**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: Якимович А.И.

Группа: 80-408Б

Вариант: 20

Оценка:

Москва, 2018

Постановка задачи.

Целью работы является исследование свойств сетей Хопфилда, Хэмминга и Элмана, алгоритмов обучения, а также применение сетей в задачах распознавания статических и динамических образов.

Ход работы.

1. Использовать слой Кохонена для выполнения кластеризации множества точек. Проверить качество разбиения.

2. Использовать карту Кохонена для выполнения кластеризации множества точек.

3. Использовать карту Кохонена для нахождения одного из решений задачи коммивояжера.

4. Использовать сеть векторного квантования, обучаемую с учителем, (LVQ-сеть) для классификации точек в случае, когда классы не являются линейно разделимыми.

Теория.

Нейронные сети Кохонена — класс нейронных сетей, основным элементом которых является слой Кохонена. Слой Кохонена состоит из адаптивных линейных сумматоров («линейных формальных нейронов»). Как правило, выходные сигналы слоя Кохонена обрабатываются по правилу «Победитель получает всё»: наибольший сигнал превращается в единичный, остальные обращаются в ноль.

По способам настройки входных весов сумматоров и по решаемым задачам различают много разновидностей сетей Кохонена. Наиболее известные из них: сети векторного квантования сигналов, тесно связанные с простейшим базовым алгоритмом кластерного анализа (метод динамических ядер или K-средних); самоорганизующиеся карты Кохонена; сети векторного квантования, обучаемые с учителем .

Входные данные и результаты.

Вариант 20.

|  |  |
| --- | --- |
| C:\Users\now20\AppData\Local\Microsoft\Windows\INetCache\Content.Word\nnt6_1.pngC:\Users\now20\AppData\Local\Microsoft\Windows\INetCache\Content.Word\weights6.png |  |
| C:\Users\now20\AppData\Local\Microsoft\Windows\INetCache\Content.Word\6_2.png | C:\Users\now20\AppData\Local\Microsoft\Windows\INetCache\Content.Word\6_2f.png |
| C:\Users\now20\AppData\Local\Microsoft\Windows\INetCache\Content.Word\6_3f.png | C:\Users\now20\AppData\Local\Microsoft\Windows\INetCache\Content.Word\6_3.png |
|  | C:\Users\now20\AppData\Local\Microsoft\Windows\INetCache\Content.Word\6_4perf.png |

Выводы:

Самоорганизующиеся карты Кохонена служат, в первую очередь, для визуализации и первоначального («разведывательного») анализа данных. Каждая точка данных отображается соответствующим кодовым вектором из решётки. Так получают представление данных на плоскости («карту данных»). На этой карте возможно отображение многих слоёв: количество данных, попадающих в узлы (то есть «плотность данных»), различные функции данных и так далее. При отображении этих слоёв полезен аппарат географических информационных систем. Карта данных является подложкой для произвольного по своей природе набора данных. Карта данных служит заменой географической карте там, где географической карты просто не существует. Принципиальное отличие в следующем: на географической карте соседние объекты обладают близкими географическими координатами, на карте данных близкие объекты обладают близкими свойствами. С помощью карты данных можно визуализировать данные, одновременно нанося на подложку сопровождающую информацию (подписи, аннотации, атрибуты, информационные раскраски). Карта служит также информационной моделью данных. С её помощью можно заполнять пробелы в данных. Эта способность используется, например, для решения задач прогнозирования.

Исходный код.

clc

X=[0 1.5; 0 1.5];

clusters=8;

points=10; %3адание количества точек в кластере

std\_dev=0.1;

p=nngenc(X,clusters,points,std\_dev);

net = competlayer(8);

net.trainParam.epochs = 50;

net = train(net,p);

y = sim(net, p);

y = vec2ind(y);

rnd = makedist('Uniform',0,1.5);

P = random(rnd,2,5);

t = sim(net,P);

t = vec2ind(t);

IW = net.IW{1};

hold on

% draw original points

plot(p(1,y == 1), p(2,y == 1),'linestyle','none','marker','o');

plot(p(1,y == 2), p(2,y == 2),'linestyle','none','marker','+');

plot(p(1,y == 3), p(2,y == 3),'linestyle','none','marker','\*');

plot(p(1,y == 4), p(2,y == 4),'linestyle','none','marker','.');

plot(p(1,y == 5), p(2,y == 5),'linestyle','none','marker','p');

plot(p(1,y == 6), p(2,y == 6),'linestyle','none','marker','<');

plot(p(1,y == 7), p(2,y == 7),'linestyle','none','marker','d');

plot(p(1,y == 8), p(2,y == 8),'linestyle','none','marker','^');

% draw additional points

plot(P(1,t == 1), P(2,t == 1),'linestyle','none','marker','o','markersize',20);

plot(P(1,t == 2), P(2,t == 2),'linestyle','none','marker','+','markersize',20);

plot(P(1,t == 3), P(2,t == 3),'linestyle','none','marker','\*','markersize',20);

plot(P(1,t == 4), P(2,t == 4),'linestyle','none','marker','.','markersize',20);

plot(P(1,t == 5), P(2,t == 5),'linestyle','none','marker','x','markersize',20);

plot(P(1,t == 6), P(2,t == 6),'linestyle','none','marker','<','markersize',20);

plot(P(1,t == 7), P(2,t == 7),'linestyle','none','marker','d','markersize',20);

plot(P(1,t == 8), P(2,t == 8),'linestyle','none','marker','^','markersize',20);

% draw centers

plot(IW(:,1),IW(:,2),'marker','s','markersize',30, 'linestyle','none','color','black');

%plotsom(net.IW{1, 1}, net.layers{1}.distances)

hold off

clc

X=[0 1.5; 0 1.5];

clusters=8;

points=10; %3адание количества точек в кластере

std\_dev=0.1;

p=nngenc(X,clusters,points,std\_dev);

net = newsom(X, [2,4]);

net.trainParam.epochs = 50;

net = train(net,p);

y = sim(net, p);

y = vec2ind(y);

rnd = makedist('Uniform',0,1.5);

P = random(rnd,2,5);

t = sim(net,P);

t = vec2ind(t);

IW = net.IW{1};

hold on

% draw original points

plot(p(1,y == 1), p(2,y == 1),'linestyle','none','marker','o');

plot(p(1,y == 2), p(2,y == 2),'linestyle','none','marker','+');

plot(p(1,y == 3), p(2,y == 3),'linestyle','none','marker','\*');

plot(p(1,y == 4), p(2,y == 4),'linestyle','none','marker','.');

plot(p(1,y == 5), p(2,y == 5),'linestyle','none','marker','p');

plot(p(1,y == 6), p(2,y == 6),'linestyle','none','marker','<');

plot(p(1,y == 7), p(2,y == 7),'linestyle','none','marker','d');

plot(p(1,y == 8), p(2,y == 8),'linestyle','none','marker','^');

% draw additional points

plot(P(1,t == 1), P(2,t == 1),'linestyle','none','marker','o','markersize',20);

plot(P(1,t == 2), P(2,t == 2),'linestyle','none','marker','+','markersize',20);

plot(P(1,t == 3), P(2,t == 3),'linestyle','none','marker','\*','markersize',20);

plot(P(1,t == 4), P(2,t == 4),'linestyle','none','marker','.','markersize',20);

plot(P(1,t == 5), P(2,t == 5),'linestyle','none','marker','x','markersize',20);

plot(P(1,t == 6), P(2,t == 6),'linestyle','none','marker','<','markersize',20);

plot(P(1,t == 7), P(2,t == 7),'linestyle','none','marker','d','markersize',20);

plot(P(1,t == 8), P(2,t == 8),'linestyle','none','marker','^','markersize',20);

% draw centers

plot(IW(:,1),IW(:,2),'marker','s','markersize',30, 'linestyle','none','color','black');

%plotsom(net.IW{1, 1}, net.layers{1}.distances)

hold off

N = 20;

T = -1.5 + (1.5 + 1.5) \* rand(2, N);

hold on

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

hold off

net = newsom(T, N);

net.trainParam.epochs = 600;

net = train(net, T);

hold on

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

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

hold off

p = [

0.8 0.7 -0.8 -0.9 -0.7 -1.4 -1.1 -1 1.4 -1.4 -0.4 -0.9;

-0.4 1.1 -1.2 -0.5 1.2 0.2 1 0 -0.5 -0.9 -0.5 1.3];

t = [1 1 1 -1 1 -1 1 -1 1 -1 1 1];

tmp1 = t;

tmp1(tmp1 < 0) = 0;

%plotpv(t,t);

t(t > 0) = 2;

t(t <= 0) = 1;

vect = ind2vec(t);

net = lvqnet(12,0.1);

net.trainParam.epochs = 300;

net = train(net, p, vect);

%points = random\_points3();

tmp = -1.5:0.1:1.5;

[x,y] = meshgrid(tmp, tmp);

points = [x(:),y(:)]';

target = sim(net, points);

target = vec2ind(target) - 1;

plotpv(points,target);

point = findobj(gca,'type','line');

set(point,'Color','g');

hold on;

plotpv(p,tmp1)

hold off;