Практичне заняття 2

Дослідження мереж RBF

Мета. Вивчити архітектуру і процес навчання радіальних базисних мереж.

Завдання

1. Створити базисну мережу з нульовою помилкою.
2. Створити нейронну мережу радіального типу для апроксимації y=sin(x)/x.
3. Створити графіки радіальних базисних функцій, визначити зважену суму даних функцій.

Контрольні питання

1. Яку архітектуру має нейронна мережа радіального типу?

2.Яка функції використовуються при навчанні радіальної базисної мережі?

3. Для вирішення яких практичних завдань використовується радіальна мережа?

4. Яким чином виконується навчання радіальної базисної мережі з нульовою помилкою?

5 .Як послідовність навчання радіальної мережі?

6 .Для яких цілей задається зсув мережі?

7. Що розуміють під радіальними базисними функціями?

8. Які функції використовує радіальний і лінійний шари радіальної мережі?

9. Яким чином проводиться навчання радіальних базисних мереж?

10. Які параметри використовуються при завданні функції newrbe?

11. Навчання радіальних базисних мереж проводиться з учителем або без учителя?

12. При вирішенні яких завдань використовуються радіальні базисні мережі?

Теоретичні відомості

Радіальна базисна мережа (Radial Basis Function, RBF-мережа) складається з двох шарів. Перший шар включає нейрони, що використовують радіальні функції radbas. Функції обчислюють відстань за допомогою функції dist між нейронами мережі в під час навчання і вхідними векторами на основі відстані (рис. 1).

При цьому відомості про розташування нейронів на кожній ітерації зберігаються в рядках матриці ваг net.IW {1,1}. Другий шар радіальної мережі включає нейрони з лінійною функцією активації purelin.

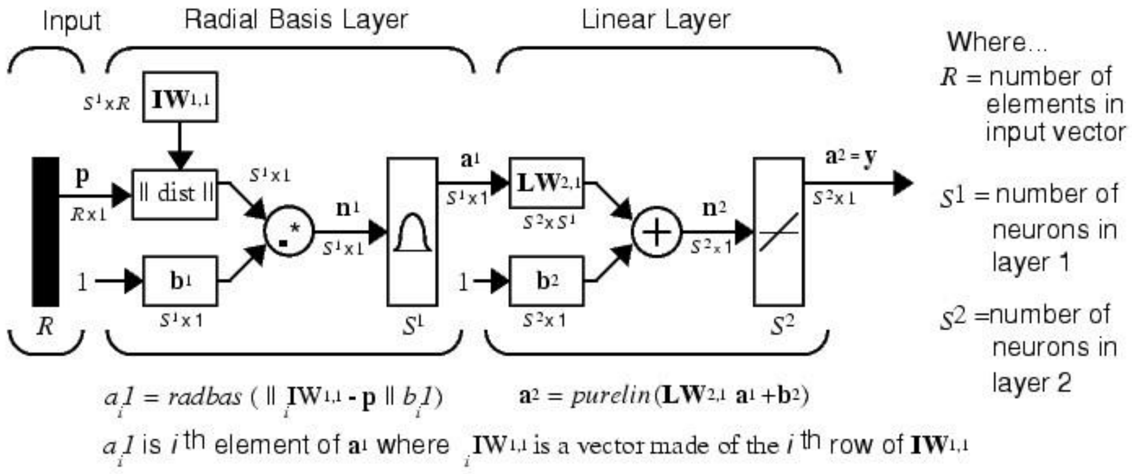

Рисунок 1 – Функції активації при навчанні радіальних базисних мереж

Для створення RBF-мережі застосовується функція newrb, синтаксис якої наведено нижче:

net = newrb (P, T, GOAL, SPREAD, MN, DF),

де P - матриця R x Q з Q вхідних векторів розмірністю R;

T – матриця S x Q з Q цільових векторів розмірністю S;

GOAL – максимально допустима помилка;

SPREAD – параметр згладжування радіальних функцій;

MN – максимально допустима кількість нейронів в першому шарі;

DF – кількість нейронів, що додаються на кожному кроці.

При створенні нейронної мережі необхідно обов’язково вказувати перші два параметри, інші параметри визначаються за замовчуванням.

Радіальна базисна мережа складається з двох шарів:

– одного прихованого шару, який називають шаром радіальних нейронів, або радіальним базисним шаром, що включає S1 нейронів;

– вихідного шару нейронів, що включає S2 нейрони з лінійною функцією активації, даний шару називають лінійним шаром (рис. 1).

Вектор вхідних даних, як правило, позначається через p. При обчисленні відстаней в блоці ||dist|| використовується вектор p і матриця вагових коефіцієнтів $IW^{1,1}$.

Результатом обчислень блоку ||dist|| є вектор, що складається з S1 елементів, для яких визначаються відстані між i-м вектором входу p і i-й вектор-рядком $IW^{1,1}$ матриці ваг. Даний вектор-рядок називають вектором ваг i-го нейрона. Вихід блоку ||dist|| множиться поелементно на вектор зміщення b^1 і формує вхід аргументу функції активації. В результаті, вихідні значення першого шару нейронної мережі записуються в вигляді:

a = radbas (net.prod (dist (net.IW, p), net.b)))

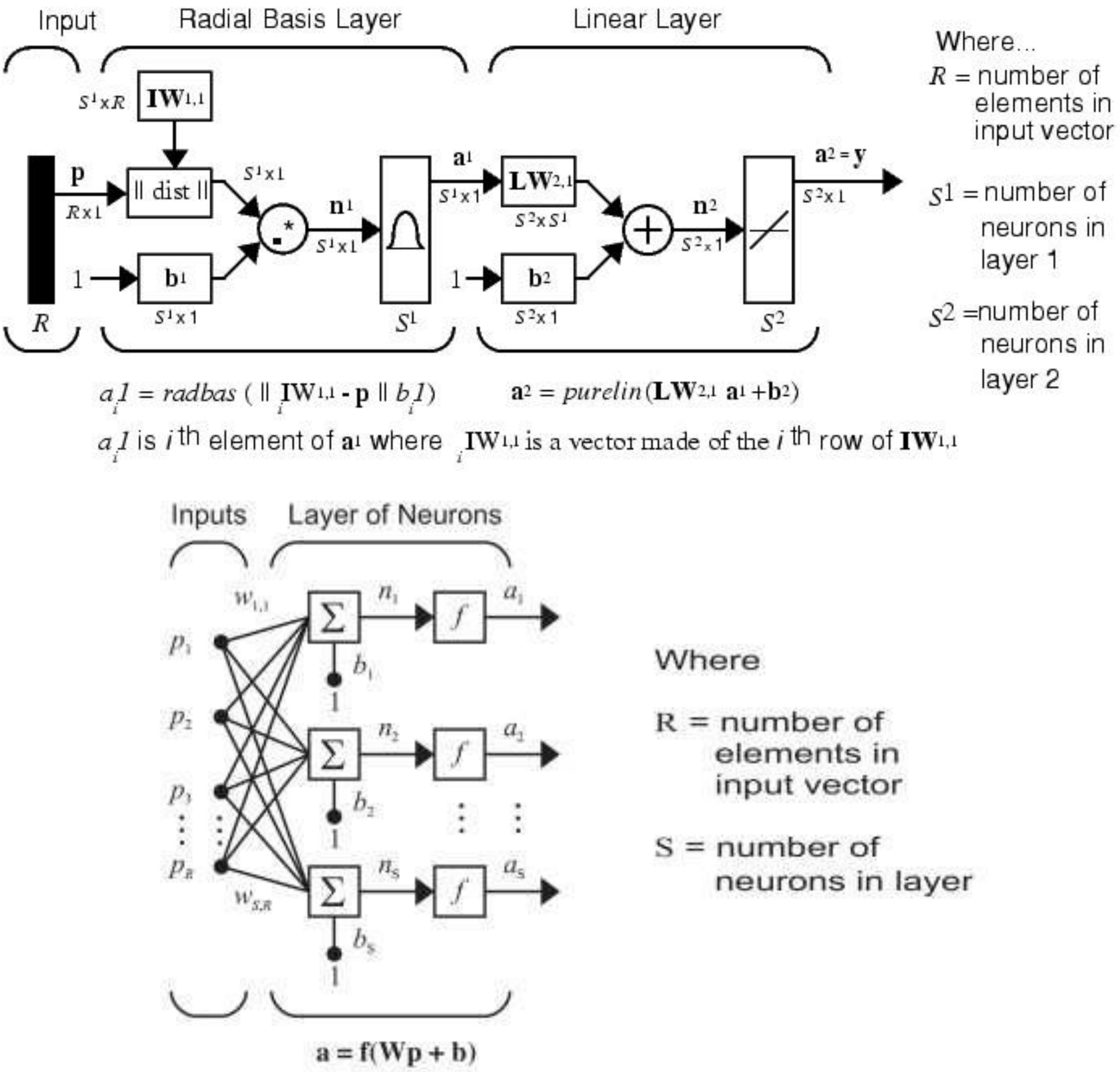

Рисунок 2 – Архітектура радіальної базисної мережі

Радіальна базисна функція має максимум, рівний 1, в тому випадку, якщо вхід мережі дорівнює 0. Радіальний базисний нейрон реагує подібно індикатору, формує на виході значення 1, якщо вхід мережі p буде ідентичний вектору вагових коефіцієнтів. У тому випадку якщо, відстань між векторами входів і вагами зменшується, то вихід радіальної базисної функції збільшується. У тому випадку, якщо тільки один радіальний базисний нейрон має вихідне значення, рівне 1, а всі інші нейрони мають виходи, рівні або дуже близькі до 0, то вихідні значення на другому шарі мережі будуть рівні вагам активного вихідного нейрона. Це рідкісний випадок, зазвичай на вихідне значення мережі впливають кілька нейронів з різними ваговими коефіцієнтами.

Зсув нейронної мережі використовується для коригування чутливості нейрона за допомогою функції radbas. Наприклад, якщо нейрон має зсув, що дрівнює 0,1, то його вихід буде дорівнювати 0,5 для будь-якого вектора входу p і вага нейрона w при відстані між векторами, що дорівнює 10.333 складе 10.833/b.

Створення базисної мережі з нульовою помилкою.

Для побудови радіальних базисних мереж призначені М-функції newrbe і newrb. Перша функція дозволяє побудувати радіальну базисну мережу з нульовою помилкою, друга дозволяє керувати кількістю нейронів прихованого шару. Функція newrbe викликається наступним чином:

net = newrbe (P, T, SPREAD)

Вхідними аргументами функції newrbe є масиви вхідних векторів P і цілей T, а також параметр впливу SPREAD. Вона повертає радіальну базисну мережу з такими вагами і зсувами, що її виходи точно дорівнюють цілям T. Функція newrbe створює стільки нейронів радіального базисного шару, скільки є вхідних векторів в масиві P і встановлює ваги першого шару рівними P'.

При цьому зсуви встановлюються рівними 0.8326/SPREAD. Це означає, що рівень перекриття радіальних базисних функцій дорівнює 0.5 і всі входи в діапазоні ± SPREAD вважаються значущими. Чим більший діапазон вхідних значень повинен бути прийнятий до уваги, тим більше значення параметра впливу SPREAD має бути встановлено (див. Додаток А – Демонстраційна програма demorb1).

Ваги другого шару IW_{21} і зсувів b_2 (b) можуть бути знайдені шляхом моделювання виходів першого шару a1 (A) і подальшого вирішення системи лінійних алгебраїчних рівнянь (СЛАР):

[W b] * [A; ones] = T

Оскільки відомі входи другого шару A і цілі T, а шар лінійний, то для обчислення ваг і зсувів другого шару досить скористатися розв'язувачем СЛАР:

Wb = T / [P; ones (1, size (P, 2))]

Матриця Wb містить ваги і зміщення. Сума квадратів похибок мережі завжди дорівнює 0, тому що маємо задачу з Q рівняннями (пари вхід/мета) і кожен нейрон має Q +1 змінних (Q ваг по числу радіальних базисних нейронів і одне зміщення). СЛАР з Q рівняннями і більш ніж Q змінними має вільні змінні і характеризується нескінченним числом рішень з нульовою похибкою.

Приклад 1. Використання радіальної функції активації в MATLAB при зміні змінної x від -5 до 5 з кроком 0,1.

```
>> x = -5: .1: 5;
>> a = radbas (x);
>> plot (x, a)
>> title ( 'Радіальна базисна функція', 'FontSize', 16);
>> xlabel ( 'Вхідні дані', 'FontSize', 16);
>> ylabel ( 'Вихідні дані', 'FontSize', 16);
>> grid on
```

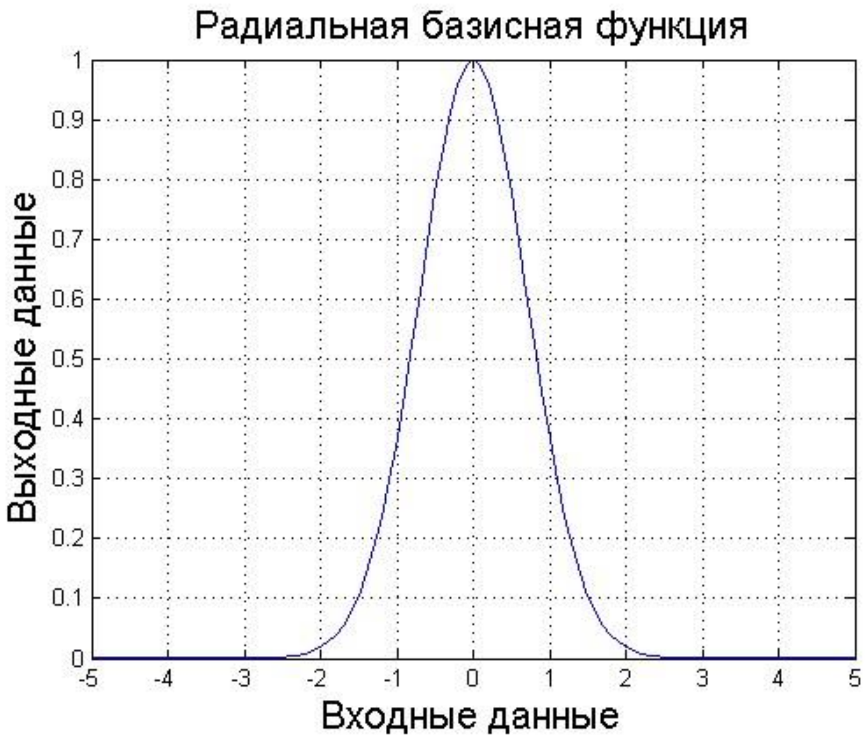

Рисунок 3 – Радіальна базисна функція

Вхідні дані для виконання завдання необхідно внести в табл. 1:

P – вектор вихідних даних змінюється -2 до 1,4 з кроком 0,2

T – вектор цілей обчислюється за формулою

$$T = \cos(P + 0.05 * N) + 0.04, \quad (2)$$

Таблиця 1 – Вхідні дані для виконання завдання

P	-2	-1,8	-1,6	-1,4	-1,2	-1,0	-0,8	-0,6	-0,4	-0,2	0	0,2	0,4	0,6	0,8	1,0	1,2	1,4
T																		

Приклад 2. Створення радіальної базисної мережі з нульовою помилкою

```
>> P = [0.2 1 1.8 2.6 3.4 4.2 5 5.8 6.6 7.4 8.2]
>> T = [1.18006 0.74030 -0.02720 -0.65688 -0.76679 -0.29026
0.483662185 1.085519 1.15023 0.63854 -0.13915]
>> plot (P, T, '* r', 'MarkerSize', 4, 'LineWidth', 2)
hold on
grid on
>>% Створення радіальної базисної мережі
net = newrbe (P, T);
% Визначення числа нейронів в прихованому шарі
>> net.layers.size
% Ans = 21
% Моделювання мережі з використання вхідних і цільових даних навчальної вибірки
V = sim (net, P)
ans =
   [11]
```

[1]
% Отримані значення в результаті навчання нейронної мережі
V = 1.1801 0.7403 -0.0272 -0.6569 -0.7668 -0.2903 0.4837 1.0855
1.1502 0.6385 -0.1392
% Створення графіка з вхідними даними
>> plot (P, V, 'ob', 'MarkerSize', 5, 'LineWidth', 2)
% Завдання вектора проміжних точок у вигляді вектора вхідних даних
p = [0.6 1.6 2.3 4.0 6.5];
% Процес перенавчання нейронної мережі
v = sim (net, p)
plot (p, v, '+ k', 'MarkerSize', 10, 'LineWidth', 2)
grid on
% Отримання результатів навчання% побудова графіка і сітки на графіку
v =
1.0155 0.1722 -0.4663 -0.4509 1.1688

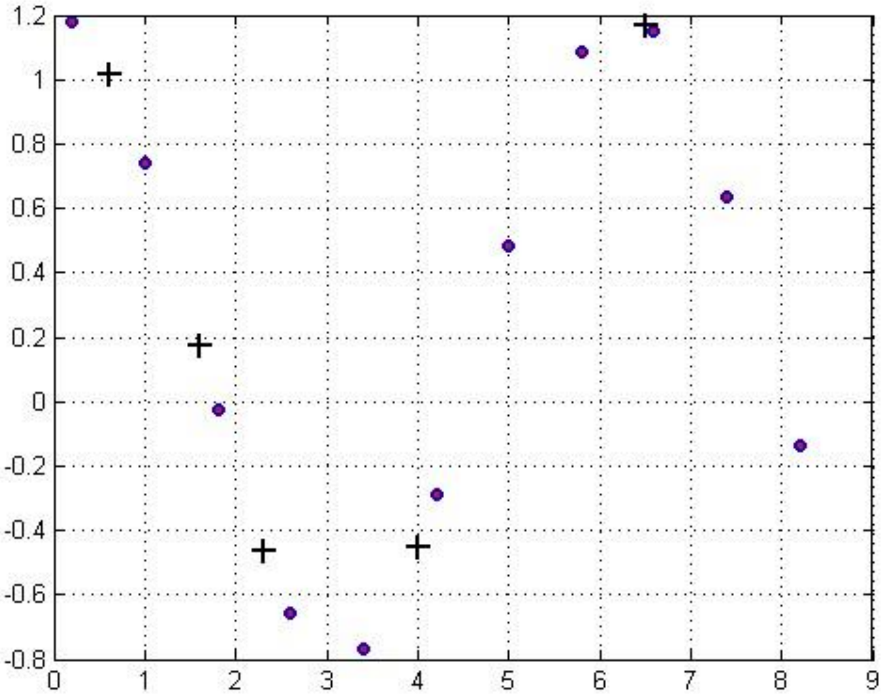

Рисунок 4 – Результати апроксимації функції за допомогою радіальної базисної мережі

Створення нейронної мережі радіального типу для апроксимації поверхні.

Вхідні дані для виконання завдання:

$z = (\cos (x \cdot \sin (y + 0{,}02 \cdot N) + 0{,}08 \cdot N)$

Змінні x, y змінюється на інтервалі від -3 до 3 з кроком 0,01 * N (N - номер варіанта завдання).

Приклад 3. Необхідно створити поверхню, заданої у вигляді формули, для якої змінні x, y змінюється на інтервалі від -3 до 3 з кроком 0,1:

$$z = (\cos (x \cdot \sin (y + 0{,}1) + 0{,}2). \quad (1)$$

Для цього потрібно задати вихідні дані, створити і навчити мережу, вивести результати навчання на екран і побудувати графік змодельованої поверхні (рис. 5). Програмний код MATLAB наведено нижче:

Приклад 3. Необхідно створити поверхню, описуваної в вигляді формули, для якої змінні x, y змінюється на інтервалі від -3 до 3 з кроком 0,1:

```
% Визначення вектора вхідних даних для навчання мережі
x = -3: 0.1: 3;
% Визначення вектора вихідних даних
% Радіальні базисні мережі навчаються тільки з учителем
y = -3: 0.1: 3;
% Завдання рівняння поверхні від двох змінних
z = cos (x'* sin (y + 0.1) +0.2);
% Визначення матриці входів і виходів нейронної мережі
P = [x; y];
T = z;
% Створення нейронної мережі радіального типу
net = newrb (P, T);
% Навчання мережі в залежності від обраної архітектури
Y = sim (net, P)
% Побудова тривимірної поверхні
mesh (x, y, z)
% Отримання помилки навчання для кожного нейрона і побудова поверхні
NEWRB, neurons = 0, MSE = 0.182572
NEWRB, neurons = 2, MSE = 0.101978
NEWRB, neurons = 3, MSE = 0.0165157
NEWRB, neurons = 4, MSE = 0.014556
NEWRB, neurons = 5, MSE = 0.00523354
NEWRB, neurons = 6, MSE = 0.00256723
NEWRB, neurons = 7, MSE = 0.00238156
NEWRB, neurons = 8, MSE = 0.00222853
NEWRB, neurons = 9, MSE = 3.7241e-05
NEWRB, neurons = 10, MSE = 1.64223e-05
NEWRB, neurons = 11, MSE = 8.21751e-06
NEWRB, neurons = 12, MSE = 3.57725e-06
NEWRB, neurons = 13, MSE = 2.89488e-06
NEWRB, neurons = 14, MSE = 2.48158e-07
NEWRB, neurons = 15, MSE = 4.84938e-08
NEWRB, neurons = 16, MSE = 2.15415e-08
NEWRB, neurons = 17, MSE = 1.50764e-08
NEWRB, neurons = 18, MSE = 5.54992e-09
NEWRB, neurons = 19, MSE = 4.55362e-09
NEWRB, neurons = 20, MSE = 4.36135e-11
NEWRB, neurons = 21, MSE = 3.36148
```

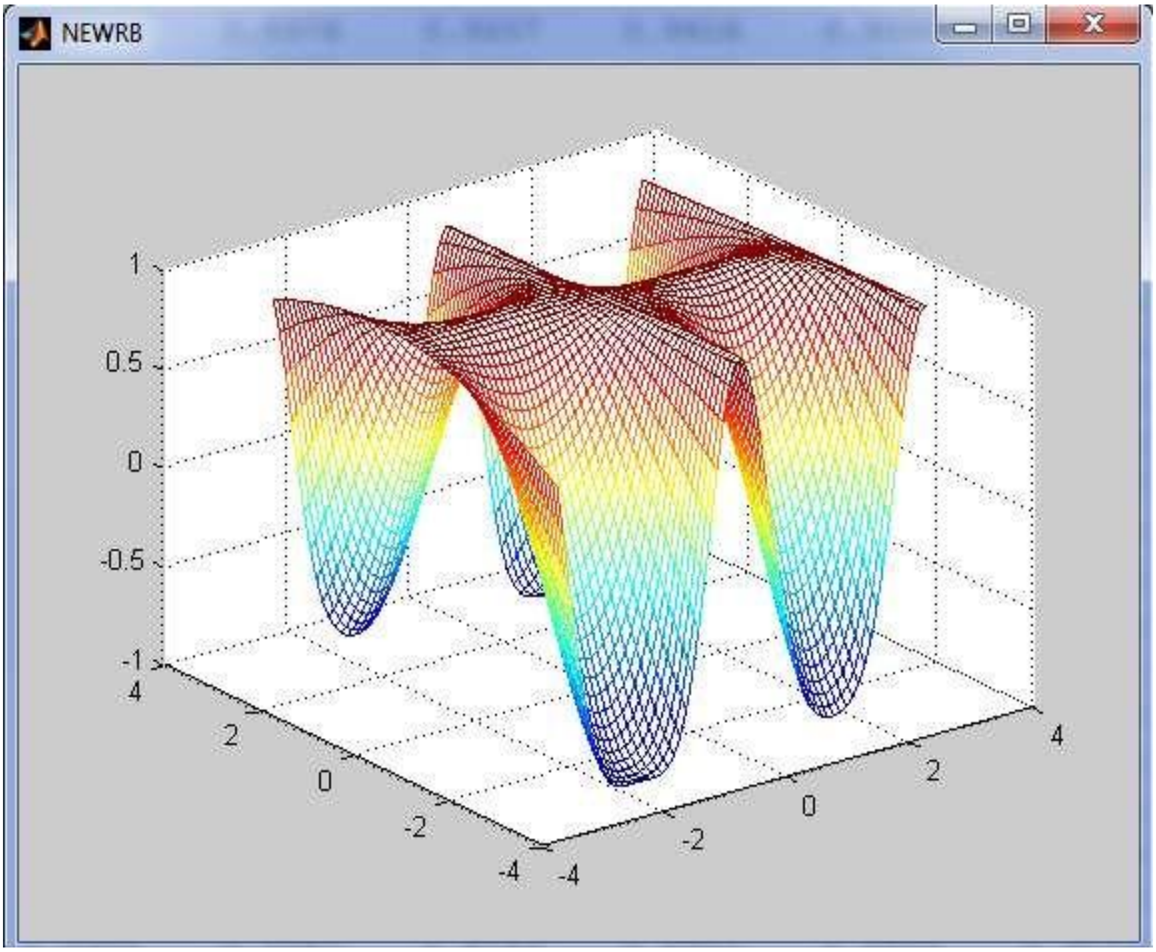

Рис.5. Результат моделювання нейронної мережі і побудова поверхні функції

Створення графіків радіальних базисних функцій, визначення зваженої суми даних функцій.

Вихідні дані для виконання завдання (N - номер варіанта завдання):

```
p = -3 * N: 0.2: 3 * N;
a1 = radbas (p);
a2 = radbas (p-0.5 * N);
a3 = radbas (p + 3 * N);
a = a1 + a2 * 1 + a3 * 0.5;
```

Приклад 4. Програмний код MATLAB для створення графіків радіальних базисних функцій.

```
p = -3: 0.1: 3;
a1 = radbas (p);
a2 = radbas (p-1.5);
a3 = radbas (p + 2);
a = a1 + a2 * 1 + a3 * 0.5;
plot (p, a1, p, a2, p, a3 * 0.5, p, a)
```

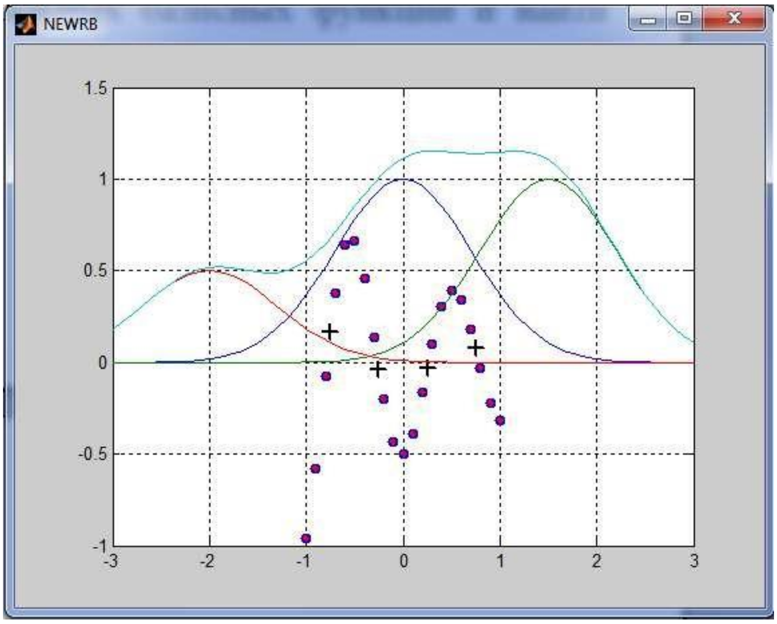

Рис.6. Графіки радіальних базисних функцій

Додаток А – Демонстраційна програма demorb1

```
%% Radial Basis Approximation
%Program Files\MATLAB\toolbox\nnet\nndemos\demorb1.m*
% This demo uses the NEWRB function to create a radial basis network that
% approximates a function defined by a set of data points.
%
% Copyright 1992-2005 The MathWorks, Inc.
% $Revision: 1.14.2.1 $  $Date: 2005/11/15 01:15:20 $

% Define 21 inputs P and associated targets T.

P = -1:.1:1;
T = [-.5602 -.4770 -.0729  .3771  .6405  .6600  .4609 ...
      .1336 -.2013 -.4344 -.5000 -.3930 -.1647  .0988 ...
      .3072  .3960  .3449  .1816 .0312 -.2189 -.3201];
plot(P,T,'+');
title('Training Vectors');
xlabel('Input Vector P');
ylabel('Target Vector T');

% We would like to find a function which fits the 21 data points.  One way to do
% this is with a radial basis network.  A radial basis network is a network with
% two layers.  A hidden layer of radial basis neurons and an output layer of
% linear neurons.  Here is the radial basis transfer function used by the hidden
% layer.

p = -3:.1:3;
a = radbas(p);
plot(p,a)
title('Radial Basis Transfer Function');
xlabel('Input p');
ylabel('Output a');

%%
% The weights and biases of each neuron in the hidden layer define the position
% and width of a radial basis function.  Each linear output neuron forms a
% weighted sum of these radial basis functions.  With the correct weight and
% bias values for each layer, and enough hidden neurons, a radial basis network
% can fit any function with any desired accuracy.  This is an example of three
% radial basis functions (in blue) are scaled and summed to produce a function
% (in magenta).

a2 = radbas(p-1.5);
a3 = radbas(p+2);
a4 = a + a2*1 + a3*0.5;
```

```
plot(p,a,'b-',p,a2,'b--',p,a3,'b--',p,a4,'m-')
title('Weighted Sum of Radial Basis Transfer Functions');
xlabel('Input p');
ylabel('Output a');

%%
% The function NEWRB quickly creates a radial basis network which
approximates
% the function defined by P and T.  In addition to the training set and targets,
% NEWRB takes two arguments, the sum-squared error goal and the spread
constant.

eg = 0.02; % sum-squared error goal
sc = 1;    % spread constant
net = newrb(P,T,eg,sc);

%%
% To see how the network performs, replot the training set.  Then simulate the
% network response for inputs over the same range.  Finally, plot the results on
% the same graph.

plot(P,T,'+');
xlabel('Input');

X = -1:.01:1;
Y = sim(net,X);

hold on;
plot(X,Y);
hold off;
legend({'Target','Output'})


displayEndOfDemoMessage(mfilename)
```