

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

**Лабораторная работа № 6**  
по курсу «Нейроинформатика»  
Тема: Сети Кохонена.

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

Москва, 2020

# Постановка задачи

Исследование свойств слоя Кохонена, карты Кохонена, а также сетей векторного квантования, обучаемых с учителем, алгоритмов обучения, а также применение сетей в задачах классификации и кластеризации.

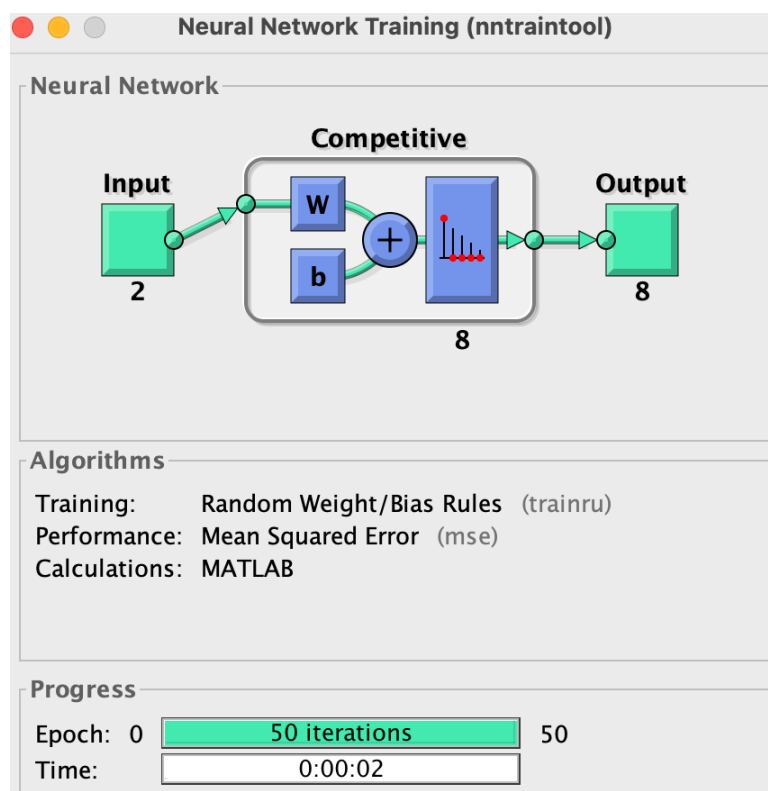
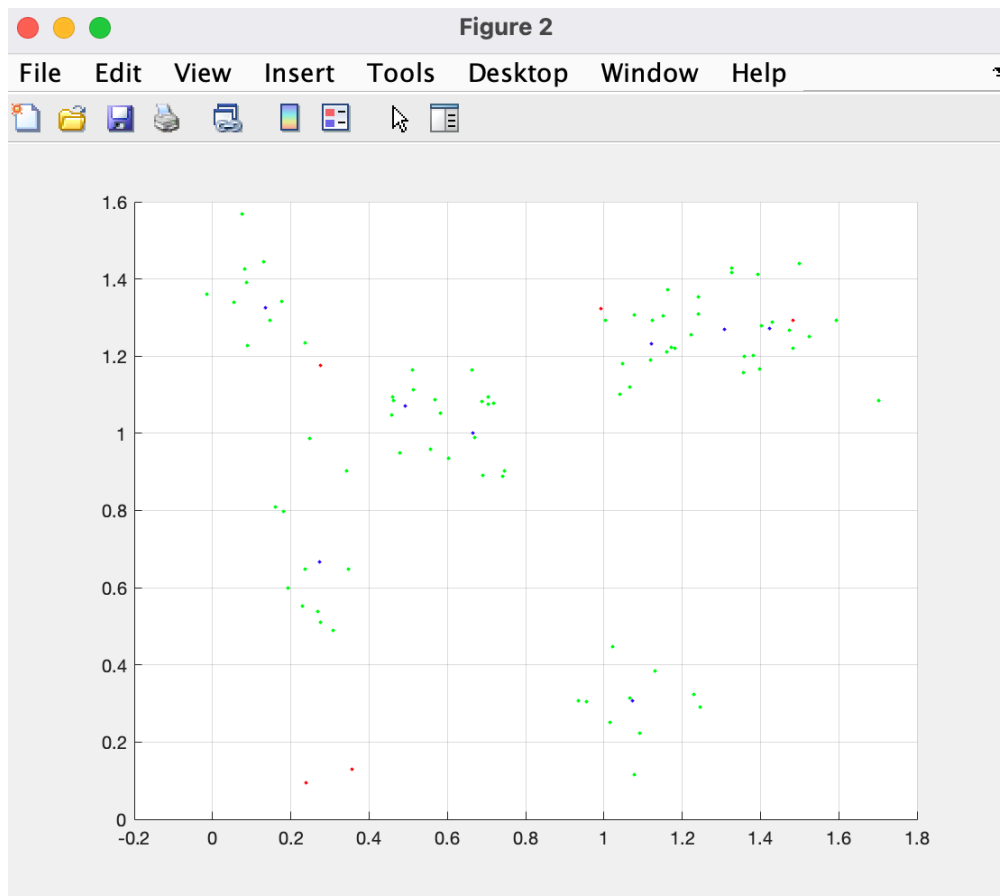
- 1. Использовать слой Кохонена для выполнения кластеризации множества точек.
- 2. Использовать карту Кохонена для выполнения кластеризации множества точек.
- 3. Использовать карту Кохонена для нахождения одного из решений задачи коммивояжёра.
- 4. Использовать сети векторного квантования, обучаемой с учителем, для классификации точек в случае, когда классы не являются нелинейно разделимыми

Вариант 6

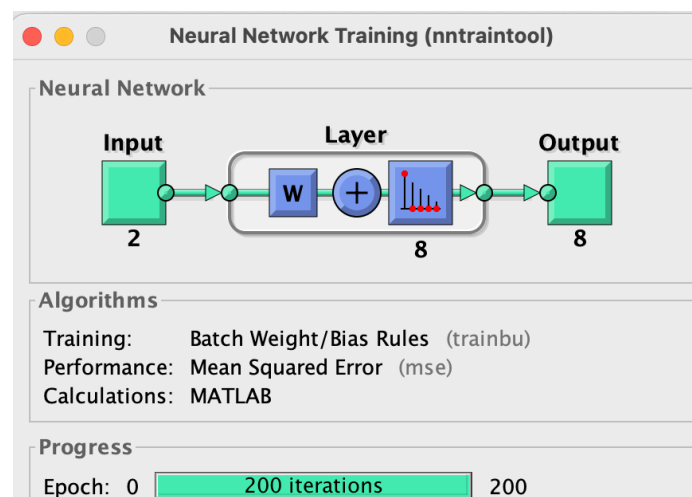
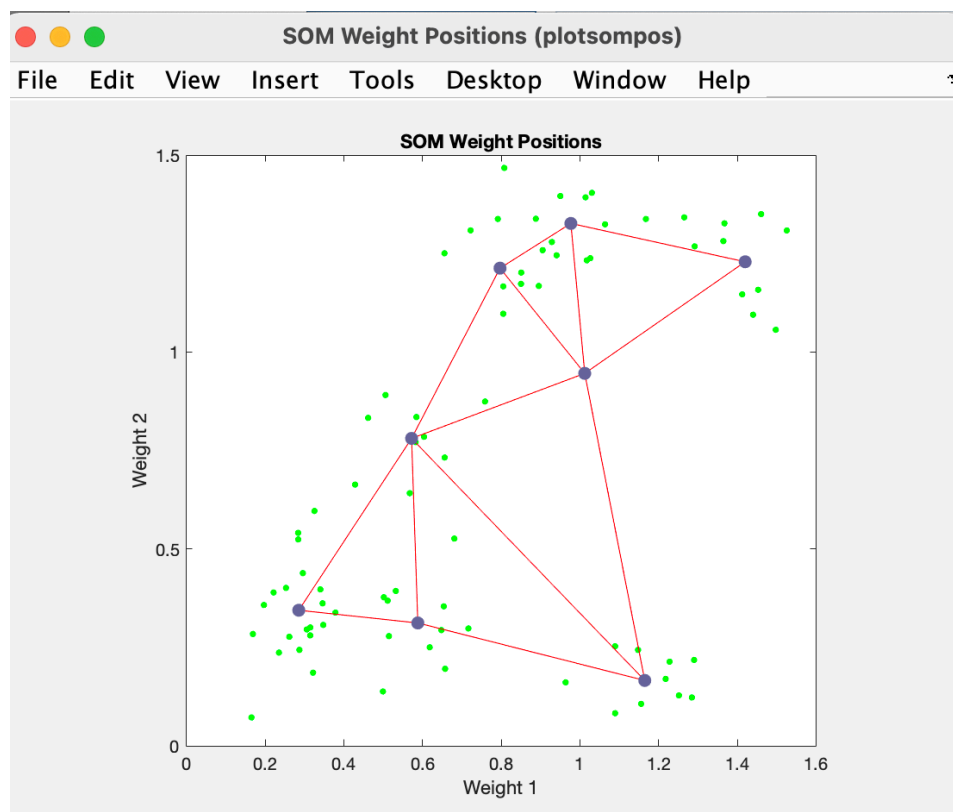
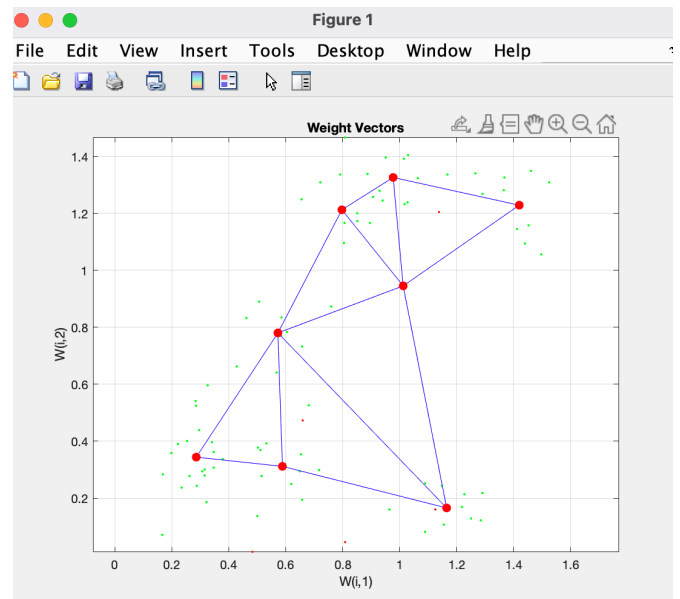
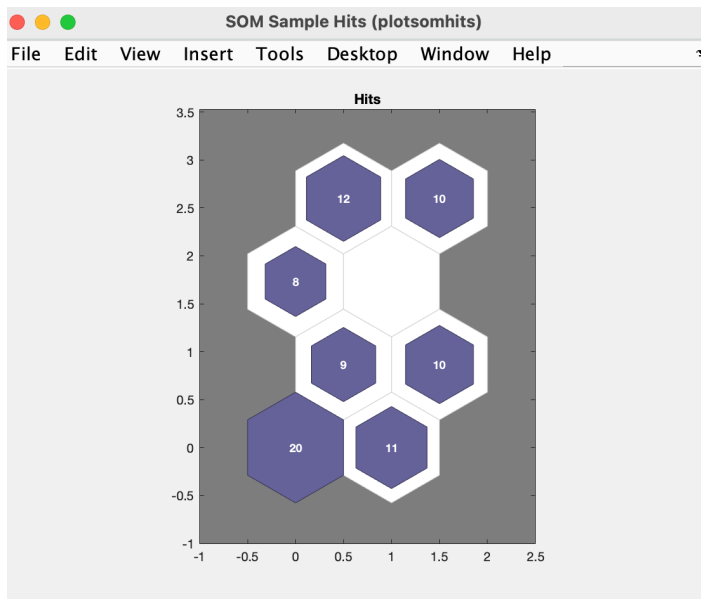
6.	$\begin{bmatrix} 0.5 & 0.7 & 0.4 & 0.6 & -0.7 & -1.3 & 0.5 & 1.3 & -0.2 & 0.7 & -1 & -0.2 \\ 0.7 & -0.4 & -1 & -1.5 & -1.4 & 0.9 & -0.6 & -1.4 & -0.4 & 0.8 & -0.1 & 0.4 \end{bmatrix}$ $\begin{bmatrix} -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$
----	--

# Работа программы

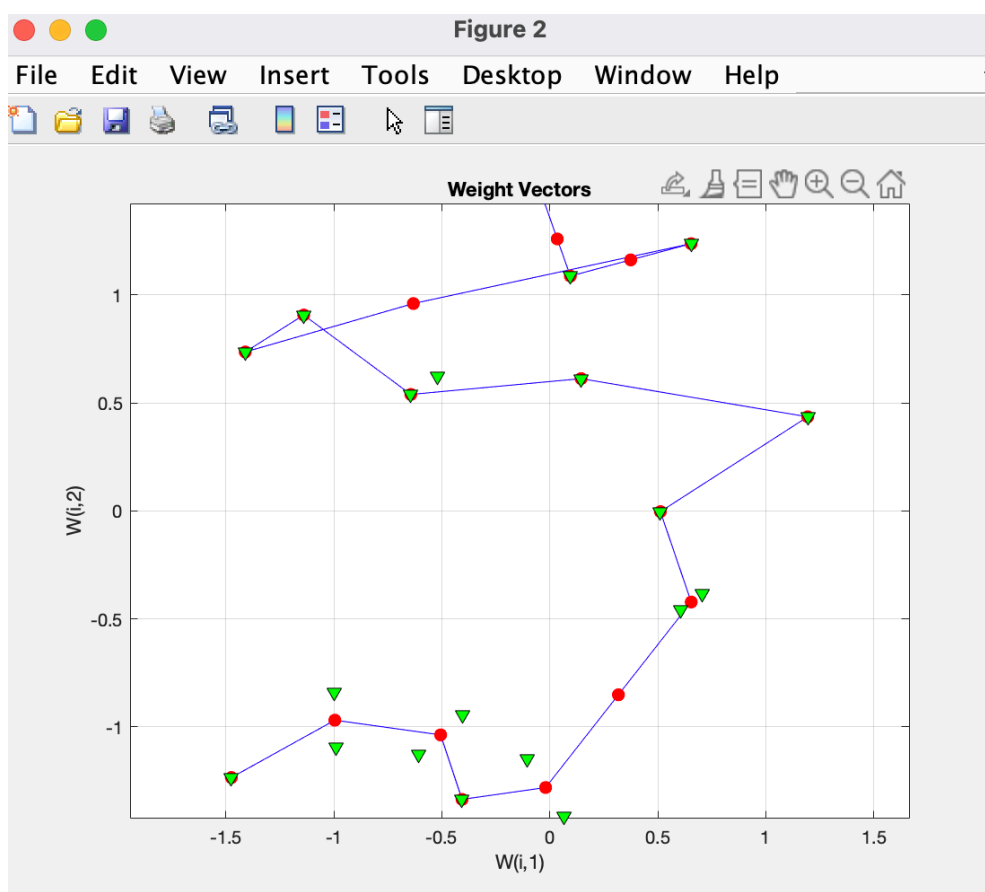
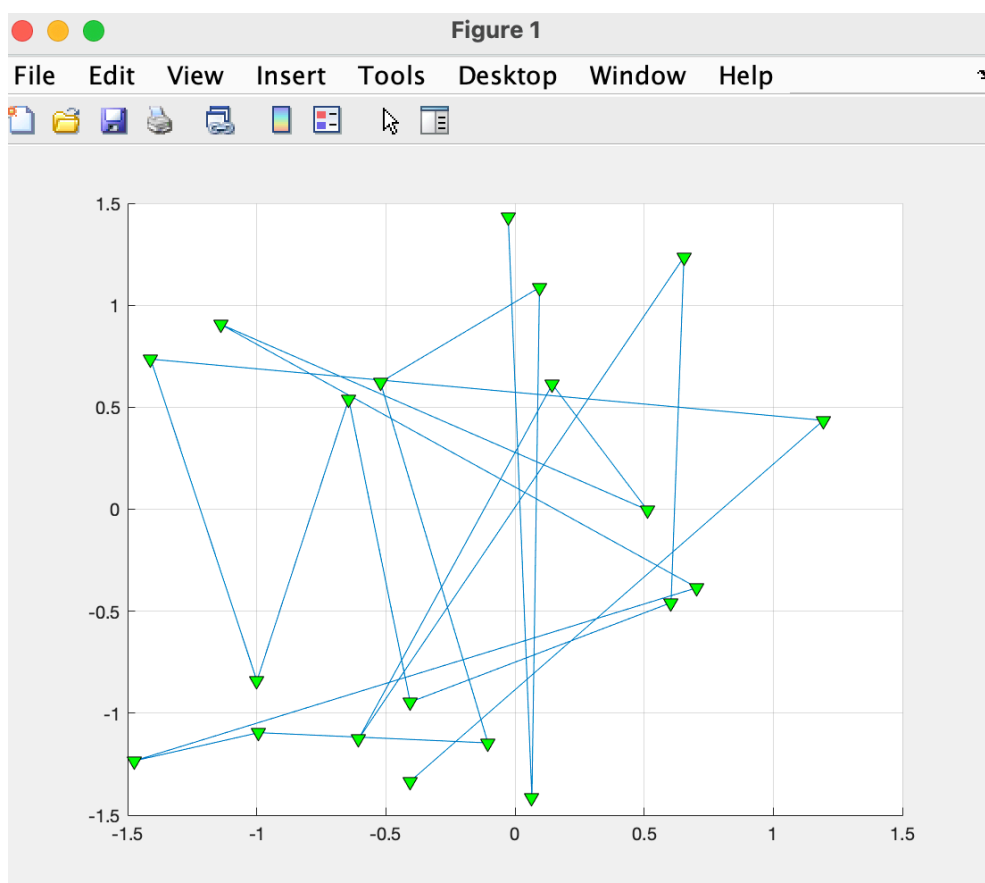
## Кластеризация слоем Кохонена

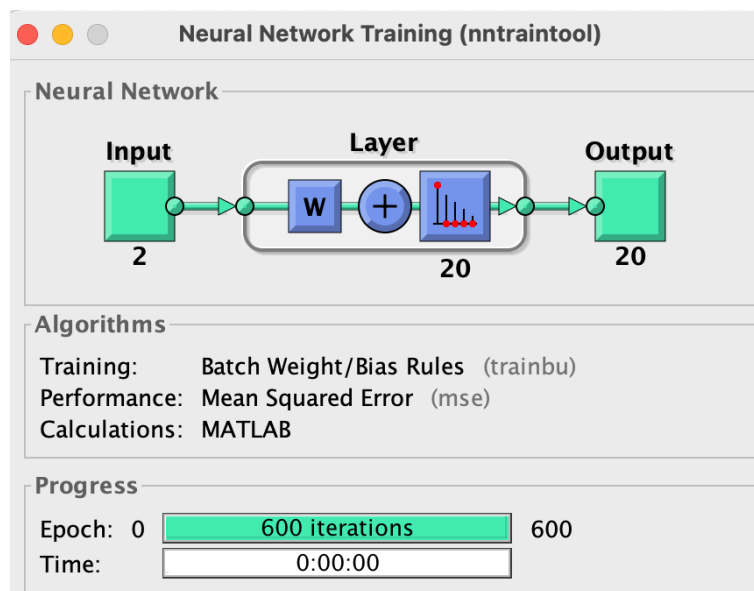


# Кластеризация при помощи SOM

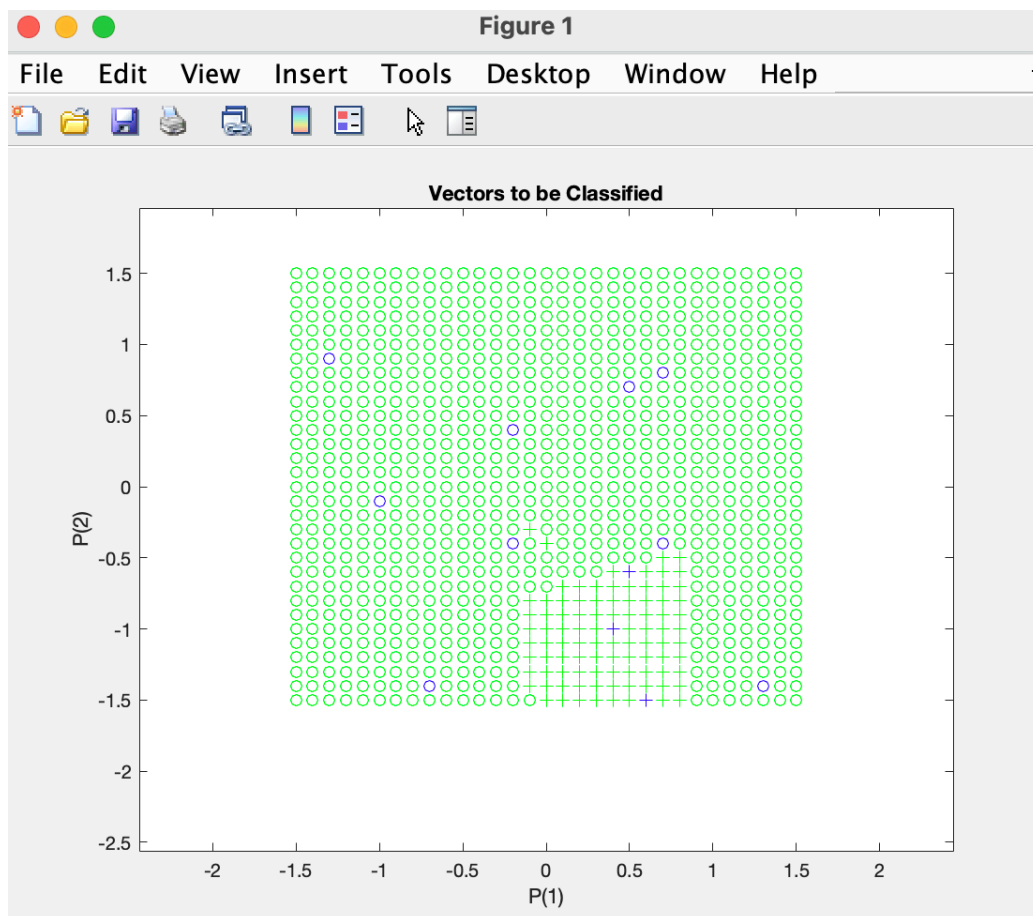


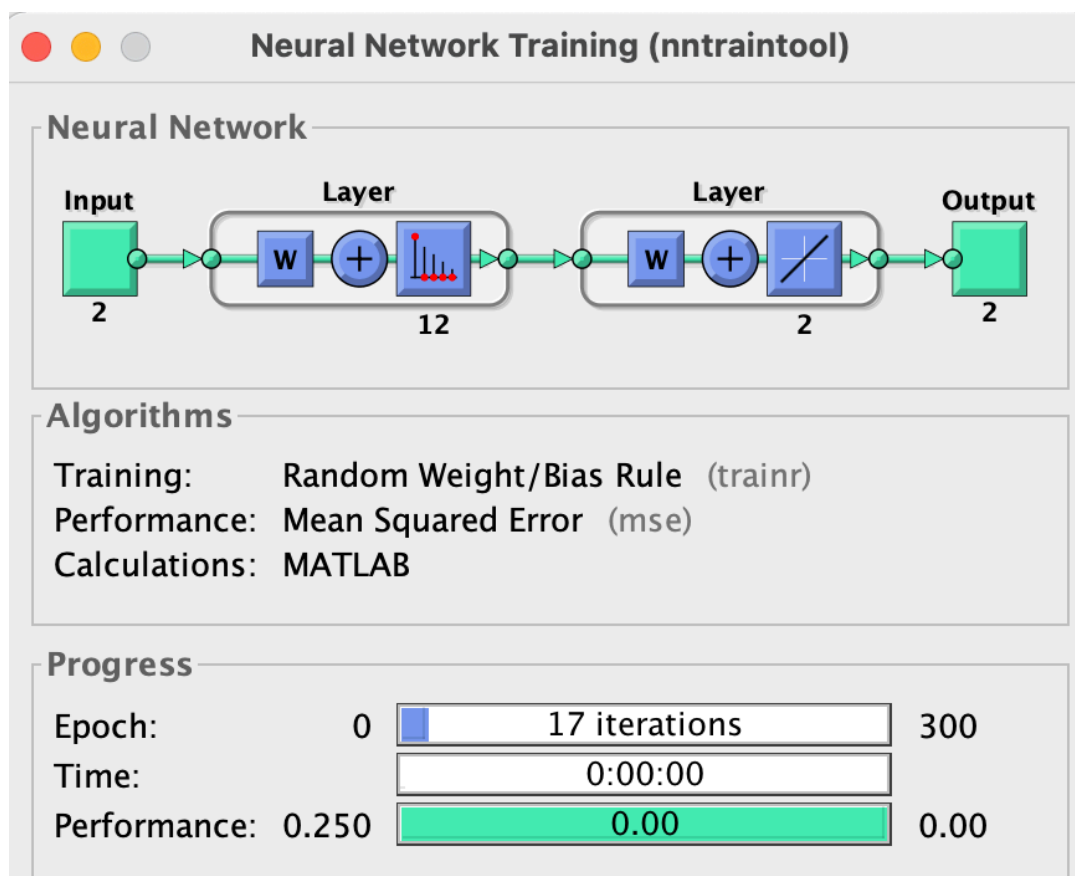
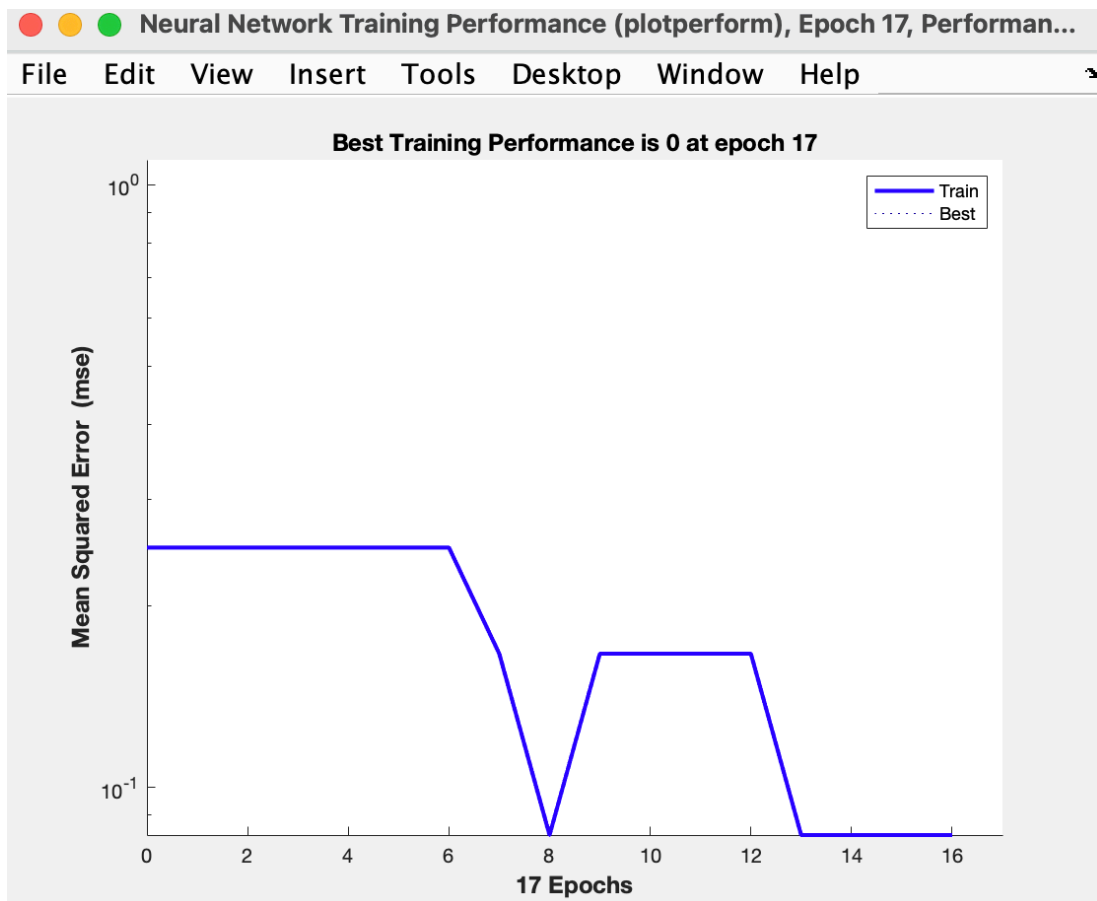
## Решение задачи коммивояжёра





## Классификация точек





```

%1.1
% Формируем множество случайных точек в 8 классов
X = [0 1.5;
     0 1.5];
clusters = 8;
points = 10;
deviation = 0.1;
P = nngenc(X, clusters, points, deviation);

figure;
hold on;
grid on;
scatter(P(1, :), P(2, :), 5, [0 1 0], 'filled');

% 1.3
% Создаем сеть
net = competlayer(8);
net = configure(net, P);
view(net);
net.divideFcn = '';

% 1.4
% Обучаем
net.trainParam.epochs = 50;
net = train(net, P);

% 1.6
% Проверяем качество разбиения
R = zeros(2, 5) + 1.5 * rand(2, 5);
res = vec2ind(sim(net, R));

figure;
hold on;
grid on;
scatter(P(1, :), P(2, :), 5, [0 1 0], 'filled');
scatter(net.IW{1}(:, 1), net.IW{1}(:, 2), 5, [0 0 1]
scatter(R(1, :), R(2, :), 5, [1 0 0], 'filled');

```

```

% 2.1
% Формируем множество случайных точек в 8 классов
X = [0 1.5;
     0 1.5];
clusters = 8;
points = 10;
deviation = 0.1;
P = nngenc(X, clusters, points, deviation);

% 2.2
% Создаем сеть
net = newsom(X, [2 4]);
net.trainParam.epochs = 300;

net = selforgmap([2 4], 'topologyFcn', 'hextop', 'distanceFcn', 'linkdist');

net = configure(net, X);

% 2.3
% Обучаем
net.divideFcn = '';
net = train(net, P);

figure(5)
plotsomhits(net, P)
figure(6)
plotsompos(net, P)

% 2.5
% Проверяем качество разбиения
R = zeros(2, 5) + 1.5 * rand(2, 5);
res = vec2ind(sim(net, R));

figure;
hold on;
grid on;
scatter(P(1, :), P(2, :), 5, [0 1 0], 'filled');
scatter(net.IW{1}(:, 1), net.IW{1}(:, 2), 5, [0 0 1], 'filled');
scatter(R(1, :), R(2, :), 5, [1 0 0], 'filled');
plotsom(net.IW{1, 1}, net.layers{1}.distances);

```

```

%3.1
N = 20;
T = -1.5 * ones(2, N) + 3 * rand(2, N);

figure;
hold on;
grid on;
plot(T(1,:), T(2,:), '-V', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'g', 'MarkerSize', 7);

% 3.3
% Создаем и обучаем
net = selforgmap(N);
net = configure(net, T);
view(net);
net.divideFcn = '';
net.trainParam.epochs = 600;
net = train(net, T);

% 3.4
% Координаты городов и центры кластеров
figure;
hold on;
grid on;
plotsom(net.IW{1,1}, net.layers{1}.distances);
plot(T(1,:), T(2,:), 'V', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'g', 'MarkerSize', 7);
|

```



```

% Вариант 6
P = [0.5 0.7 0.4 0.6 -0.7 -1.3 0.5 1.3 -0.2 0.7 -1 -0.2
      0.7 -0.4 -1 -1.5 -1.4 0.9 -0.6 -1.4 -0.4 0.8 -0.1 0.4];
T = [-1 -1 1 1 -1 -1 1 -1 -1 -1 -1 -1];

Ti = T;
Ti(Ti == 1) = 2;
Ti(Ti == -1) = 1;
Ti = ind2vec(Ti);

% 4.3
% Создаем сеть
net = lvqnet(12, 0.1);
net = configure(net, P, Ti);
view(net)
net.IW{1,1};
net.LW{2,1};

% 4.4
% Обучаем
net.trainParam.epochs = 300;
net = train(net, P, Ti);

% 4.6
% Проверка качества обучения
[X,Y] = meshgrid([-1.5 : 0.1 : 1.5], [-1.5 : 0.1 : 1.5]);
res = sim(net, [X(:)'; Y(:)']);
res = vec2ind(res) - 1;
figure;
plotpv([X(:)'; Y(:)'], res);
point = findobj(gca,'type','line');
set(point,'Color','g');
hold on;
plotpv(P, max(0, T));

```

## Вывод

Самоорганизующиеся карты Кохонена служат, в первую очередь, для визуализации и первоначального анализа данных. Каждая точка данных отображается соответствующим кодовым вектором из многомерной сетки. Так получают представление данных на плоскости - «Карту данных».

Из минусов слоев Кохонена стоит отметить то, что нам нужно заранее знать количество кластеров. Получается, мы сильно зависим от этой величины. Также не совсем очевидно, что делать в ситуации, когда точка одинаково удалена до нескольких центров классов?

Несмотря на некоторые недостатки сетей Кохонена, люди до сих пор применяют их для решения узкоспециализированных задач «Нейроинформатики».