

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

Кафедра інтелектуальних та інформаційних систем

Лабораторна робота № 7
з дисципліни
“Нейромережні технології та їх застосування”

Виконав студент
групи КН-31
Пашковський Павло Володимирович

Київ-2021

Контрольні питання

1. Архітектура нейронної мережі.

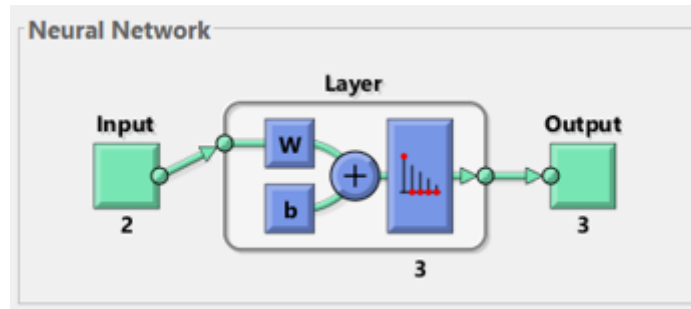


Рисунок 5 – Архітектура мережі

2. Правило знаходження кількості нейронів в мережі для розпізнавання заданого числа класів.

$$n^{(2)} = \frac{\sum_p P * k_p}{2} \quad (1)$$

P -число вхідних факторів, k_p -глибина

3. Алгоритм навчання.

Алгоритм Хебба:

1. На стадії ініціалізації всім ваговим коефіцієнтам присвоюється невелике випадкове значення.
2. На входи мережі подається вхідний образ і сигнал збудження поширюється по всіх верствах мережі.
3. На підставі повчаннях вихідних значень Y_i , Y_j проводиться зміна вагових коефіцієнтів.
4. Перехід на крок 2, доки виходи мережі не стабілізуються з заданою точністю. При цьому на другому етапі циклу поперемінно пред'являються все образи навчальної множини.

Повний алгоритм навчання має приблизно таку ж саму структуру, як і алгоритм Хебба, а на кроці 3 з усього шару вибирається 1 нейрон, значення синапсів якого максимально підходять на вхідний образ і підстроювання ваг цього нейрона виконується за формулою 1.

$$W(t) \leftarrow W(t-1) + \alpha [y^{(n-1)} - W(t-1)] \quad (1)$$

Наведена процедура розбиває безліч вхідних образів на кластери властиві вхідним даними. В правильно навченої мережі для поточного вхідного образу X активізується тільки один вихідний нейрон-переможець, відповідний кластеру, який потрапляє в поточний вхідний образ X .

4. Результат, котрий повертає функція `sim`.

Функція `sim` повертає закодовану матрицю 1 та 0, де кожен стовпчик є класом.

5. Параметри функції `newc`.

`newc(PR,S,KLR,CLR)`

PR – $R \times 2$ матриця мінімальних і максимальних значень для вхідних значень R

S – кількість нейронів

KLR – швидкість навчання Коханена, за замовчуванням – 0.01

CLR – швидкість навчання, за замовчуванням – 0.001

Індивідуальне завдання:

Варіант 7.

Кількість класів для класифікації	Координати точок перевірконої множини та номер класу, якому належить дана точка
3	$[-1;0]$ -1; $[-1;1]$ -2; $[-1;-1]$ -3

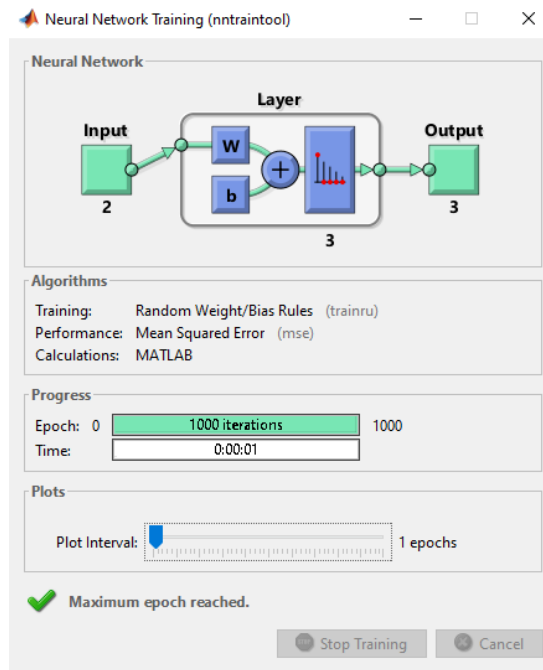


Рисунок 2.1 – Результати роботи програми

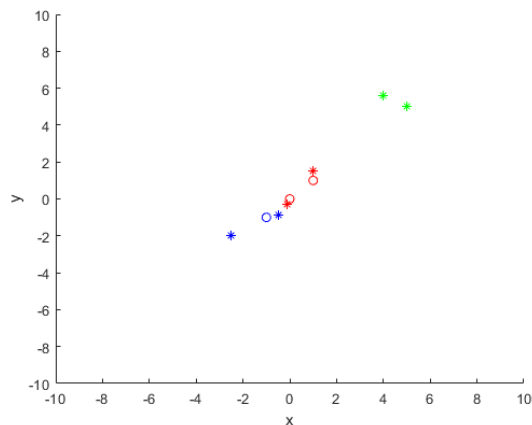


Рисунок 2.2 – Графік точок (* – тренувальні, о – тестові)

```

Train Coordinate -1.000000e-01 and -3.000000e-01 refers to 1
Train Coordinate 1 and 1.500000e+00 refers to 1
Train Coordinate -5.000000e-01 and -9.000000e-01 refers to 2
Train Coordinate -2.500000e+00 and -2 refers to 2
Train Coordinate 4 and 5.600000e+00 refers to 3
Train Coordinate 5 and 5 refers to 3
Test Coordinate 0 and 0 refers to 1
Test Coordinate 1 and 1 refers to 1
Test Coordinate -1 and -1 refers to 2
...

```

Рисунок 2.3 – Вивід програми в консолі

Висновок:

У даній роботі було створено програму для вивчення моделі шару Кохонена та алгоритму навчання без учителя. Навчання відбувалося за допомогою програмного забезпечення MATLAB.

Результати програми можна практично застосовувати для розділення класів.

Отримані навички щодо навчання мережі можна використовувати у абсолютно різних сферах, починаючи від розпізнавання образів, закінчуючи обробкою даних.

Код програми:

```
P = [-0.1 1 -0.5 -2.5 4 5; -0.3 1.5 -0.9 -2 5.6 5];
net = newc([0 1; 0 1],3);
net = train(net,P);
trained_pred = sim(net,P);
trained_vector = vec2ind(trained_pred);
hold on
for index = 1:6
    fprintf('Train      Coordinate      %d      and      %d      refers      to      %d\n',P(1,index),P(2,index),trained_vector(index));
    if trained_vector(index) == 1
        plot(P(1,index),P(2,index),'r*')
    end
    if trained_vector(index) == 2
        plot(P(1,index),P(2,index),'b*');
    end
    if trained_vector(index) == 3
        plot(P(1,index),P(2,index),'g*');
    end
end

Test = [0 1 -1; 0 1 -1];
predicted=sim(net,Test);
pred_vector = vec2ind(predicted);
for index = 1:3
    fprintf('Test      Coordinate      %d      and      %d      refers      to      %d\n',Test(1,index),Test(2,index),pred_vector(index));
    if pred_vector(index) == 1
        plot(Test(1,index),Test(2,index),'ro')
    end
    if pred_vector(index) == 2
        plot(Test(1,index),Test(2,index),'bo');
    end
    if pred_vector(index) == 3
```

```
        plot(Test(1,index),Test(2,index),'go');
    end

end

hold off

axis([-10 10 -10 10])
xlabel('x');
ylabel('y');
```