

# ЛАБОРАТОРНА РОБОТА №7

## Радіально-базисна нейронна мережа

Мета роботи: Набути практичних навичок використання системи комп'ютерної математики Matlab для розв'язання задач апроксимації функції багатьох змінних за допомогою радіально-базисних штучних нейронних мереж.

### Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Разом з викладачем вибрати варіант завдання.
3. Виконати завдання до лабораторної роботи згідно свого варіанту.
4. Скласти та оформити звіт.

### Теоретичні відомості

За допомогою радіально-базисних функцій можна як завгодно точно апроксимувати задану функцію.

Ідея навчання радіально-базисної штучної нейронної мережі полягає у застосуванні методу потенціальних функцій, що дозволяє подати деяку функцію  $y(\mathbf{x})$  у вигляді суперпозиції потенціальних або базисних функцій  $f_i(\mathbf{x})$

$$y(\mathbf{x}) = \sum_{i=1}^N a_i f_i(\mathbf{x}) = \mathbf{a}^T \mathbf{f}(\mathbf{x}),$$

де  $\mathbf{a}_i(t) = (a_{i1}, a_{i2}, \dots, a_{iN})^T$  – вектор параметрів, які підлягають визначенню;  $\mathbf{f}(\mathbf{x}) = (f_1(x), f_2(x), \dots, f_N(x))^T$  – вектор базисних функцій.

Базисними функціями в радіально-базисних штучних нейронних мережах можуть бути деякі функції відстані між векторами

$$f_i(\mathbf{x}) = f(\|\mathbf{x} - \mathbf{c}_i\|).$$

Вектори  $\mathbf{c}_i$  називаються центрами базисних функцій. Функції  $f_i(\mathbf{x})$  вибираються невід'ємними й зростаючими при зменшенні  $\|\mathbf{x} - \mathbf{c}_i\|$ . Для обчислення відстані між векторами  $\mathbf{x}$  і  $\mathbf{c}_i$  використовують звичайну, евклідову

$\|\mathbf{x} - \mathbf{c}_i\| = \left( \sum_{j=1}^N (x_j - c_{ij})^2 \right)^{\frac{1}{2}}$  або манхетенську  $\|\mathbf{x} - \mathbf{c}_i\| = \sum_{j=1}^N |x_j - c_{ij}|$  метрики, де

$$|x_j - c_{ij}| = (x_j - c_{ij}) \operatorname{sgn}(x_j - c_{ij}), \quad \operatorname{sgn}(x_j - c_{ij}) = \begin{cases} 1, & \text{якщо } (x_j - c_{ij}) > 0; \\ 0, & \text{якщо } (x_j - c_{ij}) = 0; \\ -1, & \text{якщо } (x_j - c_{ij}) < 0. \end{cases}$$

Радіально-базисні мережі мають багато спільного зі стохастичними мережами. Як і стохастичні мережі, радіально-базисні нейронні мережі мають високу швидкість навчання, а також в процесі навчання не виникає проблем з «потраплянням» у локальні мінімуми. Проте при виконанні самої класифікації

виконують досить складні обчислення, внаслідок чого зростає час отримання результату.

Особливістю радіально-базисної нейронної мережі є наявність радіально-симетричного шаблонного шару. Структура радіально-базисної нейронної мережі відповідає мережі прямого поширення першого порядку (рис. 7.1).

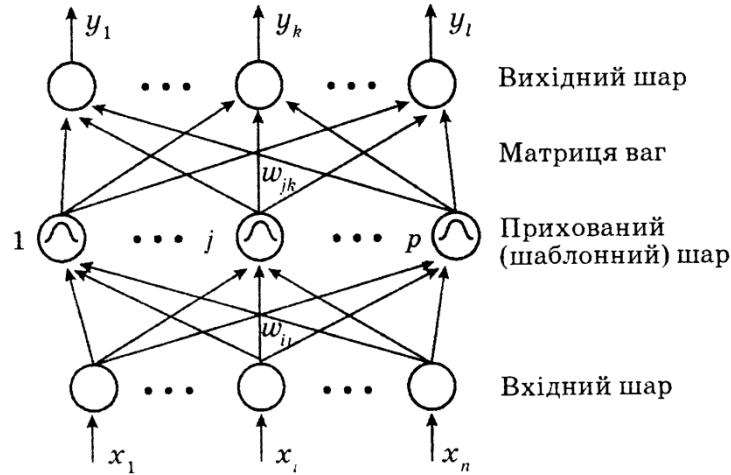


Рисунок 7.1 – Структура радіально-базисної мережі

Інформація про образи передається із вхідного шару на прихований, що є шаблонним і містить  $p$  нейронів. Кожен шар шаблонного шару, отримуючи повну інформацію про вхідні сигнали  $\mathbf{x}$ , обчислює функцію:

$$f_i(\mathbf{x}) = f\left(\left(\mathbf{x} - \mathbf{c}_i\right)^T R^{-1} \left(\mathbf{x} - \mathbf{c}_i\right)\right), i = \overline{1, p},$$

де  $\mathbf{x}$  – вектор вхідних сигналів  $(N - 1)$ ;  $\mathbf{c}_i$  – вектор центрів,  $R$  – вагова матриця.

Відстань між вхідним вектором і центром, подається у вигляді вектора у вхідному просторі. Вектор центрів визначається за навчальною вибіркою й зберігається в просторі ваг від вхідного шару до шару шаблонів.

В загальному випадку радіально-базисну штучну нейронну мережу характеризують три типи параметрів:

- лінійні вагові параметри вихідного шару  $w_{ij}$ ;
- центри  $\mathbf{c}_i$  – нелінійні параметри прихованого шару;
- відхилення (радіуси базисних функцій)  $\sigma_{ij}$  – нелінійні параметри прихованого шару.

Навчання мережі полягає у визначенні цих параметрів, може зводитися до одного з варіантів:

1. Задають центри й відхилення, а обчислюються тільки ваги вихідного шару.
2. Визначаються шляхом самонавчання центри у відхилення, а для корекції ваг вихідного шару використовується навчання із учителем.
3. Визначаються всі параметри мережі за допомогою навчання із учителем.

Перші два варіанти застосовуються в мережах, що використовують базисні функції з жорстко заданим радіусом (відхиленням). Третій варіант,

будучи найбільш складним і трудомістким у реалізації, припускає використання будь-яких базисних функцій.

Отже навчання мережі полягає в такому:

- визначаються центри  $\mathbf{c}_i$ ;
- вибираються параметри  $\sigma_i$ ;
- обчислюються елементи матриці ваг  $W$ .

Розглянемо принципи створення радіально-базисних нейронних мереж в системі комп'ютерної математики Matlab.

Для побудови радіальної нейронної мережі з нульовою помилкою існує функція **newrbe**, що у загальному випадку має такий синтаксис:

$$\text{net} = \text{newrbe}(P, T, \text{SPREAD}),$$

де

$P$  – матриця вхідних факторів, що подається вектором-стовпцем, кожний елемент якого є вектором-рядком і відповідає навчальному образу;

$T$  – матриця реальних вихідних величин (цільових значень), що подається вектором-стовпцем, кожний елемент якого є вектором-рядком і однозначно відображає реальні вихідні величини для заданого навчального образу;

$\text{SPREAD}$  – параметр впливу.

Функція **newrbe** повертає радіальну штучну нейронну мережу з нульовою помилкою з такими вагами і відхиленнями, що її виходи точно дорівнюють цілям  $T$ , а кількість створених нейронів радіального базисного шару відповідає кількості вхідних векторів масиву  $P$ . Вагові коефіцієнти першого шару дорівнюють  $P^T$ . При цьому, зміщення має становити  $0,8326/\text{SPREAD}$ , оскільки всі входи в діапазоні  $\pm \text{SPREAD}$  вважаються значимими, тобто чим більший діапазон вхідних значень повинен бути врахований, тим більше значення параметру впливу  $\text{SPREAD}$  повинне бути встановлено.

**Приклад 7.1.** В системі комп'ютерної математики Matlab створити та навчити радіально-базисну штучну нейронну мережу виконувати апроксимацію функціональної залежності. Вектор вхідних значень  $P$  визначений на інтервалі  $P = [-1, 1]$ . Вектор цілей  $T = [-0,9602 \ -0,577 \ 0,729 \ 0,3771 \ 0,6405 \ 0,66 \ 0,4609 \ 0,1336 \ 0,2113 \ 0,4344 \ 0,5 \ 0,393 \ 0,1647 \ 0,0988 \ 0,3072 \ 0,3960 \ 0,3449 \ 0,1816 \ 0,312 \ 0,2189 \ 0,3201]$ .

*Розв'язання:*

*Крок 1.* Ввести початкові дані.

```
clear
```

```
P = -1:1:1;
```

```
T = [-.9602 -.5770 -.0729 .3771 .6405 .6600 .4609 .1336 -.2013 -.4344 ...  
-.5000 -.3930 -.1647 .0988 .3072 .3960 .3449 .1816 -.0312 -.2189 -.3201];
```

*Крок 2.* Виконати графічну інтерпретація початкових даних (рис. 7.2)

```
figure(1), plot(P,T,'*r','MarkerSize',4,'LineWidth',2),  
grid on, hold on
```

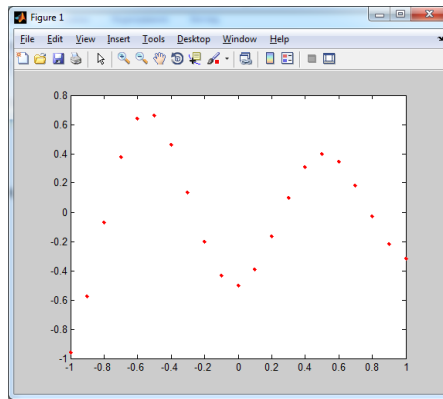


Рисунок 7.2 – Початкові дані

Крок 3. Створити радіально-базисну штучну нейронну мережу.

```
net = newrbf(P,T);
```

**Rank deficient, rank = 13, tol = 2.2386e-014**

Крок 4. Налаштувати кількість нейронів у прихованому шарі.

```
net.layers{1}.size
```

**ans = 21**

Крок 5. Виконати моделювання мережі.

```
V = sim(net,P); % Контрольні точки із навчальної множини
```

```
plot(P,V,'ob','MarkerSize',5, 'LineWidth',2)
```

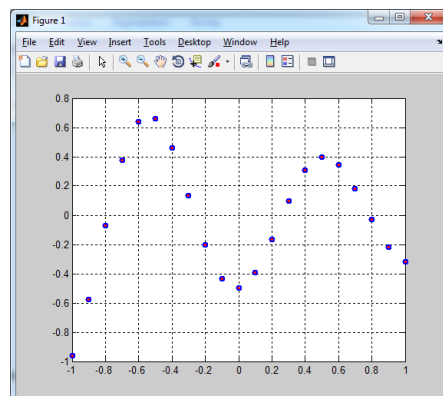


Рисунок 7.3 – Контрольні точки та навчальна множина

```
p = [-0.75 -0.25 0.25 0.75];
```

```
v = sim(net,p); % Інші контрольні точки навчальної множини
```

```
plot(p,v,'+k','MarkerSize',10, 'LineWidth',2)
```

```
xlabel('P, p'), ylabel('T, v')
```

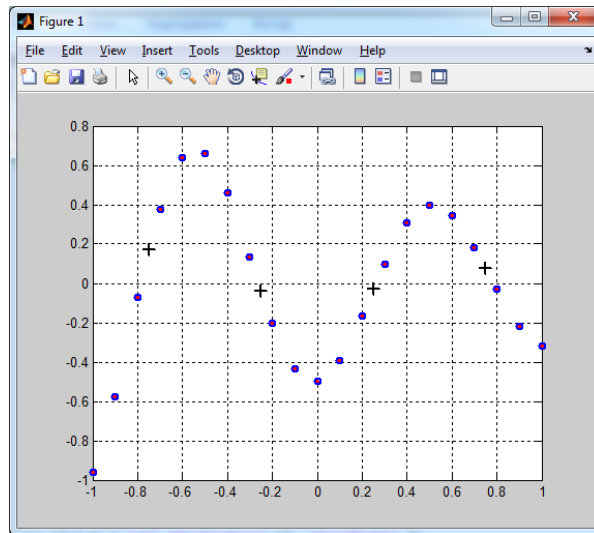


Рисунок 7.4 – Контрольні точки (вектор входу із навчальної множини) та навчальна множина

Для побудови радіальної штучної нейронної мережі з використанням ітеративної процедури, яка додає по одному нейрону на кожному кроці, існує функція **newrb**, що у загальному випадку має такий синтаксис:

$$\text{net} = \text{newrb}(P, T, \text{GOAL}, \text{SPREAD}),$$

де

$P$  – матриця вхідних факторів, що подається вектором-стовпцем, кожний елемент якого є вектором-рядком і відповідає навчальному образу;

$T$  – матриця реальних вихідних величин (цільових значень), що подається вектором-стовпцем, кожний елемент якого є вектором-рядком і однозначно відображає реальні вихідні величини для заданого навчального образу;

$GOAL$  – допустима середньоквадратична помилка;

$SPREAD$  – параметр впливу.

За цим методом нейрони до прихованого шару додаються до того моменту, коли сума квадратів помилок не стане менше заданого значення, або не буде використана максимальна кількість нейронів.

**Приклад 7.2.** В системі комп'ютерної математики Matlab створити та навчити радіально-базисну штучну нейронну мережу виконувати апроксимацію функціональної залежності із використанням ітеративної процедури. Вектор вхідних значень  $P$  визначений на інтервалі  $P = [-1, 1]$ . Вектор цілей  $T = [-0,9602 \ -0,577 \ 0,729 \ 0,3771 \ 0,6405 \ 0,66 \ 0,4609 \ 0,1336 \ 0,2113 \ 0,4344 \ 0,5 \ 0,393 \ 0,1647 \ 0,0988 \ 0,3072 \ 0,3960 \ 0,3449 \ 0,1816 \ 0,312 \ 0,2189 \ 0,3201]$ .

*Розв'язання:*

*Крок 1.* Ввести початкові дані та виконати їх графічну інтерпретація.

$$P = -1:1:1; \quad \dots$$

$$T = [-.9602 \ -.5770 \ -.0729 \ .3771 \ .6405 \ .6600 \ .4609 \ .1336 \ \dots$$

$$\quad \dots \ -.2013 \ -.4344 \ -.5000 \ -.3930 \ -.1647 \ .0988 \ .3072 \ .3960$$

```
.3449 .1816 -.0312 -.2189 -.3201];
plot(P,T,'*r','MarkerSize',4,'LineWidth',2), hold on
```

*Крок 2.* Задати допустимі значення помилки.

```
GOAL = 0.01;
```

*Крок 3.* Створити радіальну штучну нейронну мережу.

```
net = newrb(P,T,GOAL);
```

**NEWRB, neurons = 0, MSE = 0.176192**

*Крок 4.* Налаштувати кількість нейронів у прихованому шарі.

```
net.layers{1}.size
```

**ans = 6**

*Крок 5.* Виконати моделювання мережі.

```
V = sim(net,P);
```

```
plot(P,V,'ob','MarkerSize',5,'LineWidth',2)
```

```
p = [-0.75 -0.25 0.25 0.75];
```

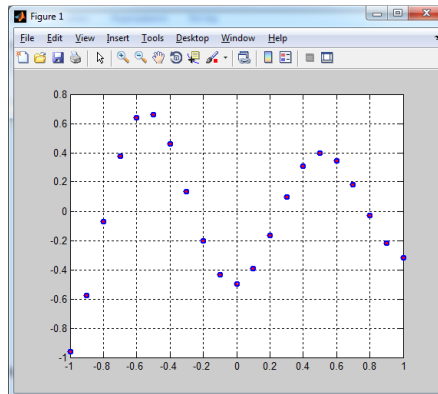


Рисунок 7.5 – Контрольні точки та навчальна множина

```
v = sim(net,p); % Новый вектор входа
```

```
plot(p,v,'+k','MarkerSize',10,'LineWidth',2), grid on
```

```
xlabel('P, p'), ylabel('T, v')
```

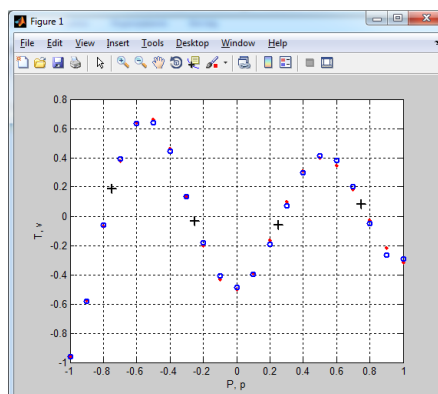


Рисунок 7.6 – Вектор входа із навчальної множини та навчальна множина

**Приклад 7.3.** В системі комп'ютерної математики Matlab створити та навчити радіально-базисну штучну нейронну мережу виконувати апроксимацію функціональної залежності однієї змінної. Вектор вхідних значень  $P$

визначений на інтервалі  $P = [-1,1]$ . Вектор цілей  $T = [-0,9602 \quad -0,577 \quad 0,0729 \quad 0,3771 \quad 0,6405 \quad 0,66 \quad 0,4609 \quad 0,1336 \quad 0,2113 \quad 0,4344 \quad 0,5 \quad 0,393 \quad 0,1647 \quad 0,0988 \quad 0,3072 \quad 0,3960 \quad 0,3449 \quad 0,1816 \quad 0,312 \quad 0,2189 \quad 0,3201]$ .

*Розв'язання:*

*Крок 0.* Представимо функції у вигляді полінома  $f(x) = \sum_{i=1}^N \alpha_i \varphi_i(x)$ , де  $\varphi_i(x)$

– радіальна базисна функція.

*Крок 1.* Ввести початкові дані та виконати їх графічну інтерпретацію.

```
clear,
p = -3:1:3;
a1 = radbas(p); % радіально-базисна функція 1
a2 = radbas(p - 1.5); % радіально-базисна функція 2
a3 = radbas(p + 2); % радіально-базисна функція 3
a = a1 + a2*1 + a3*0.5; % сума радіально-базисних функцій
figure(1), clf,
plot(p,a1,p,a2,p,a3*0.5,'LineWidth',1.5),hold on
plot(p,a,'LineWidth',3,'Color',[1/3,2/3,2/3]),grid on,
legend('a1','a2','a3*0.5','a')
```

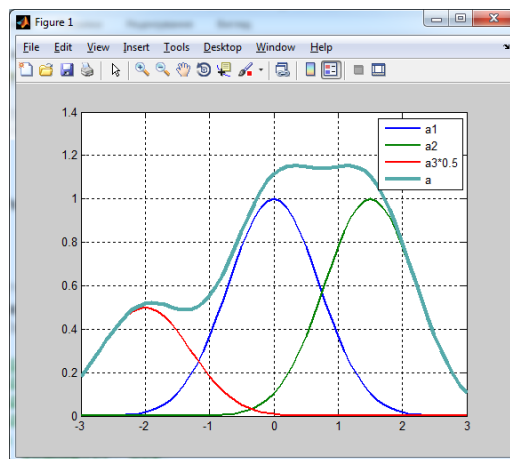


Рисунок 7.7 – Графік функції

*Крок 2.* Ввести навчальну множину, вектор цілей, середньоквадратичну помилку та значення параметру впливу.

```
P = -1:1:1;
T = [-.9602 -.5770 -.0729 .3771 .6405 .6600 .4609 .1336 ...
      -.2013 -.4344 -.5000 -.3930 -.1647 .0988 .3072 .3960 ...
      .3449 .1816 -.0312 -.2189 -.3201];
```

```
GOAL = 0.01;
```

```
SPREAD = 1;
```

*Крок 3.* Створити нейронну мережу.

```
net = newrb(P,T,GOAL,SPREAD);
```

*Крок 4.* Налаштувати кількість нейронів радіальної мережі у прихованому шарі

```
net.layers{1}.size
```

*Крок 5.* Виконати моделювання створеної нейронної мережі.

```
figure(1), clf,  
plot(P,T,'sr','MarkerSize',8,'MarkerFaceColor','y')  
hold on;  
X = -1:.01:1;  
Y = sim(net,X);  
plot(X,Y,'LineWidth',2), grid on %
```

Моделюючи нейронну мережу, створили апроксимаційну криву на інтервалі [-1 1] з кроком 0.01 для нелінійної залежності.

*Крок 6.* Дослідити вплив параметра SPREAD на структуру радіально-базисної мережі та якість апроксимації.

```
GOAL = 0.01;  
SPREAD = 0.01;  
net = newrb(P,T,GOAL,SPREAD);  
net.layers{1}.size  
NEWRB, neurons = 0, SSE = 2.758  
ans = 19
```

У наведеному прикладі параметр впливу SPREAD має значення 0,01. Це значить, що діапазон перекриття вхідних значень складає лише  $\pm 0,01$ . Оскільки елементи навчальної множини задані з інтервалом 0,1, то вхідні значення функціями активації не перекриваються.

Це призводить до того, що по-перше, збільшується кількість нейронів прихованого шару з 6 до 19, а по-друге, не забезпечується необхідна гладкість апроксимуючої функції. Зобразимо графічно отримані результати.

```
figure(2), clf,  
plot(P,T,'sr','MarkerSize',8,'MarkerFaceColor','y')  
hold on;  
X = -1:.01:1;  
Y = sim(net,X);  
plot(X,Y,'LineWidth',2), grid on
```

Тепер надамо параметру SPREAD довільне значення 12. В такому випадку усі функції активації перекриваються і кожний базовий нейрон видає значення, близьке до 1, для всіх вхідних значень. Функція newrb буде намагатися побудувати мережу, але не зможе забезпечити необхідну точність внаслідок складності обчислень.

```
GOAL = 0.01; SPREAD = 12;  
net = newrb(P,T,GOAL,SPREAD);  
net.layers{1}.size  
figure(3), clf,  
plot(P,T,'sr','MarkerSize',8,'MarkerFaceColor','y')  
hold on;
```



```
X = -1:.01:1;  
Y = sim(net,X);  
plot(X,Y,'LineWidth',2), grid on
```

Таким чином, значення параметру впливу **SPREAD** повинно бути більшим, ніж крок розбиття інтервалу заданої навчальної послідовності, але меншим розміру самого інтервалу. Для даної задачі це значить, що значення параметру впливу має бути більше 0.1, але менше 2.

Generalized regression neural network – це модифікована радіально-базисна штучна нейронна мережа, що швидко навчається. У другому шарі мережі Generalized regression neural network обчислюється нормований скалярний добуток матриці ваг і вектору входів.

Для побудови радіальної нейронної мережі, що швидко навчається, існує функція **newgrnn**, що у загальному випадку має такий синтаксис:

```
net = newgrnn(P, T, SPREAD),
```

де

*P* – матриця вхідних факторів, що подається вектором-стовпцем, кожний елемент якого є вектором-рядком і відповідає навчальному образу;

*T* – матриця реальних вихідних величин (цільових значень), що подається вектором-стовпцем, кожний елемент якого є вектором-рядком і однозначно відображає реальні вихідні величини для заданого навчального образу;

*SPREAD* – параметр впливу.

**Приклад 7.4.** В системі комп'ютерної математики Matlab створити та навчити радіально-базисну штучну нейронну мережу, що швидко навчається, виконувати апроксимацію функціональної залежності однієї змінної. Вектор вхідних значень  $P = [4 \ 5 \ 6]$ . Вектор цілей  $T = [1,5 \ 3,6 \ 6,7]$ .

*Крок 1.* Ввести початкові дані.

```
clear,  
P = [4 5 6];  
T = [1.5 3.6 6.7];
```

*Крок 2.* Створити нейронну мережу:

```
net = newgrnn(P,T);
```

*Крок 3.* Визначити кількість нейронів у прихованому шарі.

```
net.layers{1}.size  
ans = 3
```

*Крок 4.* Виконати моделювання мережі для одного входу.

```
p = 4.5;  
v = sim(net,p);
```

*Крок 5.* Виконати моделювання мережі для послідовності входів з інтервалу [4, 7].

```
p1 = 4:0.1:7;  
v1 = sim(net,p1);
```

Крок 6. Виконати графічну інтерпретація результатів моделювання.

```
figure(1), clf
plot(P,T,'*g',p,v,'or',p1,v1,'-b','MarkerSize',8,'LineWidth',2)
grid on
```

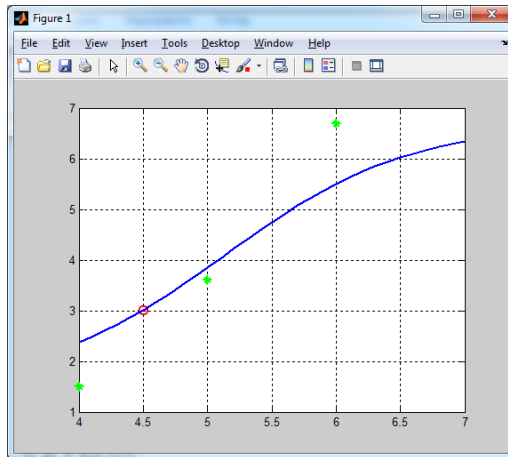


Рисунок 7.8 – Результати моделювання

Probabilistic neural network – це модифікована радіально-базисна нейронна мережа, що призначена для розв’язання ймовірнісних задач, зокрема, задач класифікації. В мережі Probabilistic neural network замість другого шару використовується «конкуруючий» шар, який обчислює ймовірність належності вхідного вектору до певного класу.

Для побудови ймовірнісної радіальної нейронної мережі, що швидко навчається, існує функція **newpnn**, яка має такий синтаксис:

$$\text{net} = \text{newpnn}(P, T),$$

де

$P$  – матриця вхідних факторів, що подається вектором-стовпцем, кожний елемент якого є вектором-рядком і відповідає навчальному образу;

$T$  – матриця реальних вихідних величин (цільових значень), що подається вектором-стовпцем, кожний елемент якого є вектором-рядком і однозначно відображає реальні вихідні величини для заданого навчального образу.

**Приклад 7.5.** В системі комп’ютерної математики Matlab створити та навчити ймовірнісну радіально-базисну штучну нейронну мережу, що швидко навчається, виконувати апроксимацію функціональної залежності. Вектор вхідних значень  $P = [0 \ 0; 1 \ 1; 0 \ 3; 1 \ 4; 3 \ 1; 4 \ 1; 4 \ 3]$ . Вектор цілей  $T_c = [1 \ 1 \ 2 \ 2 \ 3 \ 3 \ 3]$ .

Крок 1. Ввести початкові дані.

```
clear,
```

```
P = [0 0; 1 1; 0 3; 1 4; 3 1; 4 1; 4 3];
```

```
Tc = [1 1 2 2 3 3 3]; % вектор індексів класів
```

Крок 2. Встановити відповідність між векторами та класами у вигляді матриці зв’язності та повної матриці

```
T = ind2vec(Tc) % матриця зв’язності
```

```
T =
```

```
(1,1) 1
(1,2) 1
(2,3) 1
(2,4) 1
(3,5) 1
(3,6) 1
(3,7) 1
```

T = full(T)% повна матриця

```
T = 1 1 0 0 0 0 0
    0 0 1 1 0 0 0
    0 0 0 0 1 1 1
```

*Крок 3.* Створити нейронну мережу.

```
net = newpnn(P,T);
```

*Крок 4.* Налаштувати кількість нейронів у прихованому шарі.

```
net.layers{1}.size
```

```
ans = 7
```

*Крок 5.* Виконати перетворення матриці зв'язності в індексний вектор.

```
Y = sim(net,P);
```

```
Yc = vec2ind(Y)
```

```
Yc = 1 1 2 2 3 3 3
```

*Крок 6.* Виконати тестовий прогін нейронної мережі на векторі, що не входить до навчальної множини.

```
p = [1 3; 0 1; 5 2]; % вектор, який не входить до навчальної множини
```

*Крок 7.* Виконати моделювання мережі.

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

```
ac = 2 1 3
```

*Крок 8.* Виконати графічну інтерпретацію результатів моделювання.

```
figure(1), clf reset, drawnow
```

```
p1 = 0:.05:5; p2 = p1;
```

```
[P1,P2]=meshgrid(p1,p2); pp = [P1(:) P2(:)];
```

```
aa = sim(net,pp'); aa = full(aa);
```

```
m = mesh(P1,P2,reshape(aa(1,:),length(p1),length(p2)));
```

```
set(m,'facecolor',[0.75 0.75 0.75],'LineStyle','none');
```

```
hold on, view(3)
```

```
m = mesh(P1,P2,reshape(aa(2,:),length(p1),length(p2)));
```

```
set(m,'FaceColor',[0 1 0.5],'LineStyle','none');
```

```
m = mesh(P1,P2,reshape(aa(3,:),length(p1),length(p2)));
```

```
set(m,'FaceColor',[0 1 1],'LineStyle','none');
```

```
plot3(P(1,:),P(2,:),ones(size(P,2))+0.1,'.','MarkerSize',20)
```

```
plot3(p(1,:),p(2,:),1.1*ones(size(p,2)),'*','MarkerSize',10,...  
'Color',[1 0 0]), hold off, view(2)
```

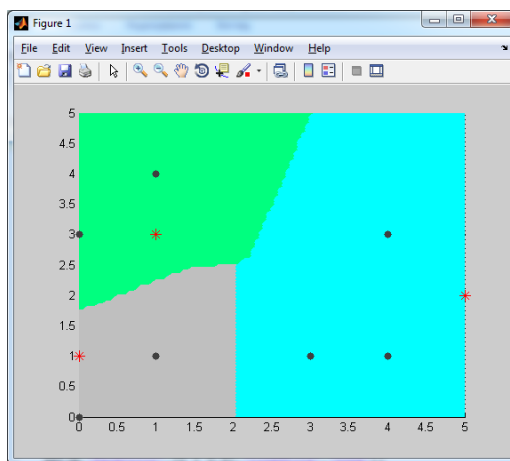


Рисунок 7.9 – Результати моделювання

### Завдання до роботи

1. В системі комп'ютерної математики Matlab створити та навчити радіально-базисну штучну нейронну мережу виконувати апроксимацію функціональної залежності. Вектор вхідних значень визначений на інтервалі  $P = 1 \cdot k : 3 \cdot k$ , вектор цілей  $T = [0 \ 2 \ 4.1 \ 5.9 \ 7.3 \ 8.7 \ 10.1 \ 11.8 \ 12.5 \ 14.2] \cdot k$ , де  $k$  – номер варіанта. Проаналізувати структуру побудованої мережі та вплив параметрів обчислювальної моделі на якість навчання нейронної мережі.

2. В системі комп'ютерної математики Matlab створити та навчити радіально-базисну штучну нейронну мережу виконувати апроксимацію функціональної залежності. Вектор вхідних значень визначений на інтервалі  $P = 1 \cdot k : 5 \cdot k$ , вектор цілей  $T = [0.3 \ 1.9 \ 4.1 \ 5.5 \ 7.32 \ 8.7 \ 10.1 \ 11.8 \ 12.5 \ 14.2] \cdot k$ , середня квадратична помилка може набути значень  $0.0179 \cdot k$  та  $0.836$ , де  $k$  – номер варіанта. Проаналізувати структуру побудованої мережі та вплив параметрів обчислювальної моделі на якість навчання нейронної мережі.

3. В системі комп'ютерної математики Matlab створити та навчити радіально-базисну штучну нейронну мережу, що швидко навчається, виконувати апроксимацію функціональної залежності. Вектор вхідних значень та вектор цілей наведені в табл. 7.1. Значення параметру впливу може набувати значень 0.1, 1, 10. Проаналізувати структуру побудованої мережі та вплив параметрів обчислювальної моделі на якість навчання нейронної мережі.

Таблиця 7.1 – Початкові дані

| Варіант | Значення вектору входів                        | Значення вектору індексів класів  |
|---------|--|-----------------------------------|
| 1       | $[-0,9 \ -0,6 \ -0,3 \ 0,3 \ 0,6 \ 0,9]$       | $[3 \ 2 \ -1 \ 1 \ -1 \ 2 \ 1]$   |
| 2       | $[-0,8 \ -0,5 \ -0,2 \ 0,1 \ 0,4 \ 0,7 \ 1]$   | $[-3 \ -2 \ 1 \ -1 \ 2 \ 1 \ -1]$ |
| 3       | $[-0,7 \ -0,4 \ -0,1 \ 0,2 \ 0,5 \ 0,8 \ 1,1]$ | $[0 \ -3 \ -1 \ 2 \ 0 \ 3 \ 1]$   |
| 4       | $[-0,6 \ -0,4 \ -0,2 \ 0 \ 0,2 \ 0,4 \ 0,6]$   | $[0 \ -3 \ -1 \ 2 \ 0 \ 3 \ 1]$   |
| 5       | $[-0,5 \ -0,3 \ -0,1 \ 0,1 \ 0,3 \ 0,5 \ 0,7]$ | $[1 \ 3 \ -3 \ 2 \ 2 \ 1 \ -1]$   |
| 6       | $[-0,5 \ -0,2 \ 0,1 \ 0,4 \ 0,7 \ 1,3]$        | $[2 \ 3 \ 1 \ 2 \ 0 \ -1 \ -2]$   |
| 7       | $[-1,5 \ -1 \ -0,5 \ 0 \ 0,5 \ 1 \ 1,5]$       | $[-2 \ -3 \ -1 \ 2 \ 1 \ 2 \ 3]$  |

Продовження табл. 7.1

| Варіант | Значення вектору входів           | Значення вектору індексів класів |
|---------|-----------------------------------|----------------------------------|
| 8       | [-1 -1 -0,7 -0,3 0,1 0,5 0,9 1,3] | [-3 -5 1 0 1 -2 -3]              |
| 9       | [0 0,3 0,6 0,9 1,2 1,5 1,8 2,1]   | [-3 -2 1 0 1 -3 -1]              |
| 10      | [-1,2 -0,8 -0,4 0 0,4 0,8 1,2]    | [0 3 -1 0 3 2 -1]                |
| 11      | [-0,6 -0,4 -0,2 0 0,2 0,4 0,6]    | [3 2 -1 1 -1 2 1]                |
| 12      | [-0,5 -0,3 -0,1 0,1 0,3 0,5 0,7]  | [-3 -2 1 -1 2 1 -1]              |
| 13      | [-0,5 -0,2 0,1 0,4 0,7 1,3]       | [3 2 -1 1 -1 2 1]                |
| 14      | [-2,5 -2 -1,5 0 1,5 2 2,5]        | [-3 -2 1 -1 2 1 -1]              |
| 15      | [-0,2 -1,8 -1,4 2 1,4 1,8 1,2]    | [6 4 -2 1 -3 2 1]                |

4. В системі комп'ютерної математики Matlab створити та навчити ймовірнісну радіально-базисну штучну нейронну мережу виконувати апроксимацію функціональної залежності. Вектор входних значень та вектор цілей наведені в табл. 7.2. Проаналізувати структуру побудованої мережі та вплив параметрів обчислювальної моделі на якість навчання нейронної мережі.

Таблиця 7.2 – Початкові дані

| Варіант | Значення вектору входів             | Значення вектору індексів класів |
|---------|-------------------------------------|----------------------------------|
| 1       | [2 3; 1 4; 0 1; 1 1; 2 0; 4 2; 5 3] | [2 2 1 1 1 3 3]                  |
| 2       | [2 1; 1 0; 1 3; 2 4; 0 5; 4 2; 5 0] | [1 1 1 2 2 3 3]                  |
| 3       | [4 0; 5 1; 3 1; 2 0; 1 1; 2 4; 1 1] | [3 3 3 1 1 2 2]                  |
| 4       | [1 3; 2 4; 0 5; 4 1; 3 3; 0 0; 1 1] | [2 2 2 3 3 1 1]                  |
| 5       | [0 1; 1 0; 3 2; 4 0; 5 2; 1 3; 2 4] | [1 1 3 3 3 2 2]                  |
| 6       | [1 3; 2 4; 1 1; 0 0; 4 2; 3 1; 5 3] | [1 1 2 2 3 3 3]                  |
| 7       | [3 2; 4 1; 1 5; 2 4; 0 3; 1 1; 2 2] | [3 3 2 2 2 1 1]                  |
| 8       | [2 5; 1 3; 1 2; 0 1; 1 0; 3 2; 4 1] | [2 2 1 1 1 3 3]                  |
| 9       | [0 1; 1 0; 2 2; 4 2; 5 3; 2 4; 1 5] | [1 1 1 3 3 2 2]                  |
| 10      | [3 2; 4 0; 5 1; 2 4; 0 3; 1 1; 0 0] | [3 3 3 2 2 1 1]                  |
| 11      | [4 0; 5 1; 3 1; 2 0; 1 1; 2 4; 1 1] | [2 2 1 1 1 3 3]                  |
| 12      | [1 3; 2 4; 1 1; 0 0; 4 2; 3 1; 5 3] | [1 1 1 2 2 3 3]                  |
| 13      | [2 1; 1 0; 1 3; 2 4; 0 5; 4 2; 5 0] | [1 1 3 3 3 2 2]                  |
| 14      | [3 2; 4 1; 1 5; 2 4; 0 3; 1 1; 2 2] | [2 2 2 3 3 1 1]                  |
| 15      | [2 3; 1 4; 0 1; 1 1; 2 0; 4 2; 5 3] | [1 1 3 3 3 2 2]                  |

**Зміст звіту**

1. Титульний аркуш.
2. Тема і мета роботи.
3. Короткі теоретичні відомості.
4. Протокол виконання завдання №1.
5. Протокол виконання завдання №2.

6. Протокол виконання завдання №3.
7. Протокол виконання завдання №4.
8. Висновки.

### **Контрольні питання**

1. Опишіть структуру радіально-базисної штучної нейронної мережі.
2. Наведіть приклади базисних функцій, що використовуються при побудові радіально-базисної штучної нейронної мережі.
3. Опишіть алгоритм навчання радіально-базисної штучної нейронної мережі.
4. Які проблеми супроводжує використання радіально-базисної штучної нейронної мережі?
5. Вкажіть переваги та недоліки радіально-базисної штучної нейронної мережі.
6. Опишіть процес створення радіально-базисної штучної нейронної мережі в системі комп'ютерної математики Matlab.
7. Опишіть процес навчання радіально-базисної штучної нейронної мережі, що швидко навчається, в системі комп'ютерної математики Matlab.
8. Опишіть процес навчання ймовірнісної радіально-базисної штучної нейронної мережі в системі комп'ютерної математики Matlab.
9. Як впливає складність розв'язуваної задачі на складність реалізації радіально-базисної штучної нейронної мережі?