Київський національний університет   
імені Т.Шевченка

**Звіт**

до лабораторної роботи 2  
з предмету Нейронні мережі та нейрообчислення «Мережі Хопфілда»

***Студента четвертого курсу   
Групи ТК-42   
Факультету комп’ютерних наук   
та кібернетики   
Мальованого Дмитра***

***Київ***

*2023*

**Теоретичні відомості:**

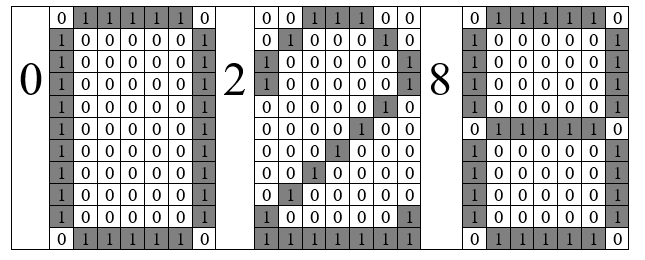
Мережа Хопфілда - це модель нейромережі зворотного зв'язку, яка використовується для асоціативної пам'яті та розв'язання оптимізаційних задач. Вона була запропонована Джоном Хопфілдом в 1982 році.

У мережі Хопфілда кожен нейрон пов'язаний з кожним іншим нейроном в мережі. Кожен з нейронів може приймати значення 0 або 1. Зв'язки між нейронами задаються матрицею ваг, яка зберігається в мережі.

Мережа Хопфілда може бути використана для зберігання та відновлення паттернів. Зберігання відбувається за допомогою вивчення мережі на основі навчальних зразків. Після навчання, якщо мережі буде подано злегка змінений зразок, вона зможе розпізнати його та повернути збережений зразок, який найбільш схожий на вхідний зразок.

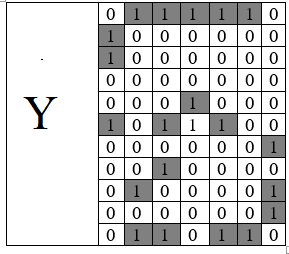
**Постановка задачі:**

1. Для роботи з нейронною мережею Хопфілда взяти в якості еталонів цифри від 0 до 9. Розмір цифр має бути 7\*11 пікселів або 9\*11 пікселів. Наприклад.



1. Навчити нейронну мереже на вибраних еталонах.
2. Створити зашумлений образ і очистити (розпізнати) його мережею Хопфілда.

Наприклад.



**Алгоритм:**

**Алгоритм навчання:**

Перший етап – навчання. Нехай еталонний зразок N-мірний вектор

**x=[x1,x2,...,xN],**

де xi рівні -1 або 1.

Результат навчання мережі Хопфілда - вагова матриця.

1. wii=0
2. Для всіх образів корегування ваг:

wij=wij + xixj,

1. wii=0

і нормуємо wij:=wij/N.

**Розпізнавання.**

Нехай шукаємо образ Y=[y1,y2,...,yN].

1. Цикл по *j* от *1* до *N*:
2. Покладемо *d = 0*
3. Вкладений цикл по *i* від *1* до *N*:
4. *d = d + wijyi*
5. Кінець вкладеного циклу
6. Якщо *d > 0*, то *zj = 1*, інакше *zj = -1*.
7. Кінець зовнішнього циклу
8. Маємо вектор: *Z=[z1,z2,...,zN].*

Якщо вектор *Z* є в множині еталонів, то алгоритм знайшов образ *Z*, що відповідає *Y*.

1. Покладемо Y = Z
2. Переходимо до кроку 1.

Якщо алгоритм не може знайти образ, то значить мережа Хопфілда не згадала такий образ. Це може бути у випадку, коли значення вектора *Z* не міняється в процесі навчання.

**Опис роботи:**

В якості данних для тренування – було взято файли з цифрами розміром 11x9, від 0 до 9. Данні доступні за посиланням: <https://github.com/DiMalovanyy/University_Term9/tree/main/NeurNet/Lab2/digits>

Спочатку зчитуємо файли та перетворюємо їх у масив, що має значення -1 або 1: якщо цифра 0, то значення в масиві -1, інакше 1. Ініціалізуємо матрицю вагів нулями. Зчитуємо зашумлені зразки для розпізнавання.

Потім тренуємо мережу за допомогою набору масивів для тренування.

Потім вибираємо випадкове число та зашумлюємо його. Перевіряємо що число зчитуєтеся навченою мережою.

Весь код наведений за посиланням: <https://github.com/DiMalovanyy/University_Term9/blob/main/NeurNet/Lab2/lab2.py>

**Результати:**

Розмір тренувального масиву: 4 [0,1,5,7]

|  |  |
| --- | --- |
| Вхід | Вихід |
| **000010000**  **000110000**  **001010000**  **010010000**  **000010000**  **000001000**  **000001000**  **000001000**  **000010000**  **000010000**  **111111111** | **000010000**  **000110000**  **001010000**  **010010000**  **000010000**  **000010000**  **000010000**  **000010000**  **000010000**  **000010000**  **111111111** |

Якщо на вхід подати зашумлену п’ятірку з файлу src0.txt, то за даного тренувального набору програма не знайде її еталон, хоча якщо змінити тренувальні набори і форму цифр у еталонах налаштувати краще, то розпізнавання можливе.

|  |  |
| --- | --- |
| Вхід | Вихід |
| **111111111  100000000  111000000  111100000  100000000  111111011  000000001  000000001  000000101  000001111  111111111** | **111111111**  **100000001**  **100000001**  **100000001**  **100000001**  **100000001**  **000000001**  **000000001**  **000000001**  **000000001**  **011111111** |

Якщо зменшити тренувальний набір до 2 еталонів, то і зашумлена трійка і зашумлена одиниця буде розпізнаватися

|  |  |
| --- | --- |
| Вхід | Вихід |
| **111111111**  **000000001**  **000000001**  **000000001**  **000000001**  **111111111**  **000000001**  **000000000**  **000000010**  **000000110**  **111111111** | **111111111  000000001  000000001  000000001  000000001  111111111  000000001  000000001  000000001  000000001  111111111** |

Тренувальний набір складається з 5 та 3. На вхід подаємо зашумлену 3 та 5, система добре розпізнає їх.

|  |  |
| --- | --- |
| Вхід | Вихід |
| **111111111**  **100000000**  **110000000**  **100000000**  **110000000**  **111111011**  **000000001**  **000000111**  **000000001**  **000001111**  **111111111** | **111111111  100000000  100000000  100000000  100000000  111111111  000000001  000000001  000000001  000000001  111111111** |

Якщо за такого тренувального масиву подати на вхід цілковитий шум, то система не буде розпізнавати взагалі.

|  |  |
| --- | --- |
| Вхід | Вихід |
| **010101010**  **110000110**  **010101010**  **010101010**  **010101010**  **111000101**  **101010010**  **101010010**  **101001010**  **101001010**  **101001010** | **000000000**  **111111110**  **111111110**  **111111110**  **111111110**  **000000000**  **111111110**  **111111110**  **111111110**  **111111110**  **000000000** |

Якщо збільшувати розмір навчального датасету до 10 цифр, то система почне гірше працювати за наявних зразків. Якщо зразки краще настроїти то система буде краще їх розпізнавати.