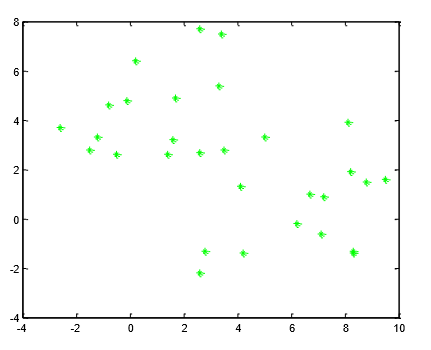
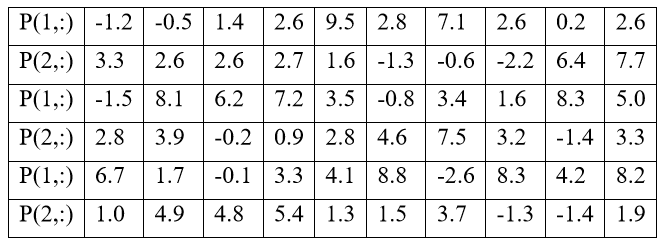
**Определение центров кластеризации при помощи самоорганизующихся слоя и карты Кохенена**

На графике даны точки, которые необходимо кластеризовать. Для этого необходимо определить координаты точек (оцифровать данные) и образовать входную матрицу

Визуально изучая график можно прийти к выводу, что это множество можно разбить на 6 кластеров ( можно и на 7, как нравится)

Входные данные (координаты точек)

P(1,:)=[-1.2 -0.5 1.4 2.6 9.5 2.8 7.1 2.6 0.2 2.6 -1.5 8.1 6.2 7.2 3.5 -0.8 3.4 1.6 8.3 5.0 6.7 1.7 -0.1 3.3 4.1 8.8 -2.6 8.3 4.2 8.2 ]

P(2,:)=[3.3 2.6 2.6 2.7 1.6 -1.3 -0.6 -2.2 6.4 7.7 2.8 3.9 -0.2 0.9 2.8 4.6 7.5 3.2 -1.4 3.3 1.0 4.9 4.8 5.4 1.3 1.5 3.7 -1.3 -1.4 1.9 ]

plot(P(1,:), P(2,:), '\*g');

title('Input vectors');

xlabel('P(1,:)');

ylabel('P(2,:)'); % Построить график входных данных и сравнить с данным. (Они должны быть

% идентичны)

nclusters = 6; % Количество кластеров

a1 = -3;

a2 = 10;

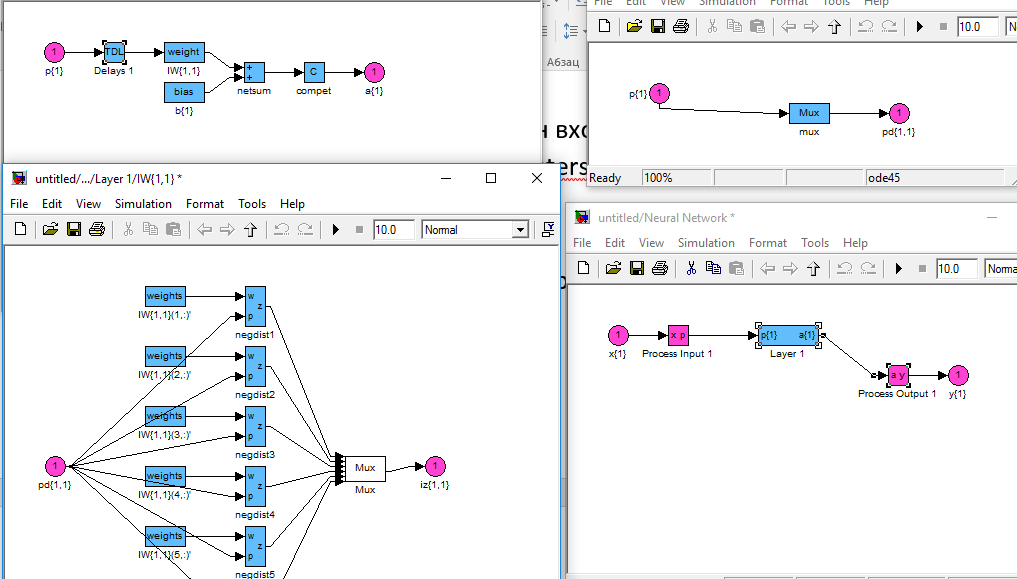
b1 = -3;

b2 = 9; % Диапазон входных данных

net = newc([a1 a2; b1 b2], nclusters, 0.1, 0.0005); % Создадим слой Кохенена уровни обученности

% установим, например, 0.1 и 0.0005

gensim(net) % Посмотрим архитектуру созданной сети



wo = net.IW{1} % Посмотрим начальные веса и смещения

wo =

3.5000 3.0000

3.5000 3.0000

3.5000 3.0000

3.5000 3.0000

3.5000 3.0000

3.5000 3.0000

bo = net.b{1}

bo =

16.3097

16.3097

16.3097

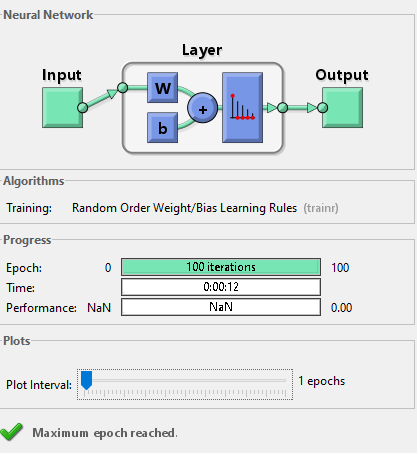
16.3097

16.3097

16.3097

net.trainParam.epochs=100; % Задать количество циклов обучения

net = train(net,P) % Провести обучение на входной последовательности



w = net.IW{1} % Веса и смещения после обучения

-1.4816 3.3600

4.4306 0.6528

1.7840 6.5936

7.4610 -0.4851

8.7915 2.1516

1.1026 3.4334

bn = net.b{1}

15.4024

16.8403

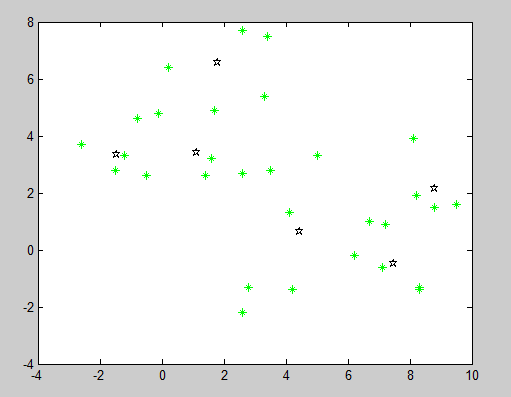
16.2416

16.9039

16.2298

16.3319

plot(w(:,1),w(:,2),'kp',P(1,:),P(2,:),'\*g');

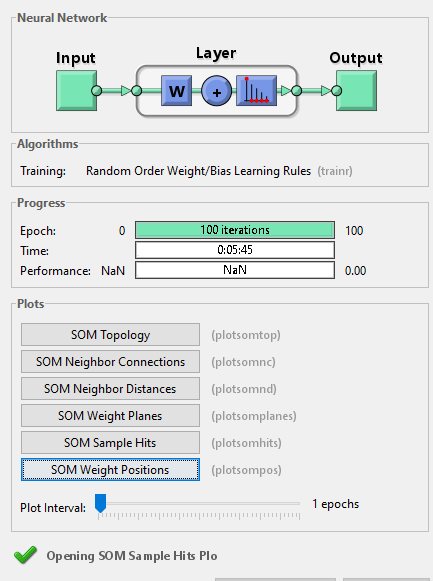


Теперь создадим самоорганизующуюся карту Кохенена с теми же входами

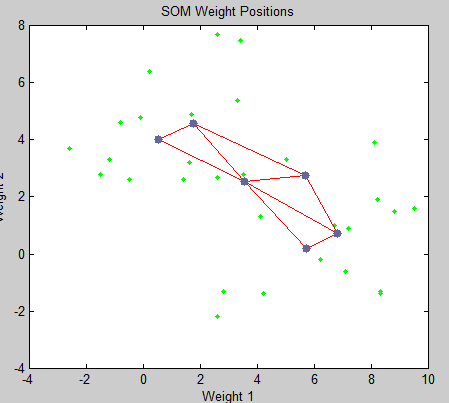
net = newsom ([a1 a2; b1 b2], [2 3]); % [2 3] – это размерность сети (6 кластеров как и раньше)

net.trainParam.epochs = 100;

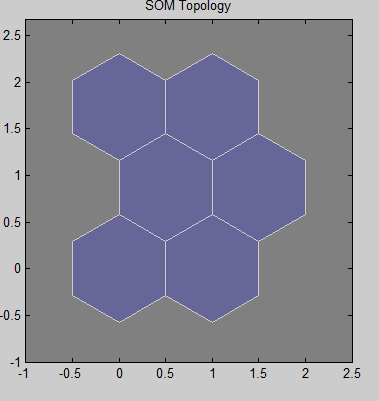
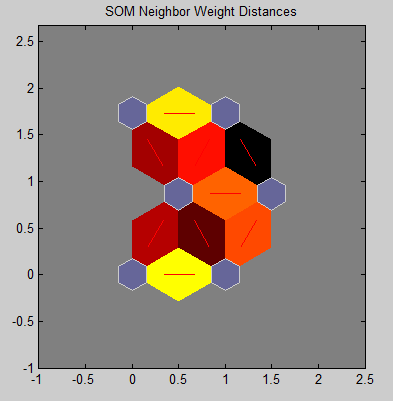
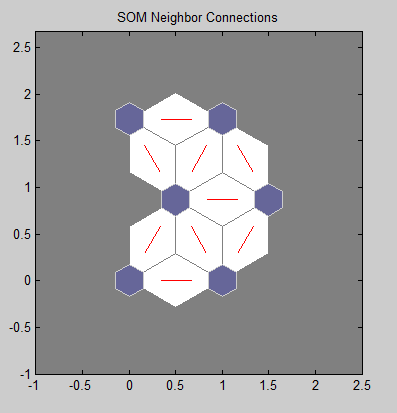
>> net = train(net, P)

В процессе обучения появляется это окно:

Нажмите SOM Weight Positions и наблюдайте изменение карты Кохенена в процессе обучения. Расположение точек данных и весовых векторов меняется



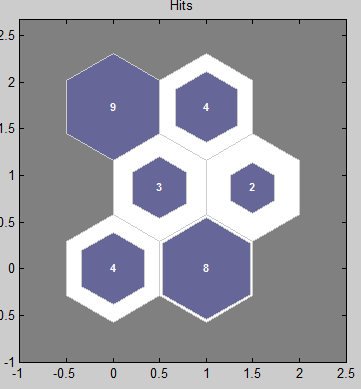
После обучения можно посмотреть другие визуализации сети

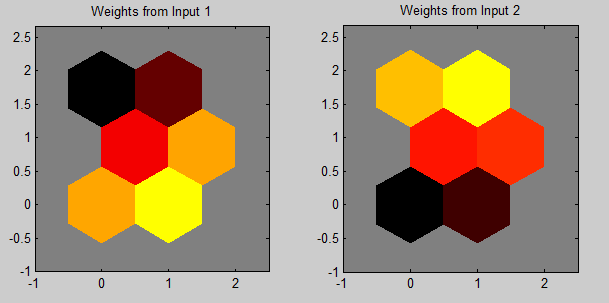


Расположение Связи между Сила взаимодействия

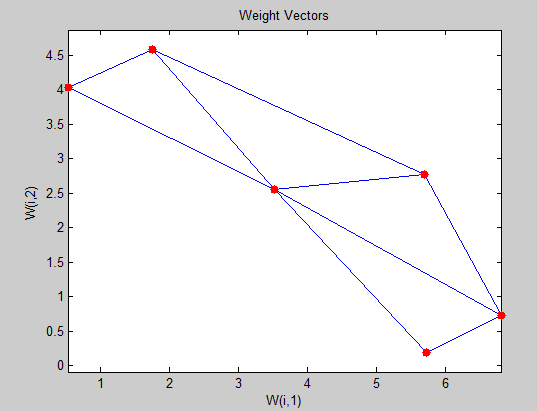
кластеров нейронами (веса нейронов) соседних

нейронов в карте Кохонена





Количество точек данных, Визуализация весов, соединяющих каждый вход с связанных с каждым нейроном каждым из нейронов



plotsom(net.iw{ 1,1}, net.layers{1}.distances)

% визуализация SOM

-

Введем значение точки, не присутствующей во входной выборке, например [2 2]

p = [2; 2];

a = 0; % начальная установка переменной a

a = sim(net, p) % получение отклика SOM на вход р

a =

(3,1) 1

Значит эта точка принадлежит 3 кластеру

**Кластеризация стран по 2-м входным значениям население и площадь**

Входные данные в виде таблицы:



Представить данные в векторном виде

P=[9519431 9598961 377835 3287590 357021 17075400 8511965 244820 547030 301230 1972550 98480 9976139 1904556 780580 2218000 2766890 1219912 7686850; 311028000 1343480000 127380000 1195950000 81757600 142052966 194481000 62041708 65447374 60340328 112100000 49773145 34470000 231369500 73723000 25721000 40134425 49320500 22649000]

И по столбцам для дальнейшей нормализации

x1=P(1,:)

x2=P(2,:)

u=minmax(P);

x11=(x1-u(1,1))/(u(1,2)-u(1,1)); % нормализация

x22=(x2-u(2,1))/(u(2,2)-u(2,1));

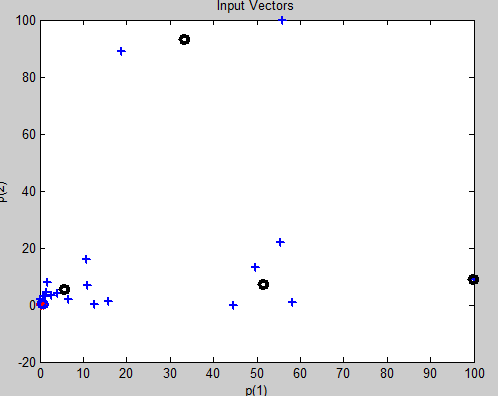
x11=100\*x11;

x22=100\*x22;

XYN=[x11;x22]; % нормализованная входная матрица

>> figure(1);

>> plot(x11,x22,'+','LineWidth',1.5) % построить график входных векторов ( черным намечены центры кластеризации)

>> title('Input vectors HxT');

>> xlabel('H');

>> ylabel('T');

>> grid on

hold on

>> nclusters=4; % Установить количество кластеров

>> net = newc([0 100; 0 100], nclusters, 0.1, 0.0005); % создать нейронную сеть слой Кохонена

% с нормализованными входными данными

% и параметрами обученности 0.1 и 0.0005

>> wo = net.IW{1} % просмотрим начальные веса, смещения

wo =

50 50

50 50

50 50

50 50

>> bo = net.b{1}

bo =

10.8731

10.8731

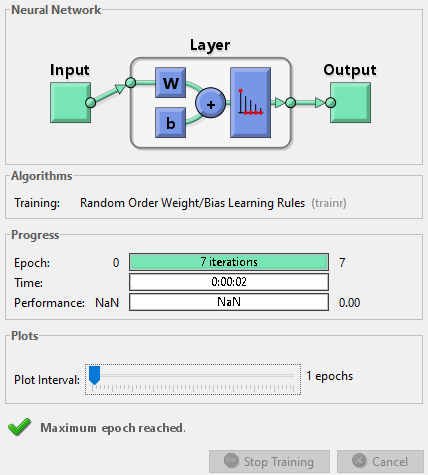
10.8731

10.8731

>> co = exp(1)./bo % и активность нейронов

co =

0.2500

 0.2500

0.2500

0.2500

>> net.trainParam.epochs=200; % задать количество эпох обучения

>> net.trainParam.show=20; % частоту показа результатов обучения

>> net = train(net,XYN) % провести обучение

w = net.IW{1} % посмотреть веса, смещения и активность

w = % нейронов после обучения

5.5977 5.2805

33.2745 93.1839

100.0000 9.0401

51.5334 7.0976

bn = net.b{1}

bn =

4.7505

21.0864

24.7162

14.3904

cn = exp(1)./bn

cn =

0.5722

0.1289

0.1100

0.1889

plot(w(:,1),w(:,2),'ko','LineWidth',3.5) % Построим центры кластеризации

a=sim(net,XYN); % моделируем работу сети на входной выборке

ac=vec2ind(a); % строим вектор строку для отображения результатов кластеризации

ас = (4 3 1 3 1 2 4 1 1 1 1 1 4 1 1 1 1 1 4)

Теперь можно по названию и номеру страны выяснить к какому кластеру она относится и вручную нанести на график

strana =' США '; class=ac(1) ;

strana =' КИТАЙ '; class=ac(2);

strana ='Япония'; class=ac(3) ;

strana ='Индия'; class=ac(4);

strana ='Германия'; class=ac(5);

strana ='Россия'; class=ac(6);

strana ='Бразилия'; class=ac(7);

strana ='Великобритания' ; class=ac(8);

strana ='Франция' ; class=ac(9);

strana ='Италия' ; class=ac(10);

strana ='Мексика'; class=ac(11) ;

strana ='Корея'; class=ac(12);

strana ='Канада'; class=ac(13);

strana ='Индонезия'; class=ac(14);

strana ='Турция'; class=ac(15);

strana ='Саудовская Аравия'; class=ac(16);

strana ='Аргентина'; class=ac(17);

strana ='ЮАР'; class=ac(18);

strana ='Австралия'; class=ac(19)

Получаем вот такой график для визуализации результатов кластеризации

