Практичне заняття 4
Дослідження мережі Хопфілда

Мета роботи: отримати практичні навички рішення задачі розпізнавання образів з застосуванням нейромережі Хопфілда у пакеті MATLAB.

Завдання до роботи

1. Вивчити функцію **newhop()** на прикладі з 4 нейронами.
2. Згідно з варіантом (табл.4.1) створити мережу для розпізнавання трьох символів з файлу Alphabet.csv (аналог набору **prprob** MATLAB, символи A-Z) .
3. Подати спотворені символи з завдання на мережу Хопфілда з різним ступенем спотворення, зробити висновки з точності розпізнавання.
4. Подати символ, що не увійшов в начальну вибірку, зробити висновки з результату розпізнавання.

Таблиця 4.1 – Варіанти завдань

№ з/п	Символи	№ з/п	Символи
1.	B, D, Y	11.	C, H, I
2.	F, U, M	12.	E, K, V
3.	N, D, X	13.	P, Q, Z
4.	L, O, S	14.	T, W, A
5.	G, J, Z	15.	S, T, O
6.	C, I, Z	16.	F, Y, S
7.	D, V, J	17.	Z, O, I
8.	N, P, A	18.	M, W, H
9.	K, L, B	19.	T, W, R
10.	R, C, Z	20.	Q, F, U

Зміст звіту

1. Мета роботи.
2. Скріпт-файл програми для пунктів 3, 4.
3. Прінт-скріни результатів дослідження мережі з різним ступенем спотворення символів.
4. Прінт-скріни розпізнавання символу, що не увійшов в навчальну вибірку.
5. Висновки.

Питання для самоперевірки

1. Які типи задач розв'язуються мережею Хопфілда.
2. Умови стійкості мережі Хопфілда.
3. Недоліки мережі Хопфілда.
4. Призначення енергетичної функції.
5. Критерії завершення розпізнавання.

Теоретичні відомості

Структура мережі Хопфілда наведена на рис. 4.1.

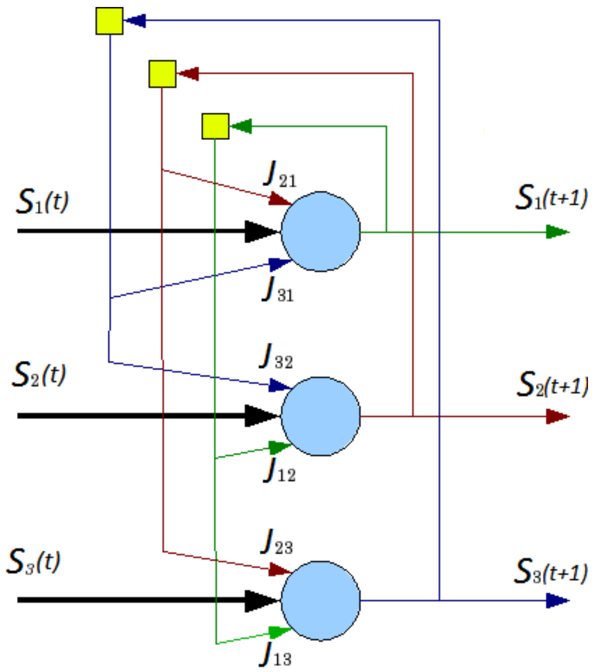

Рисунок 4.1 – Структурна схема мережі Хопфілда.

Слід зазначити, що в мережах зі зворотними зв'язками стан нейронів обчислюється доти, поки вони не виявляться сталими, не змінюваними згодом. Можна сказати, що мережа Хопфілда за певних умов збігається до сталого стану за скінченний час.

Енергетична функція

Використовується для оцінки стану мережі Хопфілда, що дозволяє оцінювати стани і обирати стани з найменшою енергією. Це може вказувати на те, що стан з найменшою енергією є найкращим (оптимальним) серед інших.

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij} y_i y_j - \sum_{j=1}^{N} x_j y_j + \sum_{j=1}^{N} \theta_j y_j$$

Для енергетичної функції вхідні значення повинні бути 0, 1.

Мережа Хопфілда є автоасоціативною мережею. Дискретна мережа Хопфілда має наступні характеристики: вона містить один шар елементів, кожен елемент зв'язується з усіма іншими елементами, але не пов'язаний з самим собою; за один крок роботи оновлюється лише один елемент мережі; елементи оновлюються у випадковому порядку; вихід елемента обмежений значеннями 0 або 1.

У мережі Хопфілда вхідні сигнали нейронів є одночасно і вихідними сигналами мережі: $x_i(k)= y_i(k-1)$, при цьому збудливий вектор особливо не виділяється. У класичній системі Хопфілда відсутній зв'язок нейрона з власним виходом, що відповідає $w_{ij}=0$, а вся матриця ваг є симетричною: $w_{ij}= w_{ji}$

$$\hat{W} = \hat{W}^T$$

Симетричність матриці ваг гарантує збіжність процесу навчання. Дана мережа не використовує ані навчання з учителем, ані навчання без учителя. Вагові коефіцієнти в ній розраховуються тільки перед початком функціонування мережі на основі інформації про оброблювані дані, і все навчання мережі зводиться саме до цього розрахунку. Мережа фактично запам'ятовує образи до того, як на її вхід надходять реальні дані, і не може змінювати свою поведінку, тому говорити про зворотний зв'язок із учителем не доводиться.

Процес навчання мережі формує зони тяжіння деяких точок рівноваги, відповідних навчальним даним. При використанні мережі в якості асоціативної пам'яті маємо справу з навчальним вектором, або з множиною цих векторів, які в результаті проведеного навчання визначають розташування конкретних точок тяжіння (атракторів). Кожен нейрон має порогову (Signum) функцію активації зі значеннями ± 1:

$$sign(a) = \begin{cases} 1, & a \geq 0 \\ -1, & a < 0 \end{cases}$$

Це означає, що вихідний сигнал *i*-го нейрона визначається функцією:

$$y_i = sign\left(\sum_{j=1}^{N} w_{ij} x_j + b_i \right)$$

де N позначає кількість нейронів. Часто постійна складова b_i , яка визначає поріг спрацьовування окремих нейронів, дорівнює 0. Тоді циклічне проходження сигналу в мережі Хопфілда можна надати співвідношенням:

$$y_i = sign\left(\sum_{j=1} w_{ij} y_j (k-1)) \right)$$

з початковою умовою $y_i\ (0)=x_i$.

У процесі функціонування мережі Хопфілда можна виділити два режими: навчання та класифікації. У режимі навчання на основі відомих навчальних вибірок обчислюються вагові коефіцієнти w_{ij}. У режимі класифікації при зафіксованих значеннях ваг і введенні конкретного початкового стану нейронів $y(0)=x$ виникає перехідний процес, що протікає відповідно з виразом (3) і закінчується в одному з локальних стійких станів, що задається біполярним вектором зі значеннями, $y=\pm 1$ для якого $y(k)=y(k-1)$.

Навчання не носить рекурентного характеру. Досить ввести значення (правило Хебба) ваг, визначивши їх через проекції вектору точки тяжіння еталонного образу:

$$w_{ij} = \frac{1}{N} x_i x_j$$

Згідно з цим правилом мережа дає правильний результат при вхідному векторі, що збігається з еталонним зразком, оскільки:

$$\sum_{j=1}^{N} w_{ij} x_j = \frac{1}{N} \sum_{j=1}^{N} x_i (x_j x_j) = x_i \sum_{j=1}^{N} 1 \frac{1}{N} = x_i$$

бо внаслідок біполярності значень елементів вектору x завжди $x_j^2 = 1$.

При введенні великої кількості навчальних вибірок $x^{(k)}$ для k=1, 2, ..., p ваги w_{ij} обчислюються згідно з узагальненим правилом Хебба відповідно до якого:

$$w_{ij} = \frac{1}{N}\sum_{k=1}^{l} x_i^{(k)} x_j^{(k)}$$

Завдяки такому режиму навчання ваги набирають значень, що є усередненням множини вибірок, пред'явлених до навчання. У разі декількох вибірок, актуальним стає питання про стабільність асоціативної пам'яті.

Після того як початкові умови задані у вигляді масиву T, що визначає ряд цільових вершин замкнутого гіперкуба, мережа для кожної вершини генерує вихід, який по зворотному зв'язку подається на вхід. Цей процес при створенні мережі повторюється багато разів, поки її вихід не перейде в стан рівноваги для кожної з цільових вершин. При подачі потім довільного вхідного вектора мережа Хопфілда переходить в результаті рекурсивного процесу до однієї з точок рівноваги, найбільш близької до вхідного сигналу.

Коли мережа Хопфілда спроектована, вона може бути перевірена з одним або більшим числом векторів входу. Досить імовірно, що вектори входу, близькі до цільових точок рівноваги, знайдуть свої цілі. Здатність мережі Хопфілда швидко обробляти набори векторів входу дозволяє перевірити мережу за відносно короткий час. Спочатку слід перевірити, що точки рівноваги цільових векторів дійсно належать вершинам гіперкуба, а потім можна визначити області тяжіння цих точок і виявити паразитні точки рівноваги (хибна пам'ять).

Створення нейронної мережі Хопфілда

Функция створення мережі Хопфілда має вигляд:

net=newhop(T),

де T – масив розміру R * Q, який об'єднує Q цільових векторів зі значеннями –1, +1 або 0, 1 для елементів; R – число елементів вектору входу.

Створимо нейронну мережу Хопфілда з чотирма нейронами і визначимо 4 точки рівноваги у вигляді наступного масиву цільових векторів:

T = [1 –1; –1 1; 1 1; –1 –1];

Покажемо точки рівноваги на графіку

```
T1 = T';
plot(T1(1,:), T1(2,:),'*r')
axis([-1.1 1.1 -1.1 1.1])
title('Точки рівноваги мережі Хопфілда')
xlabel('a(1)'), ylabel('a(2)')
```

На рис. 4.1 показані ці 4 точки рівноваги на площині станів мережі Хопфілда

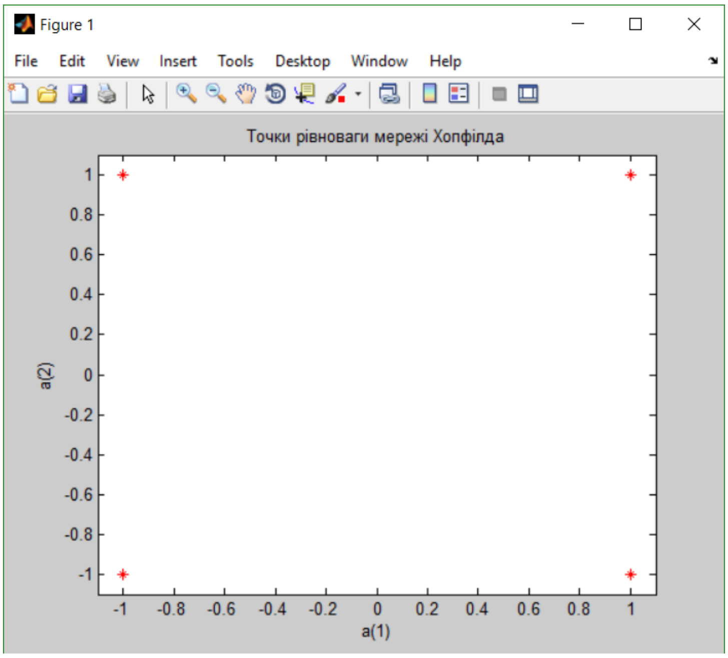

Рисунок 7.1 – Точки рівноваги на площині станів мережі Хопфілда

Розрахуємо ваги і зсуви мережі Хопфілда, скористаємось М-функцію **newhop.**

```
net = newhop(T1);
W = net.LW{1,1}
b = net.b{1,1}
W =
1.1618          0
     0     1.1618
b =
1.0e-016 *
      0
–0.1797
```

Перевіримо, чи належать вершини квадрата мережі Хопфілда:

```
Ai = T1;
[Y,Pf,Af] = sim(net, 4, [], Ai)
Y =
   1   -1    1   -1
  -1    1    1   -1
Pf =
   {}
Af =
  [2x4 double]
```

Як і слід було очікувати, виходи мережі дорівнюють цільовим векторам. Тепер перевіримо поведінку мережі при випадкових початкових умовах.

```
plot(T1(1,:), T1(2,:),'*r', 0,0,'rh'), hold on, axis([–1.1  1.1  –1.1  1.1])
xlabel('a(1)'), ylabel('a(2)')
new = newhop(T);
[Y,Pf,Af] = sim(net,4,[],T1);
```

```
for i = 1:20
    a = {rands(2,1)};
    [Y, Pf, Af] = sim(net, {1,20}, {}, a);
    record = [cell2mat(a) cell2mat(Y)];
    start = cell2mat(a);
    plot(start(1,1), start(2,1), 'kx', record(1,:), record(2,:))
end
```

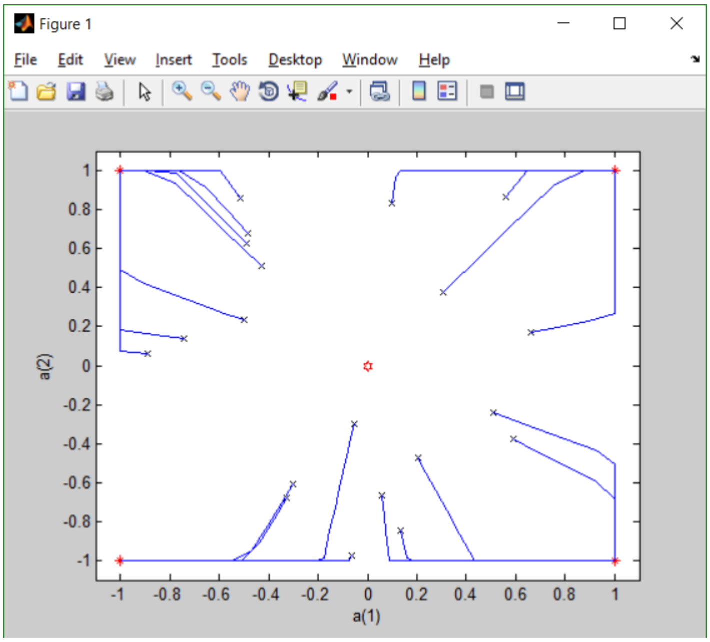

Рисунок 4.3 – Поведінка мережі при випадкових початкових умовах

Використання мережі Хопфіда в якості асоціативної пам'яті.

Завдання: створити і навчити мережу Хопфілда розпізнавати спотворені символи латинського алфавіту, дослідити вплив величини спотворень на точність розпізнавання. Скористаємось файлом Alphabet.csv, який містить матрицю 35x26 кодів великих літер латинського алфавіту. Кожна літера – це колонка, отримана з зображення літери розміром 5x7 елементів.

Приклад для символу A та J.

```
alphabet=csvread('Alphabet.csv');
L1 = alphabet(:, 1); %Завантаження літери A
L2 = alphabet(:, 10); %Завантаження літери J
%Виведення літери A J та у виді зображення 5x7 елементів
letterA = reshape(L1, 5, 7)';
letterA
subplot(1,4,1);  plotchar(L1);
letterA =
     0 0 1 0 0
     0 1 0 1 0
     0 1 0 1 0
     1 0 0 0 1
     1 1 1 1 1
     1 0 0 0 1
```

```
    1 0 0 0 1
        %Створення мережі Хопфілда для розпізнавання літер A та J
T=[L1,L2];
net = newhop(T);
%Спотворення літер
noisyL1 = L1 + randn(35,1) *0.4;
noisyL2 = L2 + randn(35,1) *0.4;
%Виведення спотворених літер;
subplot(2,4,2); plotchar(noisyL1);
subplot(2,4,6); plotchar(noisyL2);
         %Розпізнавання спотворених літер і виведення результату
[Y,Pf,Af] = sim(net, 1,{}, noisyL1);
subplot(2,4,3); plotchar(Y);
[Y1,Pf,Af] = net(1,{}, noisyL2);
subplot(2,4,7); plotchar(Y1);
        %Друге розпізнавання спотворених літер і виведення результату
        %Спотворюємо літеру A з коефіцієнтом 0.8
noisyL1 = L1 + randn(35,1) *0.8;
[Y2,Pf,Af] = net(1,{}, noisyL1);
subplot(2,4,4); plotchar(Y2);
        %В якості спотвореної літери подаємо літеру V
noisyL3 = alphabet(:, 22);
[Y3,Pf,Af] = net(1,{}, noisyL3);
subplot(2,4,8); plotchar(Y3);
%Результати подано на рис. 7.3
```

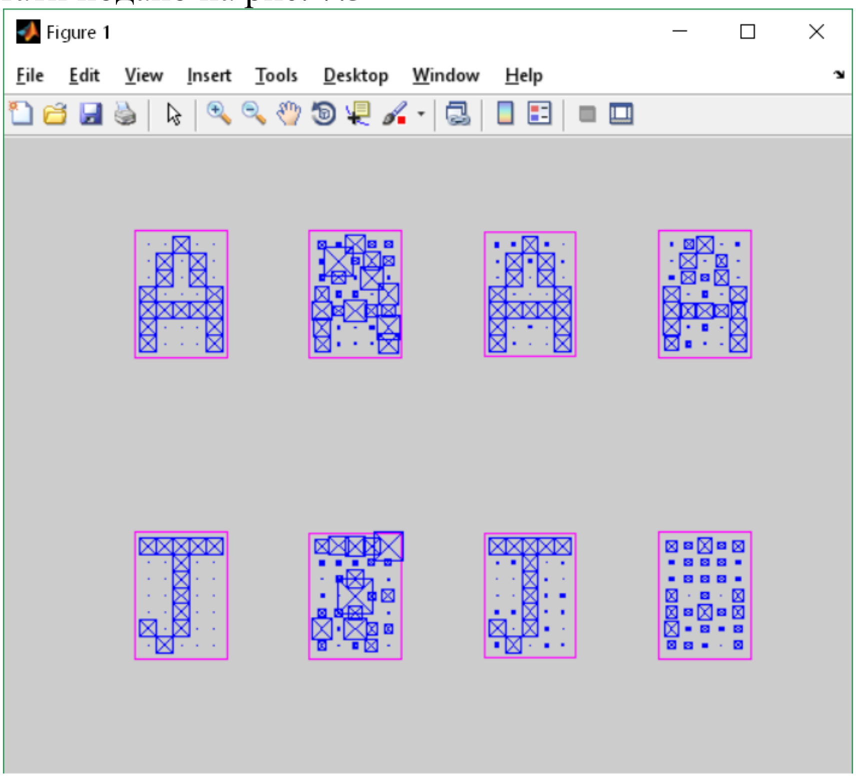

Рисунок 7.3 – Результат моделювання

Додаток А
Опис функцій MATLAB

Функція	Опис
B = reshape(A, m, n) або B = reshape(A,[m n])	Повертає матрицю B розміром m на n, елементи якої беруться по колонках з A. Якщо A не має m*, виникає помилка.
plotchar	Plot a 35 element vector as a 5x7 grid.
[Y,Pf,Af]=sim(net, {m n},{}, L) або [Y,Pf,Af]=net({m n}, {}, L)	Повертає матрицю Y розміром L x n з виходу мережі, n – кількість ітерацій (сходження к стійкому стану) мережі (подача виходу на вхід), L – вхідний образ (колонка). '. Вхідний образ L подається на вхід мережі, здійснюється 5 циклів ітерації, змінна B набуває значення другої ітерації. [Y, Pf, Af]=sim(net, {1 5},{}, L); B=Y{2};
T=csvread(FailMName)	Читання файлу даних формату .cvs (дані розділені комами) в змінну T. файл повинен бути в CurrentFolder MATLABб або вказується повний шлях до файлу. **Приклад.** alphabet=csvread('Alphabet.csv'); alphabet=csvread('C:\User\Documents\Alphabet.csv')

Література

1. Howard Demuth Mark Beale Neural Network Toolbox For Use with MATLAB User's Guide Version 4, The MathWorks, Inc., 2004, 846p.
2. Ямпольський Л.С. Нейротехнології та нейрокомп'ютерні системи: підручник / Л.С. Ямпольський, О.І. Лісовиченко, В.В. Олійник. – К.: «Дорадо-Друк», 2016. – 576 с.: іл.