Лабораторна робота №6

ШТУЧНА НЕЙРОННА МЕРЕЖА КОХОНЕНА

Мета роботи: набути практичних навичок використання системи комп'ютерної математики Matlab для розв'язання задач кластеризації за допомогою штучної нейронної мережі Кохонена.

Короткі теоретичні відомості

Штучна нейронна мережа Кохонена належить до мереж, що самоорганізуються, які під час надходження вхідних сигналів, не отримують інформацію про бажаний вихідний сигнал. Як наслідок, неможливо сформувати критерій налаштування, який би базувався на неузгодженості реальних і необхідних вихідних сигналів штучної нейронної мережі, тому вагові параметри мережі корегують, виходячи з інших міркувань. Усі подані вхідні сигнали із заданої навчальної множини самоорганізовувана мережа у процесі навчання розділяє на класи, будуючи так звані топологічні карти.

До нейронів, розташованих в одному шарі, що є двовимірною площиною, підходять нервові волокна, по яких надходить N-вимірний вхідний сигнал. Кожен нейрон характеризується своїм розміщеним у шарі й ваговим коефіцієнтом. Розміщення нейронів, у свою чергу, характеризується деякою метрикою й визначається топологією шару, при якій сусідні нейрони під час навчання впливають один на одного сильніше, ніж розташовані далі.

Наявність зв'язків між нейронами призводить до того, що при збуджені одного з них можна обчислити збудження інших нейронів у шарі, причому це збудження зі збільшенням відстані від збудженого нейрона зменшується. Тому центр реакції шару, що виникає у відповідь на отримане роздратування, відповідає місцезнаходженню збудженого нейрона. Зміна вхідного сигналу, що навчає, призводить до максимального збудження іншого нейрона ц відповідно – до іншої реакції.

Виконання роботи

6.1. Використовуючи систему комп’ютерної математики Matlab визначити до якого із двох кластерів належать чотири двохелементні вектори

Варіант 7 [6.907 7.826 7.099 8.05; 6.71 7.98 7.25 8.16]

```
Command Window
>> p = [6.907 7.826 7.099 8.05; 6.71 7.98 7.25 8.16]

p =

    6.9070    7.8260    7.0990    8.0500
    6.7100    7.9800    7.2500    8.1600

>> net = newc([0 1; 0 1], 2)

net =

    Neural Network

              name: 'Custom Neural Network'
          userdata: (your custom info)

    dimensions:

         numInputs: 1
         numLayers: 1
        numOutputs: 1
    numInputDelays: 0
    numLayerDelays: 0
 numFeedbackDelays: 0
 numWeightElements: 6
        sampleTime: 1

    connections:

       biasConnect: true
      inputConnect: true
      layerConnect: false
     outputConnect: true
```

Рис. 6.1. Введення початкових даних та створення шару Когонена

```
>> wts = net.IW{1,1}

wts =

    0.5000    0.5000
    0.5000    0.5000

>> net.layers{1}

ans =

    Neural Network Layer

              name: 'Layer'
        dimensions: 2
       distanceFcn: (none)
     distanceParam: (none)
         distances: []
            initFcn: 'initwb'
       netInputFcn: 'netsum'
     netInputParam: (none)
         positions: []
             range: [2x2 double]
              size: 2
        topologyFcn: (none)
        transferFcn: 'compet'
     transferParam: (none)
          userdata: (your custom info)
```

Рис. 6.2. Визначення параметру шару Когонена

```
>> net.biases{1}

ans =

    Neural Network Bias

            initFcn: 'initcon'
              learn: true
           learnFcn: 'learncon'
         learnParam: .lr
               size: 2
           userdata: (your custom info)

>> net.b{1}

ans =

    5.4366
    5.4366
```

Рис. 6.3. Визначення параметрів зміщення шару Когонена

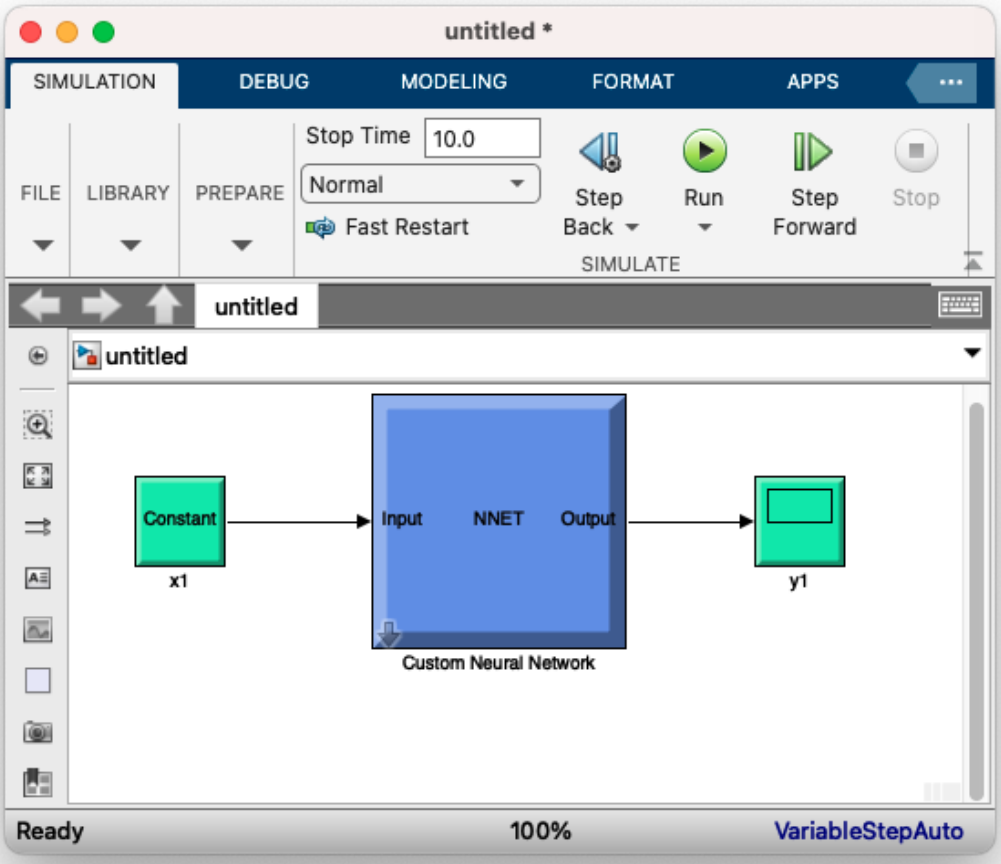


Рис. 6.4. Графічна інтепретація архітектури шару Когонена

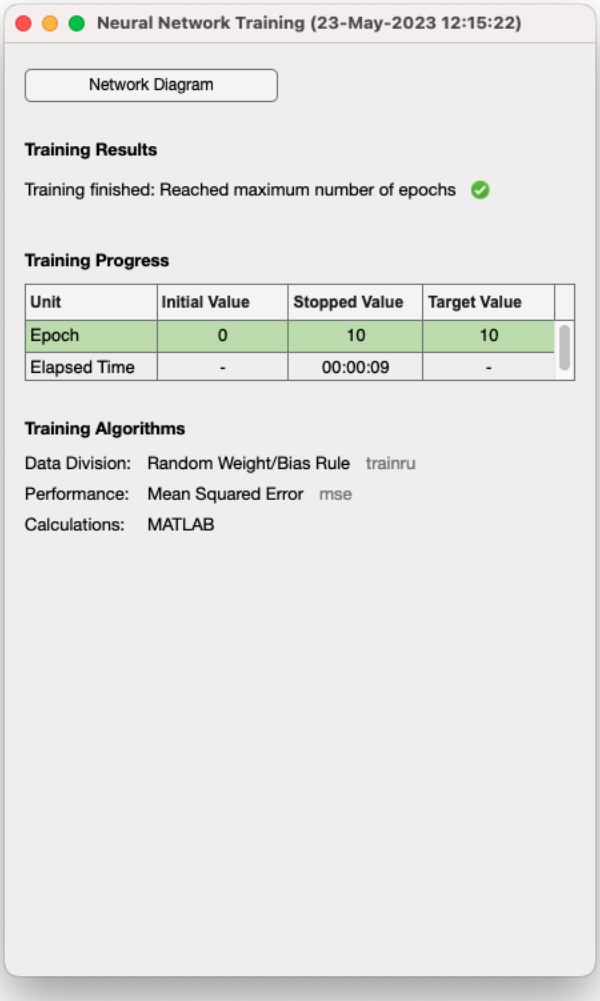


Рис. 6.5. Навчаємо шар Когонена кластеризувати ознаки об'єктів

```
>> a = sim(net, p);
ac = vec2ind(a)

ac =

     1     1     1     1

>> wts1 = net.IW{1,1}

wts1 =

    2.8073    2.8254
    0.5000    0.5000

>> b1 = net.b{1}

b1 =

    5.2313
    5.6585
```

Рис. 6.6. Визначення кластеризації даних векторів, значення зміщення та вагових коефіцієнтів

6.2. В системі комп’ютерної математики Matlab, використовуючи конкуренту мережу із (6+0.25k) нейронів, виконати кластеризацію об’єктів, якщо кожен із них має (4+0.25k) ознаки, які набувають значень з проміжку [- 3·1.12·k 13·1.12·k; -2·1.12·k 7·1.12·k]. Розраховуючи кількість нейронів та кількість ознак, отримані величини округлити до найближчого цілого числа. Для генерації навчальної послідовності прийняти, що середнє квадратичне відхилення дорівнює (0.48+0.013·k), де k – номер варіанта.

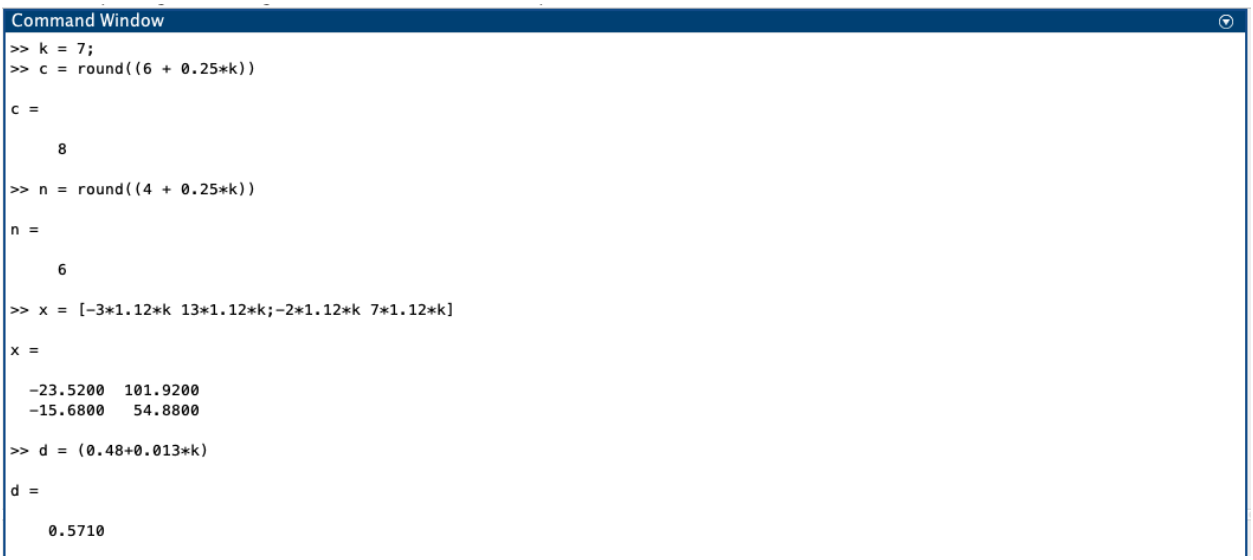


Рис. 6.7. Введення початкових даних

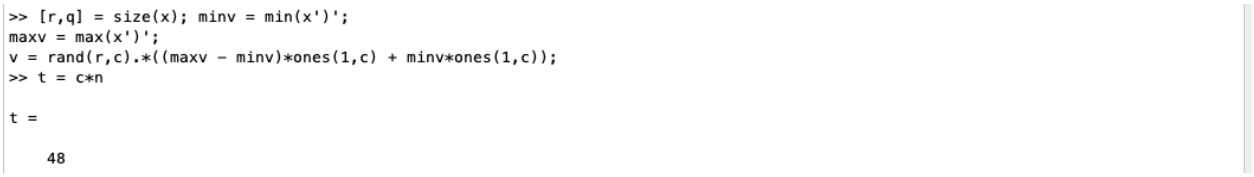


Рис. 6.8. Розрахунок загального числа яке необхідно кластеризувати

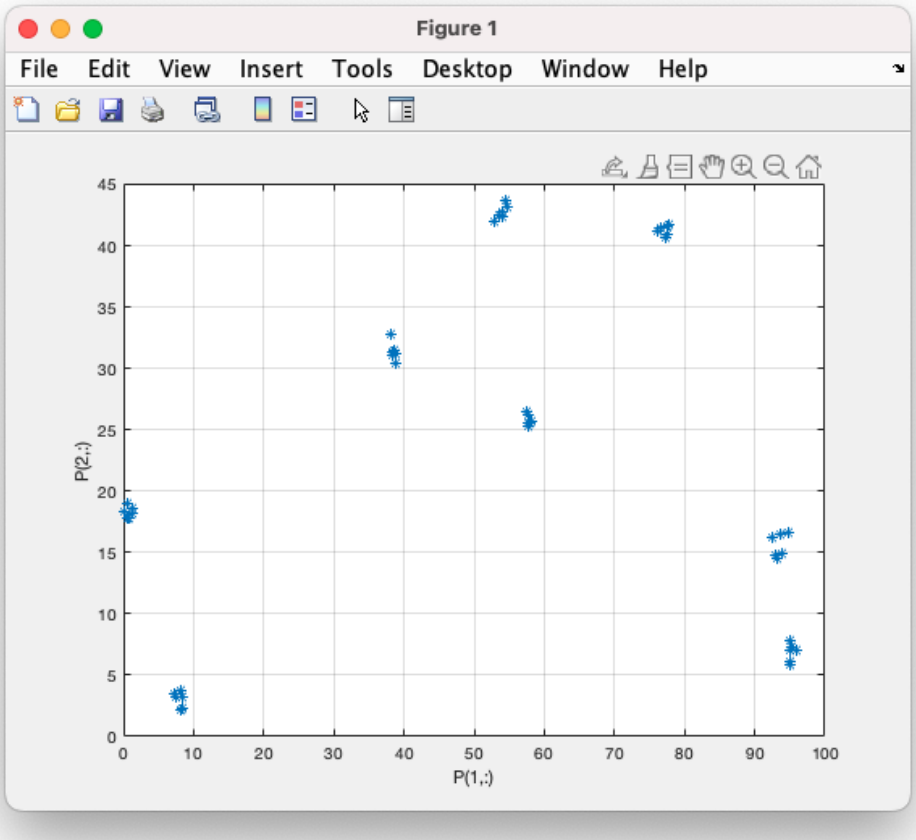


Рис. 6.9. План розміщення точок у просторі

```
>> net = newc([-2 12;-1 6], 8 ,0.1);
>> w0 = net.IW{1}

w0 =

    5.0000    2.5000
    5.0000    2.5000
    5.0000    2.5000
    5.0000    2.5000
    5.0000    2.5000
    5.0000    2.5000
    5.0000    2.5000
    5.0000    2.5000

>> b0 = net.b{1}; c0 = exp(1)./b0;
>> b0 = net.b{1}; c0 = exp(1)./b0

c0 =

    0.1250
    0.1250
    0.1250
    0.1250
    0.1250
    0.1250
    0.1250
    0.1250
```

Рис. 6.10. Визначення параметрів зміщення та вагових коефіцієнтів

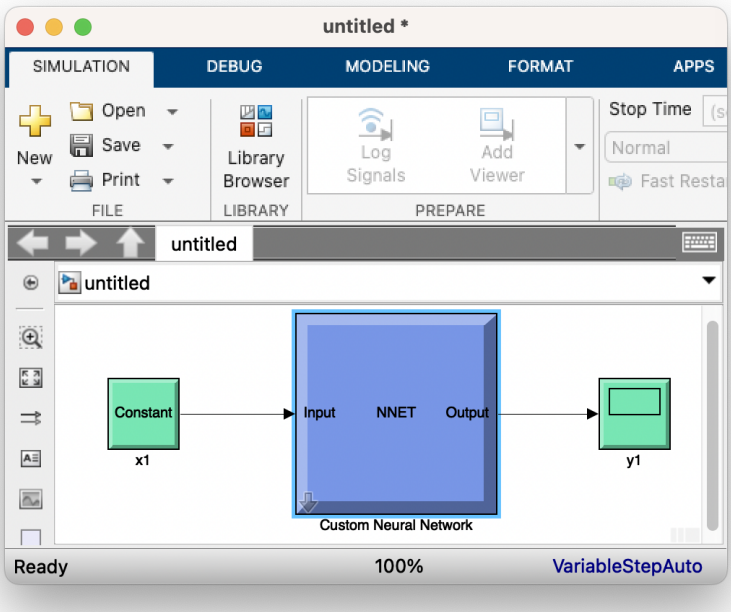


Рис. 6.11. Графічна інтепритація архітектури шару когонена

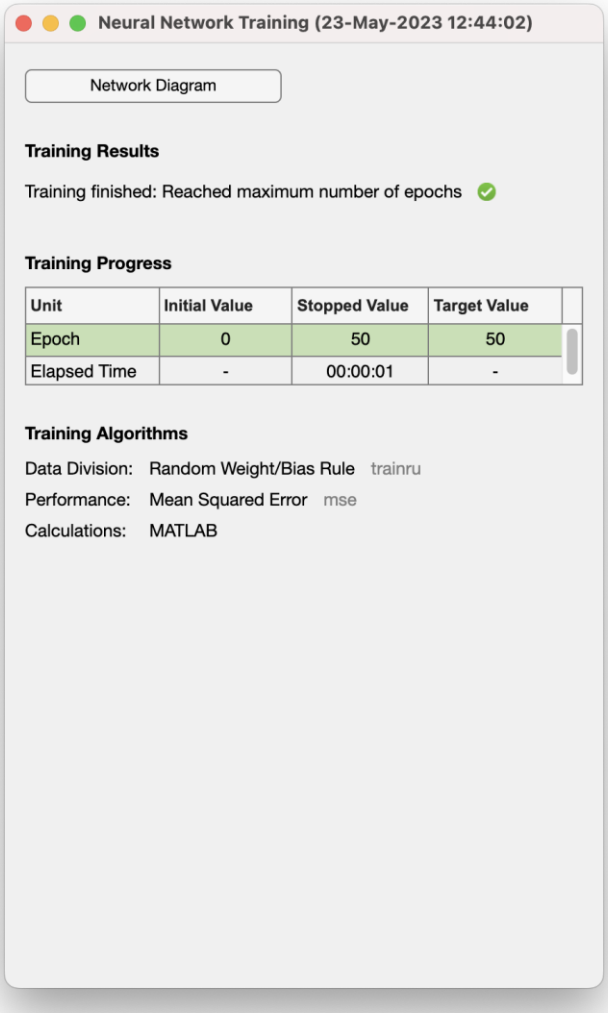


Рис. 6.12. Начення шару Когонена кластеризувати ознаки об'єктів

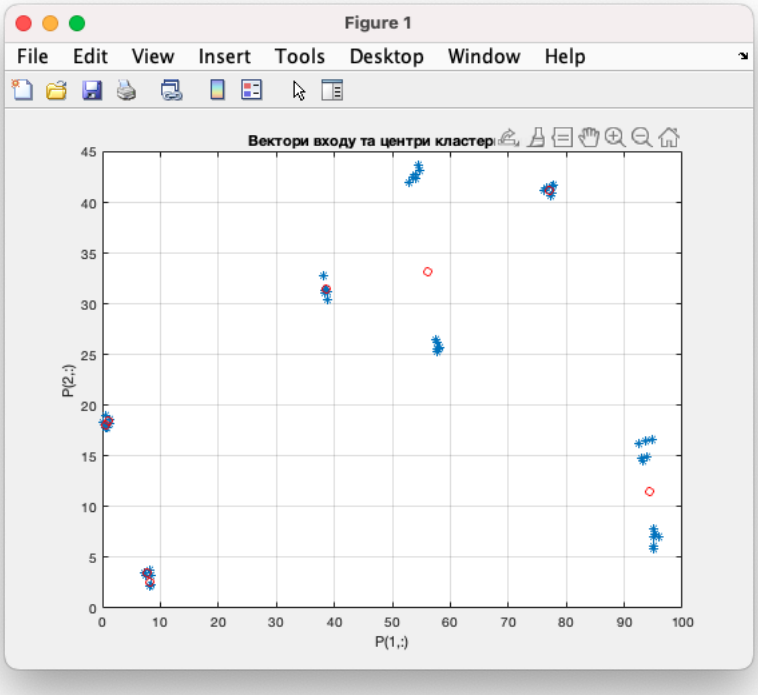


Рис. 6.13. Графік центрів кластеразиції на основі векторів входу

6.3. В системі комп'ютерної математики Matlab виконати кластеризацію двохелементних векторів, які набувають значень з проміжку [2.4·k 4.1·k; 1·k 7·k]. Розмір гексагональної сітки апріорно заданий і дорівнює 3×4. Навчальна вибірка повинна містити не менше 140 елементів, а кількість циклів навчання становити 173+k, де k – номер варіанта. Для генерації навчальної множини використати команду P = randint(n, m, [a,b]), де n – кількість рядків, m – кількість стовпців, [a,b] – інтервал, якому мають належати елементи навчальної множини.

```
Command Window
>> k = 7;
>> x = [2.4*k 4.1*k; 1*k 7*k]

x =

   16.8000   28.7000
    7.0000   49.0000

>> x = [2.4*k 4.1*k; 1*k 7*k]

x =

   16.8000   28.7000
    7.0000   49.0000

>> P = randi(49,[3, 4])

P =

    43    26    22    34
    30     3    27    19
    10    43    28     4
```

Рис. 6.14. Введення початкових даних

```
>> net = newsom(x, [3 4]);
net.layers{1}

ans =

    Neural Network Layer

            name: 'Layer'
      dimensions: [3 4]
     distanceFcn: 'linkdist'
   distanceParam: (none)
       distances: [12x12 double]
         initFcn: 'initwb'
     netInputFcn: 'netsum'
   netInputParam: (none)
       positions: [2x12 double]
           range: [12x2 double]
            size: 12
     topologyFcn: 'hextop'
     transferFcn: 'compet'
   transferParam: (none)
        userdata: (your custom info)
```

Рис. 6.15. Створення самоорганізованої карти Когонена

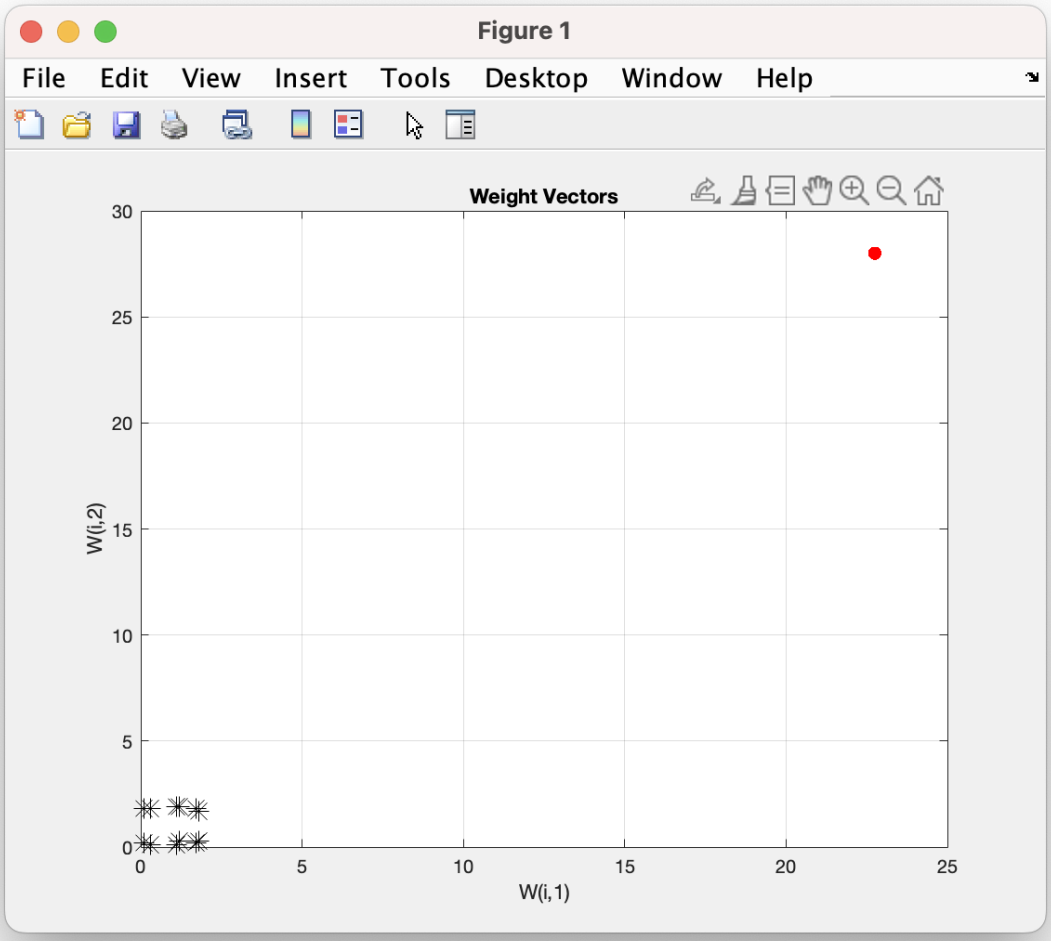


Рис. 6.16. Топологічна карта

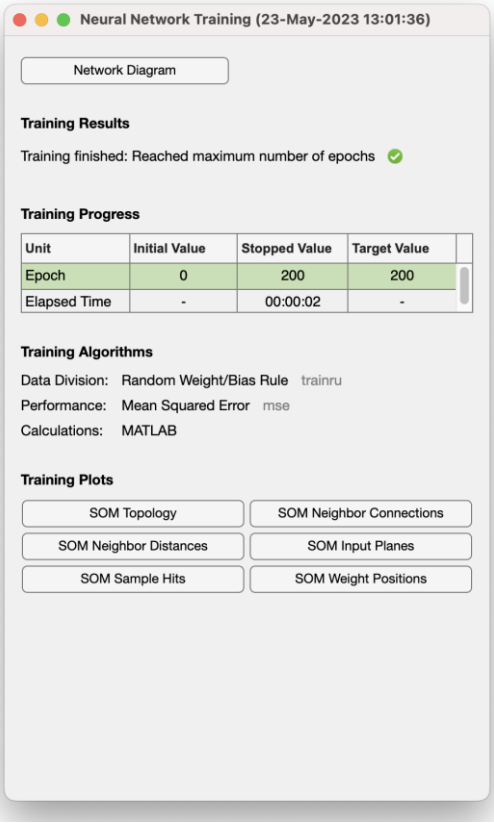


Рис. 6.17. Навчити карту Когонена кластеризувати ознаки об’єктів.

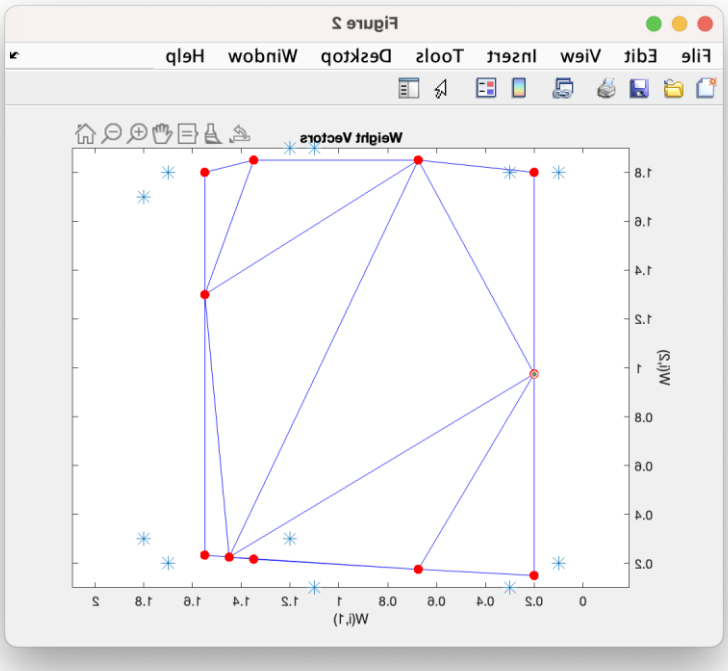
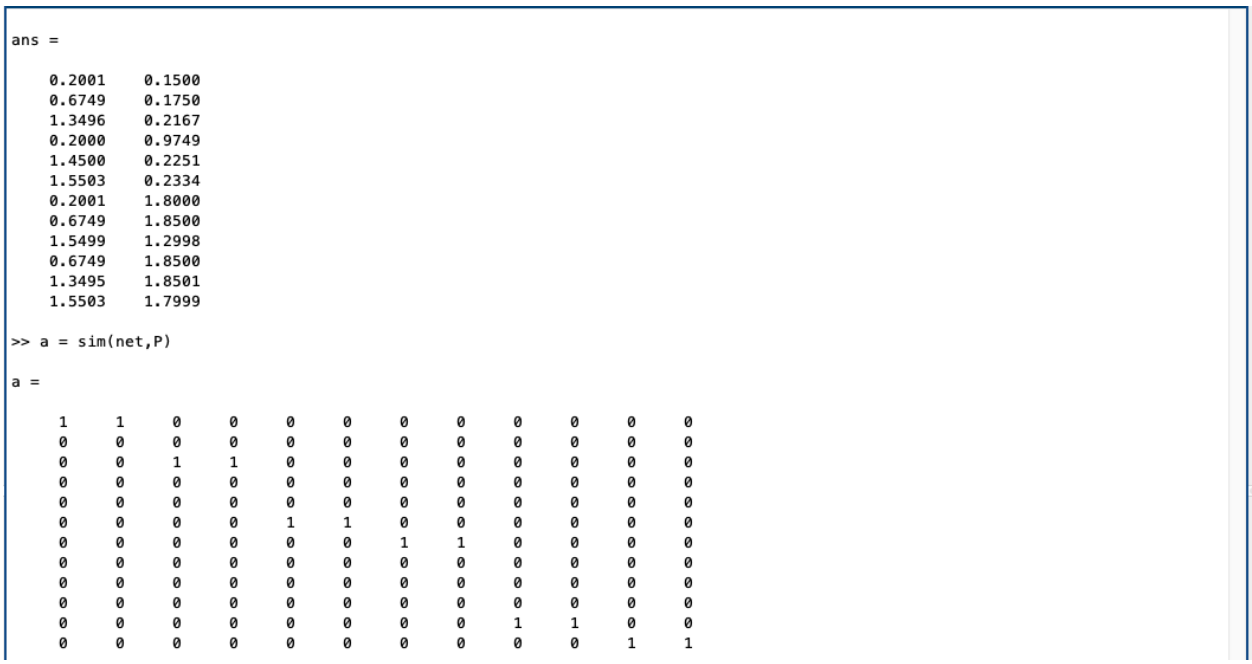


Рис. 6.18. Побудувати топологічну карти навченої системи



```
ans =

    0.2001    0.1500
    0.6749    0.1750
    1.3496    0.2167
    0.2000    0.9749
    1.4500    0.2251
    1.5503    0.2334
    0.2001    1.8000
    0.6749    1.8500
    1.5499    1.2998
    0.6749    1.8500
    1.3495    1.8501
    1.5503    1.7999

>> a = sim(net,P)

a =

     1     1     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     1     1     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     1     1     0     0     0     0     0     0
     0     0     0     0     0     0     1     1     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     1     1     0     0
     0     0     0     0     0     0     0     0     0     0     1     1
```

Рис. 6.19. Виконати кластеризацію ознак навчального вектора.

Висновок: набуто практичні навички з використання системи комп’ютерної математики Matlab для розв’язання задач кластеризації за допомогою штучної нейронної мережі Кохонена.