

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Институт информационных технологий
Кафедра «Информационные системы»**

**ЛАБОРАТОРНАЯ РАБОТА 6
«Основы нейронных сетей»**

Выполнил:
студент гр. ИС/б-21-1-о
Степанишина М.А.

Севастополь

2023

Лабораторная работа №6

«Исследование состязательных сетей и сетей векторного квантования»

Цель работы:

Углубление теоретических знаний в области обучения нейросетей без учителя, исследование свойств алгоритмов обучения состязательных сетей на основе правил Кохонена, приобретение практических навыков обучения самоорганизующихся карт Кохонена и сетей векторного квантования при решении задач классификации.

Ход работы:

Вариант задания:

7	2D	gridtop, mandist	25	0.0-5.0
---	----	------------------	----	---------

Задание:

-Сформировать в соответствии с таблицей 7.1 множество входных данных P для обучения SOM. Для генерации данных использовать генератор случайных чисел с равномерным распределением в диапазонах, указанных в таблице 7.1.

-Написать программу, создающую и обучающую SOM, заданной топологии (в соответствии с таблицей 7.1), на сгенерированных данных.

-Визуализировать топологию SOM в начале и конце обучения.

-Сгенерировать множество входных векторов P в виде кластеров с помощью функции `ann_gencluster(M, D, C, Std)`, где M – число векторов в кластере, D – размерность пространства, C – число кластеров, Std – стандартное отклонение относительно центра кластера. При этом распределить координаты векторов P в пределах заданных диапазонов входного пространства. Обучить SOM на этих данных, определить нейроны-

победители, распознающие кластеры, построить диаграмму распределения кластеров и узлов обученной SOM.

-Используя данные, сгенерированные в соответствии с п.3.5, создать и обучить LVQ сеть, которая: - содержит в скрытом слое такое же число нейронов как SOM, заданная по варианту в соответствии с таблицей 7.1; - группирует каждые 2 (при четном числе подклассов) или 3 (при нечетном числе подклассов) подкласса, формируемые скрытым слоем, в один класс.

-Оценить точность обучения LVQ классификатора.

Код программы построения диаграммы распределения данных:

```
C = 25; // число классов (по варианту)
N=[4 4]; // структура сети SOM (исходя из варианта 2D или 3D)
D=max(size(N)); // число измерений
M = 20; // выбираем число точек каждого класса
rng = [0 5] // диапазон значений данных (по варианту)
// генерируем равномерно распределенные данные (матрица размером Dx (CxM))
P=grand(D,C*M,'unf', rng(1),rng(2));
figure(0); clf
title('Исходные данные')
//диаграмма распределения данных
scatter(P(1,:), P(2,:)) //это для 2D
```

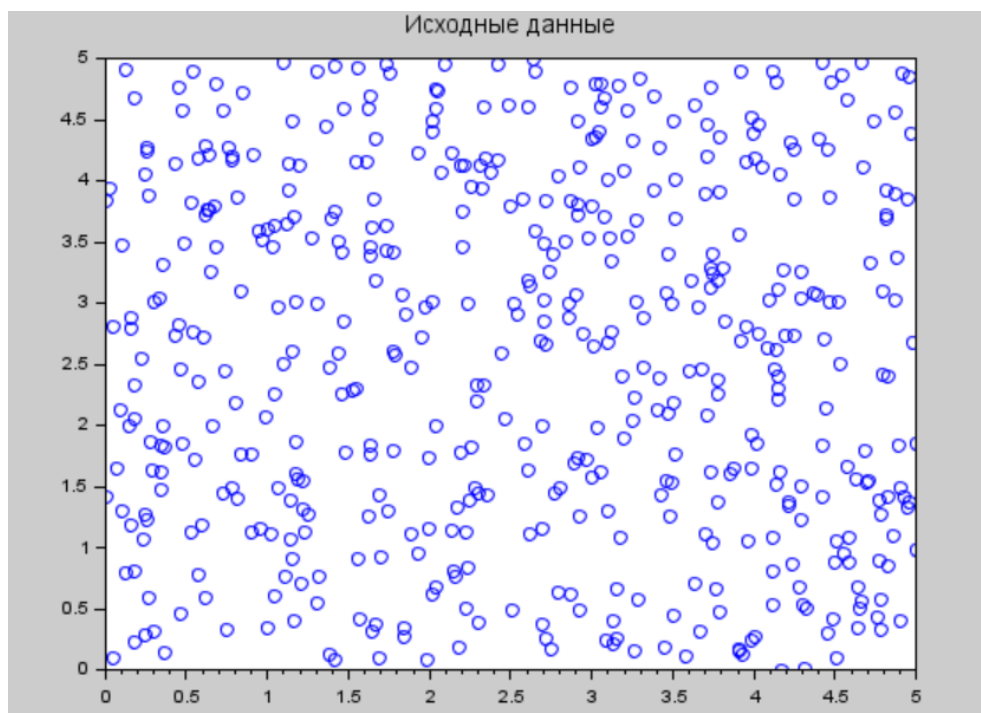


Рисунок 1 - Диаграмма распределения данных

Код программы SOM в начале и конце обучения:

```
exec('C:\Users\Мария\Documents\ОСН\Лаба 6\ann_SOM_visualize2d.sce');
exec('C:\Users\Мария\Documents\ОСН\Лаба 6\ann_som_gridtop.sce');
exec('C:\Users\Мария\Documents\ОСН\Лаба 6\ann_som_mandist.sce');
exec('C:\Users\Мария\Documents\ОСН\Лаба 6\ann_som_plot2d.sce');
exec('C:\Users\Мария\Documents\ОСН\Лаба 6\ann_negdist.sce');
exec('C:\Users\Мария\Documents\ОСН\Лаба 6\ann_dist.sce');
exec('C:\Users\Мария\Documents\ОСН\Лаба 6\ann_compet_activ.sce');
C = 25; // число классов (по варианту)
N=[4 4]; // структура сети SOM (исходя из варианта 2D или 3D)
D=max(size(N)); // число измерений
M = 20; // выбираем число точек каждого класса
rng = [0 5] // диапазон значений данных (по варианту)
// генерируем равномерно распределенные данные (матрица размером Dx (CxM))
P=grand(D,C*M,'unf', rng(1),rng(2));
figure(0); clf
title('Исходные данные')
//диаграмма распределения данных
scatter(P(1,:), P(2,:)) //это для 2D
//задание функции топологии
topfcfn = 'ann_som_gridtop' // или 'ann_som_hextop' или 'ann_som_randtop'
// задание функции расстояния
distfcfn = 'ann_som_mandist' // или 'ann_som_eudist' или 'ann_som_linkdist'
или
// или 'ann_som_boxdist'
//Обучение SOM
W = ann_SOM_visualize2d(P, N, 200, 100, 3, topfcfn, distfcfn);
//вызов функции топологии для вычисления позиций нейронов
neuron_pos=evstr(topfcfn + '(N)');
//вызов функции вычисления расстояний
neuron_dist=evstr(distfcfn + '(neuron_pos)');
figure(2);clf
drawlater
subplot(1,2,1)
// визуализация начальной топологии 2D SOM
// для визуализации 3D SOM замените вызов ann_som_plot2d на
// ann_som_plot3d(neuron_pos',neuron_dist)
ann_som_plot2d(neuron_pos',neuron_dist)
title('Начальная топология SOM')
drawnow
drawlater
subplot(1,2,2)
// визуализация топологии SOM после обучения
// для визуализации 3D SOM замените вызов ann_som_plot2d на
// ann_som_plot3d(neuron_pos',neuron_dist)
ann_som_plot2d(W, neuron_dist)
title('Топология SOM после обучения')
drawnow
// классификация входных данных обученной SOM
// y - выход состязательного слоя
// classes - индексы нейрона победителя
[y, classes] = ann_SOM_run(W,P)
```

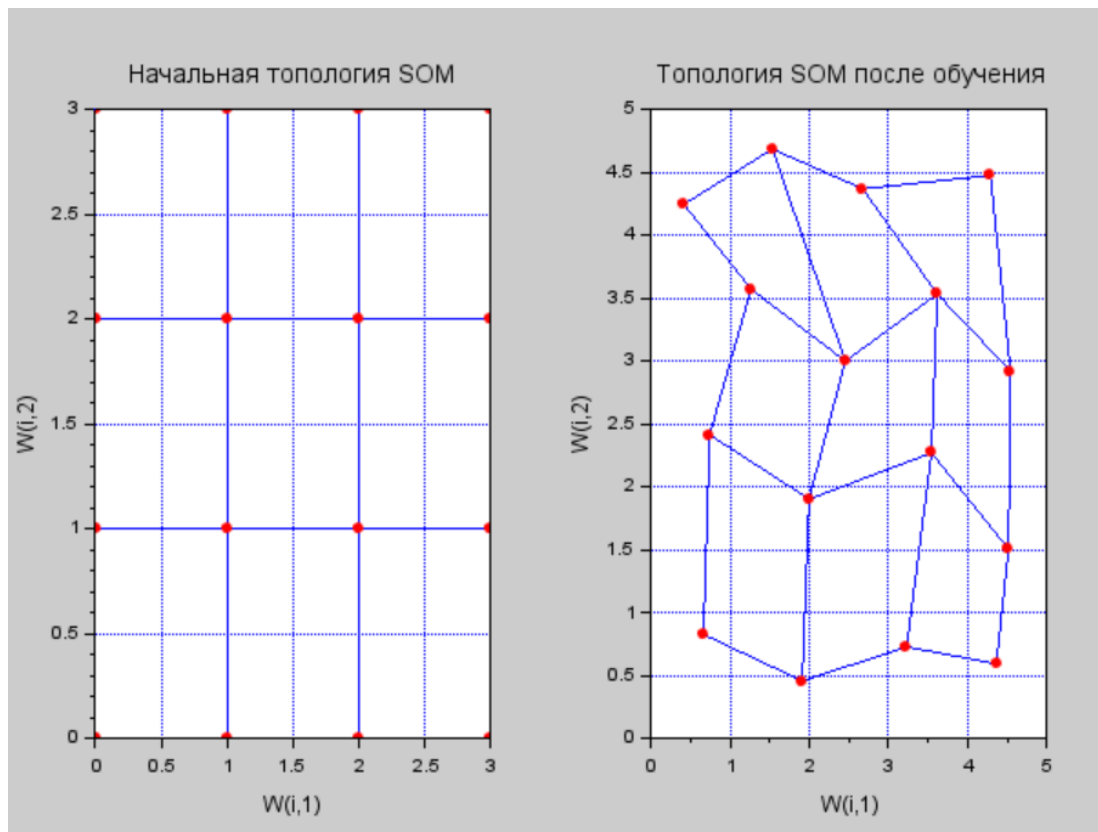


Рисунок 2 - Топология 2D SOM в начале (слева) и конце обучения (справа)

Код программы распределения кластеров данных и узлов обученной SOM:

```

exec('C:\Users\Мария\Documents\ОСН\Лаба 6\ann_SOM_visualize2d.sce');
exec('C:\Users\Мария\Documents\ОСН\Лаба 6\ann_som_gridtop.sce');
exec('C:\Users\Мария\Documents\ОСН\Лаба 6\ann_som_mandist.sce');
exec('C:\Users\Мария\Documents\ОСН\Лаба 6\ann_som_plot2d.sce');
exec('C:\Users\Мария\Documents\ОСН\Лаба 6\ann_negdist.sce');
exec('C:\Users\Мария\Documents\ОСН\Лаба 6\ann_dist.sce');
exec('C:\Users\Мария\Documents\ОСН\Лаба 6\ann_compet_activ.sce');
exec('C:\Users\Мария\Documents\ОСН\Лаба 6\ann_SOM_run.sce');
exec('C:\Users\Мария\Documents\ОСН\Лаба 6\ann_gencluster.sce');
//задание функции топологии
topfcn = 'ann_som_gridtop' // или 'ann_som_hextop' или 'ann_som_randtop'
// задание функции расстояния
distfcn = 'ann_som_mandist' // или 'ann_som_eudist' или 'ann_som_linkdist'
или
// или 'ann_som_boxdist'

//Обучение SOM
W = ann_SOM_visualize2d(P, N, 200, 100, 3, topfcn, distfcn);
//вызов функции топологии для вычисления позиций нейронов
neuron_pos=evstr(topfcn + '(N)');
//вызов функции вычисления расстояний
neuron_dist=evstr(distfcn + '(neuron_pos)');
figure(2);clf
drawlater
subplot(1,2,1)
// визуализация начальной топологии 2D SOM
// для визуализации 3D SOM замените вызов ann_som_plot2d на

```

```

// ann_som_plot3d(neuron_pos',neuron_dist)
ann_som_plot2d(neuron_pos',neuron_dist)
title('Начальная топология SOM')
drawnow
drawlater
subplot(1,2,2)
// визуализация топологии SOM после обучения
// для визуализации 3D SOM замените вызов ann_som_plot2d на
// ann_som_plot3d(neuron_pos',neuron_dist)
ann_som_plot2d(W, neuron_dist)
title('Топология SOM после обучения')
drawnow
// классификация входных данных обученной SOM
// y - выход состязательного слоя
// classes - индексы нейрона победителя
[y, classes] = ann_SOM_run(W,P)

// формирование кластеров данных
diap=(rng(2)-rng(1)); // диапазон данных
std=diap/(10*C); // стандартное отклонение данных ль центра кластера
P = diap*ann_gencluster(M,D,C,std); // генерация кластеров данных
// т.к. функция ann_gencluster формирует кластеры данных по очереди группами
// из M элементов, то выполняем случайную выборку значений из P
maxindP=max(size(P))
randind=int(maxindP*rand(1,maxindP))+1 // генерируем случайные индексы
P1=P(:,randind) // делаем выборку из P по случайным индексам

// Формируем обучающе мн-во для LVQ сети на основе P1
Plvq=P1
lenP=max(size(Plvq))
numC=int(C/2)
Tlvq=zeros(numC,lenP) // инициализируем T нулями
//формируем правильное мн-во T, объединив каждый 2-ой подкласс
for t=1:lenP
    Tlvq(ceil(classes(t)/2),t)=1
end
//Обучение LVQ
[Wlvq,blvq] = ann_LVQ1(Plvq,Tlvq,C,0.01,400);
// проверка на обучающем множестве
[y_lvq, classes_lvq] = ann_LVQ_run(Wlvq, Plvq)
// точность обучения
acc=sum(Tlvq==y_lvq)/(numC*lenP);
end;

```

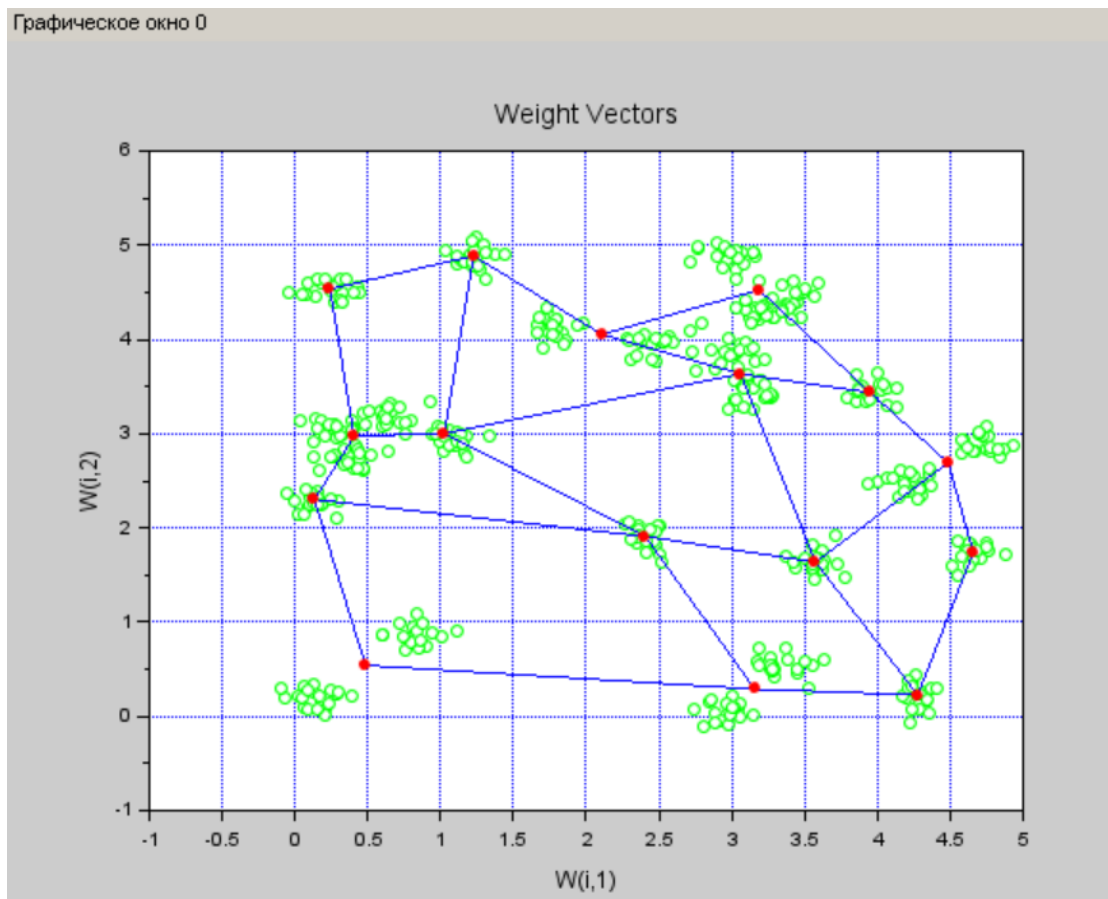


Рисунок 3 – Диаграмма распределения кластеров данных и узлов обученной SOM

Вывод:

В ходе лабораторной работы были углублены теоретические знания в области обучения нейросетей без учителя. Были исследованы свойства алгоритмов обучения состязательных сетей на основе правил Кохонена.

Были приобретены практические навыки обучения самоорганизующихся карт Кохонена и сетей векторного квантования при решении задач классификации.

Были построены диаграммы распределения данных и распределения кластеров данных и узлов обученной SOM. Так же была получена топология 2D SOM в начале и конце обучения.