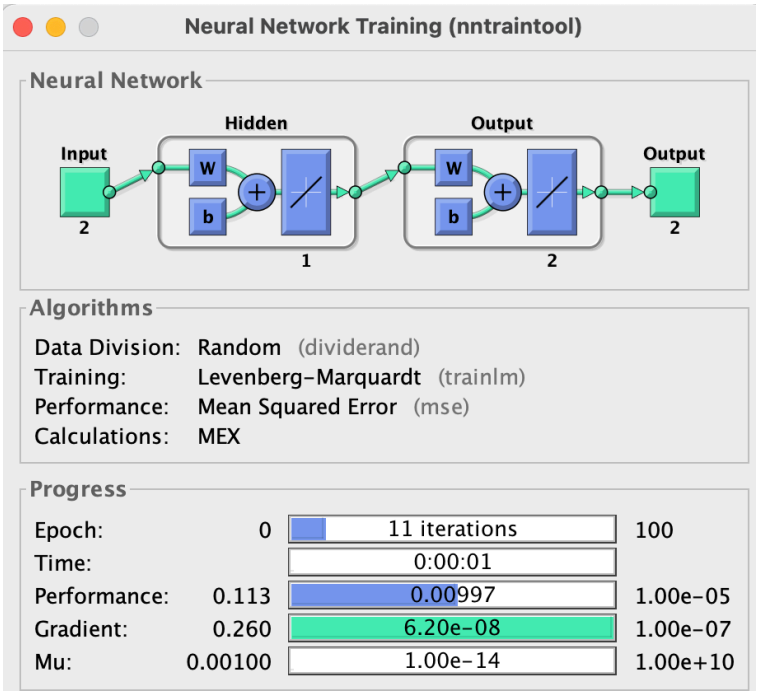
Вариант 6:

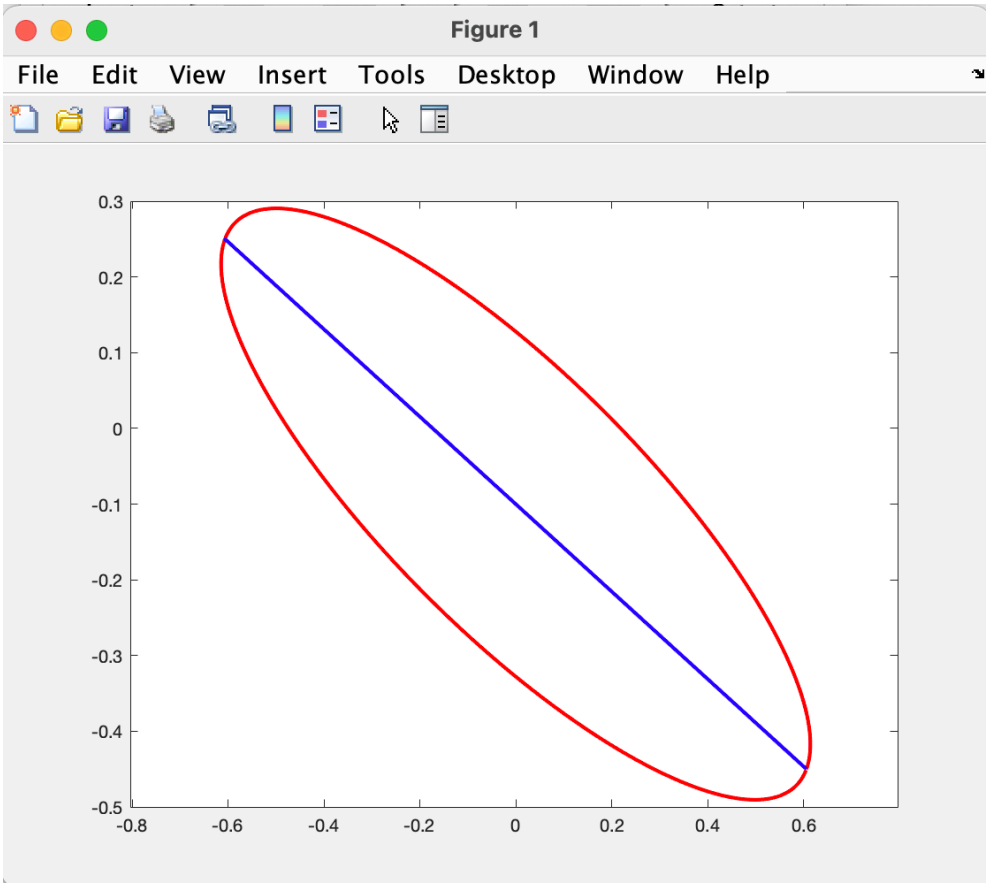
6.	Эллипс: $a = 0.7, b = 0.2, \alpha = -\pi/6, x_0 = 0, y_0 = -0.1$

$$r = \varphi^2;$$

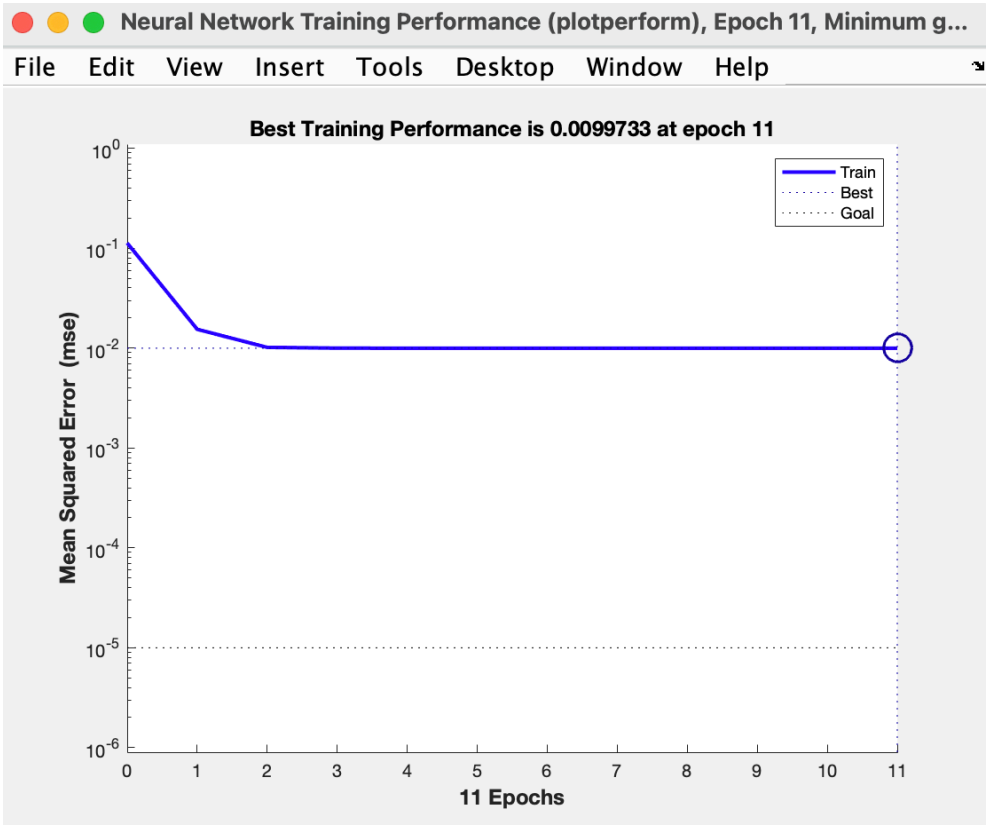
Ход работы

Выявление первой главной компоненты

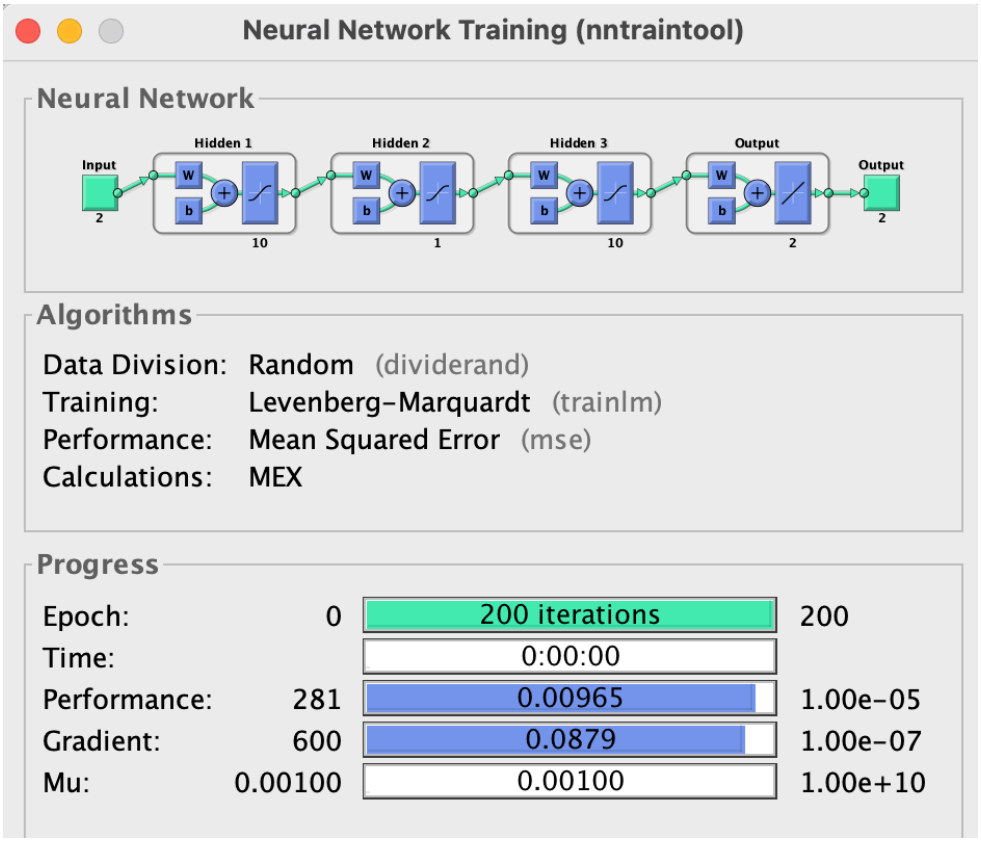


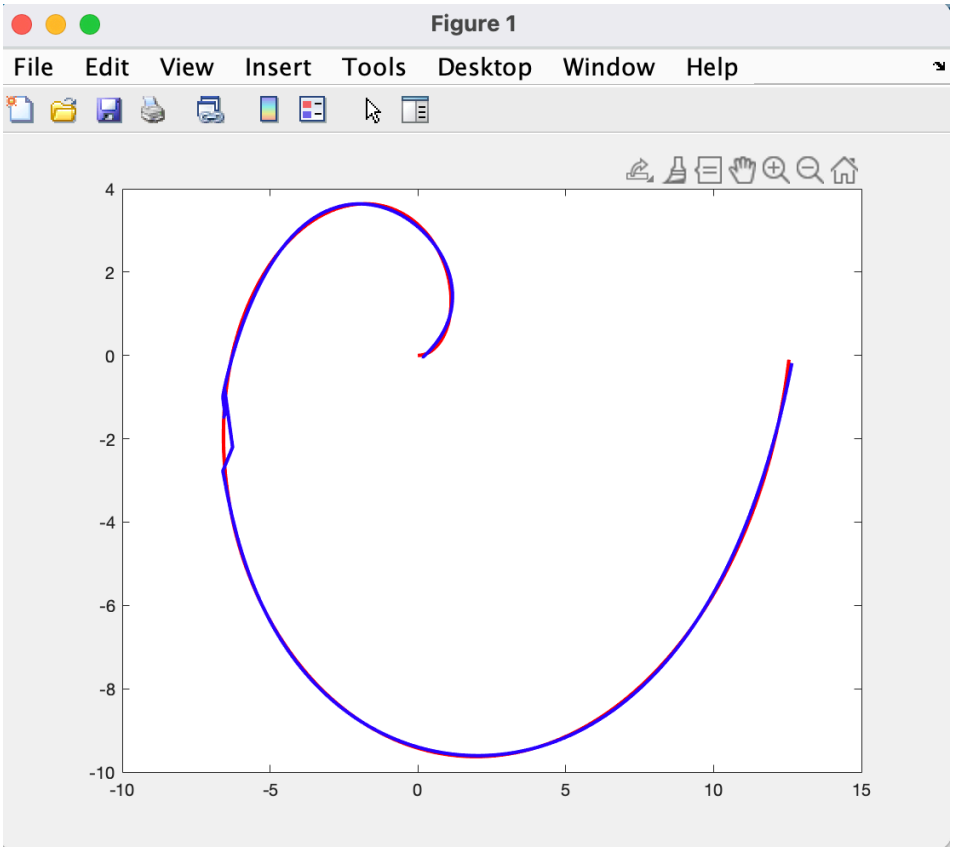


Performance

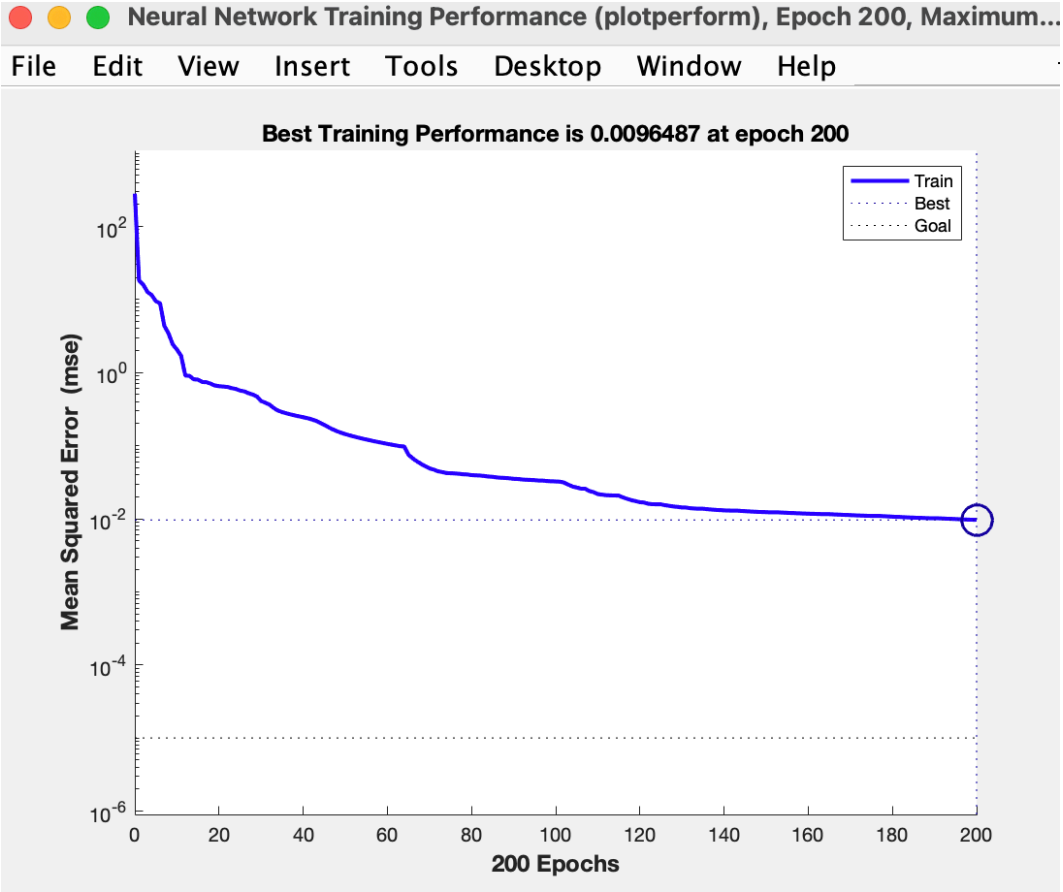


Выявление первой нелинейной главной компоненты

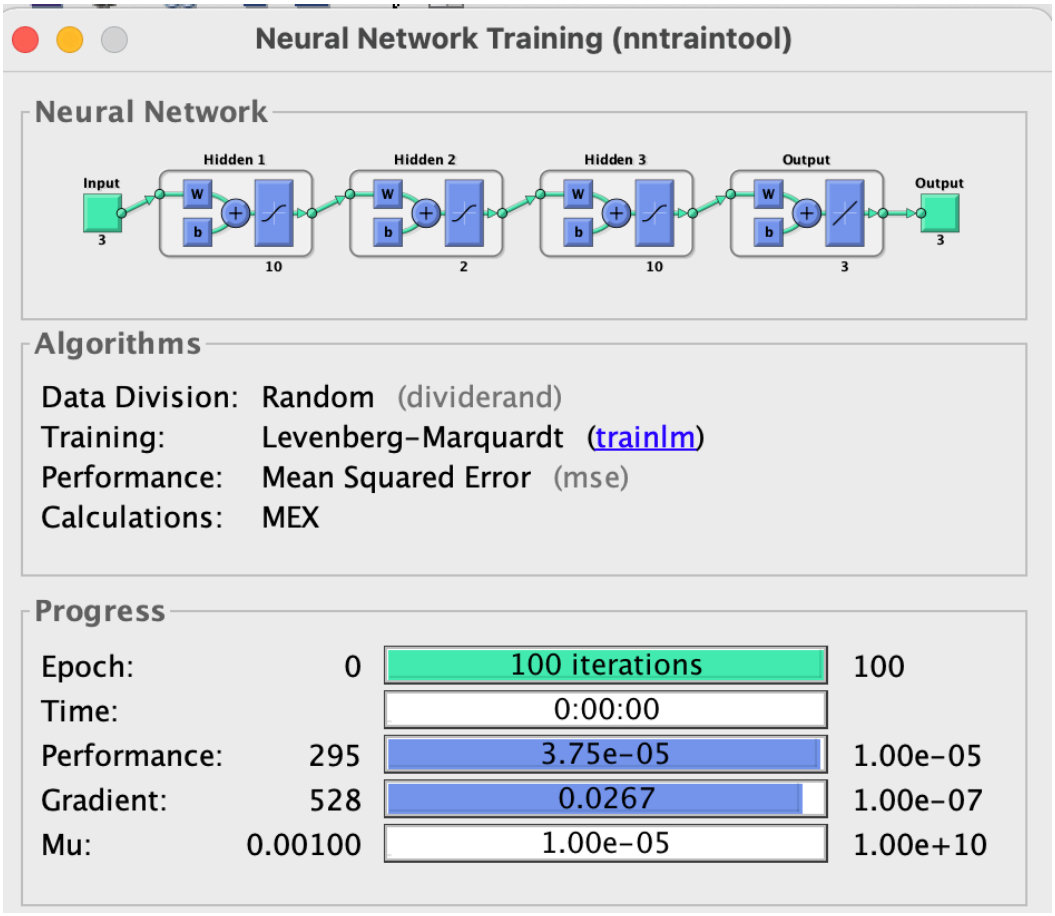


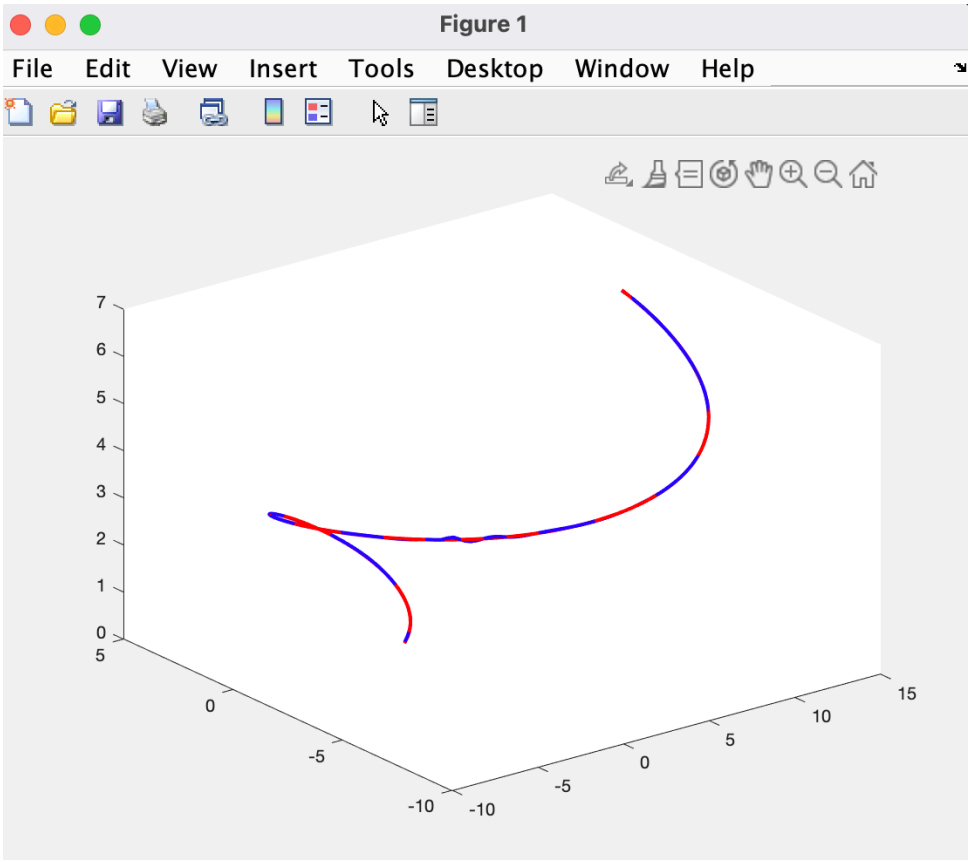


Performance

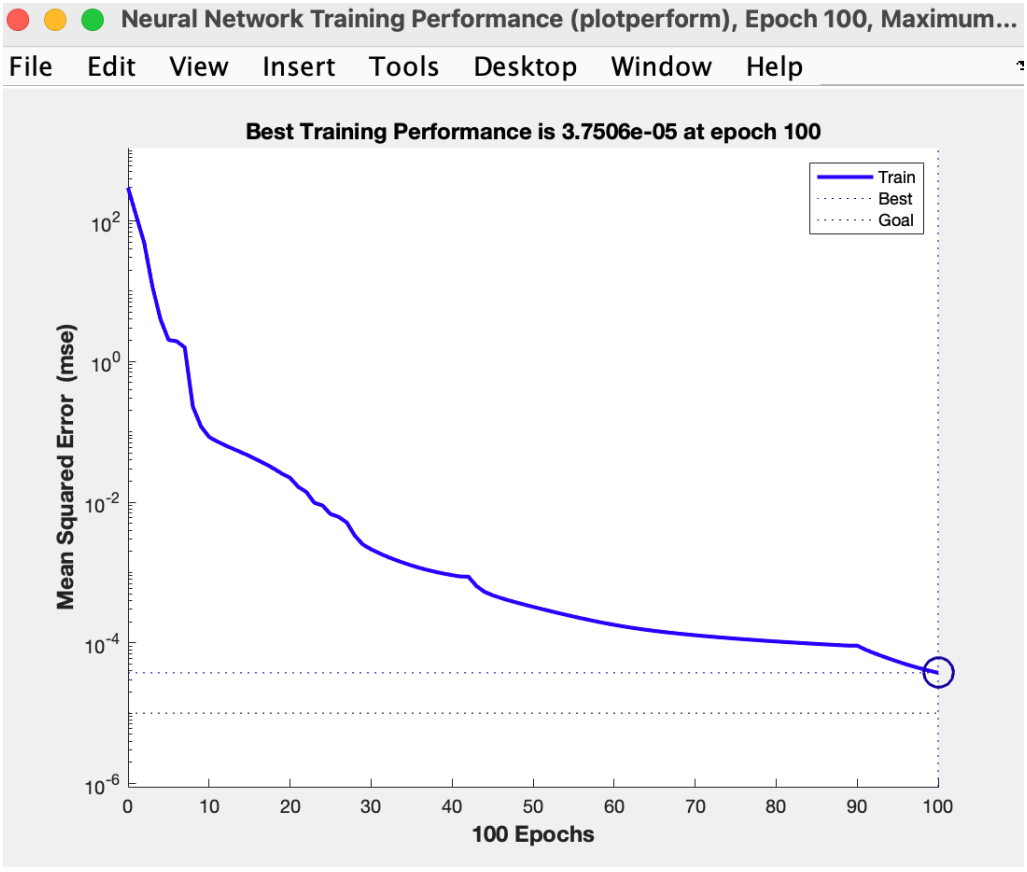


Выявление старших нелинейных главных компонент





Performance



```
% 1.1
% Генерация обучающего множества
trange = 0 : 0.025 : 2 * pi;
x = ellipse(trange, 0.7, 0.2, 0, -0.1, -pi / 6);
xseq = con2seq(x);

plot(x(1, :), x(2, :), '-r', 'LineWidth', 2);

% 1.2
% Создание сети
net = feedforwardnet(1, 'trainlm');
net.layers{1}.transferFcn = 'purelin';

net = configure(net, xseq, xseq);
net = init(net);
view(net);

% 1.4
% Параметры обучения
net.trainParam.epochs = 100;
net.trainParam.goal = 1.0e-5;

% 1.5
% Обучение
net = train(net, xseq, xseq);
display(net);

% 1.7
% Выход сети
yseq = sim(net, xseq);
y = cell2mat(yseq);

% 1.8
% Отображение обучающего множества
plot(x(1, :), x(2, :), '-r', y(1, :), y(2, :), '-b',
```

```
% 2.1
% Генерация обучающего множества
phi = 0 : 0.025 : 2 * pi;
r = phi.*2;
x = [r .* cos(phi); r .* sin(phi)];
xseq = con2seq(x);

plot(x(1, :), x(2, :), '-r', 'LineWidth', 2);

% 2.2
% Создание многослойной сети прямого распространения
net = feedforwardnet([10 1 10], 'trainlm');
net = configure(net, xseq, xseq);

% 2.3
% Инициализация весовых коэффициентов
net = init(net);
view(net);

% 2.4
% Параметры обучения
net.trainParam.epochs = 200;
net.trainParam.goal = 1.0e-5;

% 2.5
% Обучение
net = train(net, xseq, xseq);

% 2.6
% Выход сети
yseq = sim(net, xseq);
y = cell2mat(yseq);

% 2.7
% Отображение обучающего множества
plot(x(1, :), x(2, :), '-r', y(1, :), y(2, :), '-b', 'LineWidth', 2);
```

```
% 3.1
% Генерация обучающего множества
phi = 0 : 0.025 : 2 * pi;
r = phi.*2;
x = [r .* cos(phi); r .* sin(phi); phi];
xseq = con2seq(x);

plot3(x(1, :), x(2, :), x(3, :), '-r', 'LineWidth', 2);

% 3.2
% Создание сети
net = feedforwardnet([10 2 10], 'trainlm');
net = configure(net, xseq, xseq);

% 3.3
% Инициализация весовых коэффициентов
net = init(net);
view(net);

% 3.4
% Параметры обучения
net.trainParam.epochs = 100;
net.trainParam.goal = 1.0e-5;

% 3.5
% Обучение сети
net = train(net, xseq, xseq);

% 3.7
% Выход сети
yseq = sim(net, xseq);
y = cell2mat(yseq);

% 3.8
% Отображение обучающего множества
plot3(x(1, :), x(2, :), x(3, :), '-r', y(1, :), y(2, :), y(3, :), '-b', 'LineWidth', 2);
```

Вывод

Выполнив лабораторную работу, я научился применять автоассоциативную сеть с узким горлом для аппроксимации функций и отображения данных, выделяя линейные и нелинейные компоненты данных.

Итеративную автоассоциативную сеть с узким горлом используют по двум причинам:

Нейронные алгоритмы легко обобщаются на случай нелинейного сжатия информации, когда никаких явных решений уже не существует. Также не запрещается заменять линейные нейроны - нелинейными.

Иногда обучение необходимо проводить в режиме «онлайн», т. е. уметь адаптироваться к потоку данных. Примером может служить борьба с нестационарными помехами в каналах связи. Итерационные методы идеально подходят в этой ситуации, когда нет возможности собрать воедино весь набор примеров и произвести необходимые матричные операции над ним.