

ПРИДНІПРОВСЬКА ДЕРЖАВНА АКАДЕМІЯ БУДІВНИЦТВА ТА АРХІТЕКТУРИ

Кафедра комп'ютерних наук, інформаційних технологій
та прикладної математики

Лабораторна робота № 1
з дисципліни «Нейронні мережі»

Виконав(ла) студент(ка) III курсу

група КНз-22 спеціальність 122
«Комп'ютерні науки»

_____ Пойманова Дар'я _____
(прізвище та ініціали)

Перевірів

(прізвище та ініціали)

Національна шкала _____

Кількість балів: _____ Оцінка: ECTS _____

м. Дніпро – 2025

Зміст

ВСТУП	3
ЛАБОРАТОРНА РОБОТА №1 (ЧАСТИНА ПЕРША)	4
ЛАБОРАТОРНА РОБОТА №1 (ЧАСТИНА ДРУГА)	16
ВИСНОВОК	24

Вступ

Штучні нейронні мережі (ШНМ) є важливим інструментом для вирішення задач класифікації, розпізнавання образів та прогнозування. Однією з найпростіших, але базових моделей є одношаровий персептрон, який дозволяє класифікувати вхідні вектори на основі лінійного розділення простору ознак. У лабораторній роботі розглядається побудова персептрона для задач класифікації дво- та тривимірних образів, а також багатокласових задач.

Особлива увага приділяється нормуванню вхідних даних, що є важливим кроком перед навчанням моделі. Якщо значення вхідних ознак суттєво відрізняються за масштабом, це може призвести до повільної або неефективної роботи навчального алгоритму. Масштабування дозволяє рівномірно впливати кожному вектору на зміну ваг, що пришвидшує збіжність моделі. У MATLAB це досягається шляхом використання нормованої функції навчання 'learnnp'.

У межах цієї лабораторної роботи було сформовано архітектуру мережі, підготовлено навчальні шаблони згідно з варіантом, проведено навчання з використанням `train` або `adapt`, а також виконано моделювання з візуалізацією роздільних гіперплощин. Отримані результати дозволяють краще зрозуміти принципи функціонування персептрона, його обмеження та ефективність при використанні нормованого навчання.

Лабораторна робота №1 (частина перша)

Варіант №4

Тема: Класифікація за допомогою нейронних мереж у MATLAB

Мета роботи: Навчитися створювати, налаштовувати та навчати нейронну мережу для класифікації даних. Ознайомитися з архітектурою одношарових та багатшарових персептронів для розв'язання задач класифікації на два або більше класів за допомогою MATLAB.

Завдання:

1. Вибрати архітектуру та сформувати штучну нейронну мережу для вирішення задачі класифікації двоелементних образів на два класи.

Набори навчальних шаблонів представлено в таблицях:

Варіант 4

Образ	Клас
[1.5; -2.0]	1
[1.0; -0.5]	0
[-1.5; -1.0]	0
[-0.5; -1.5]	1

2. Побудувати графік, який зображує простір входів, вхідні вектори та лінію, яка відображує, як навчена нейронна мережа розділяє простір входів на дві області.

Вхідні вектори (P) та відповідні класи (T):

$P = [1.5 \ 1.0 \ -1.5 \ -0.5; -2.0 \ -0.5 \ -1.0 \ -1.5];$

$T = [1 \ 0 \ 0 \ 1];$

Код для навчання та побудови гіперплощини:

$P = [1.5 \ 1.0 \ -1.5 \ -0.5; -2.0 \ -0.5 \ -1.0 \ -1.5];$

$T = [1 \ 0 \ 0 \ 1];$

`net = perceptron;`

% Отримуємо додатковий вихід tr

`[net, tr] = train(net, P, T);`

$Y = \text{net}(P);$

```
% Network Diagram
```

```
view(net);
```

```
% Графік performance з інформацією tr
```

```
plotperform(tr);
```

```
% Побудова графіка простору входів для 2D
```

```
figure;
```

```
plotpv(P, T);
```

```
hold on;
```

```
plotpc(net.IW{1}, net.b{1});
```

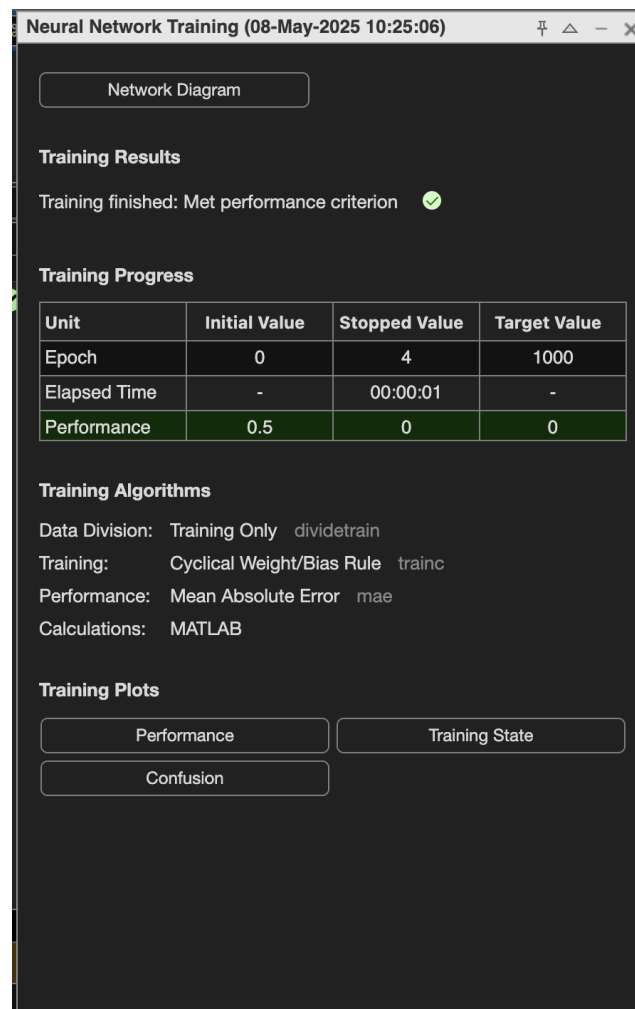


Рисунок 1.

Network diagram, performance:

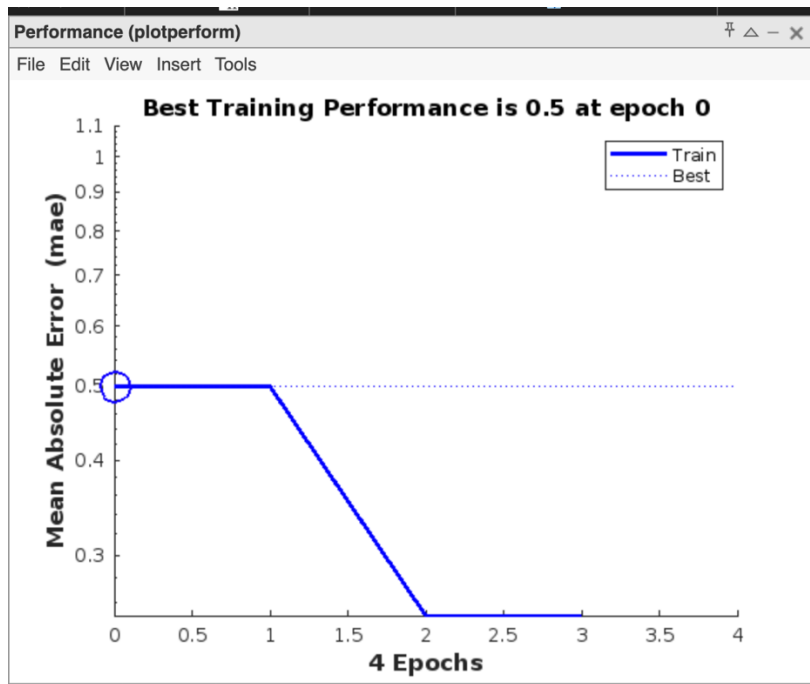


Рисунок 2. Performance

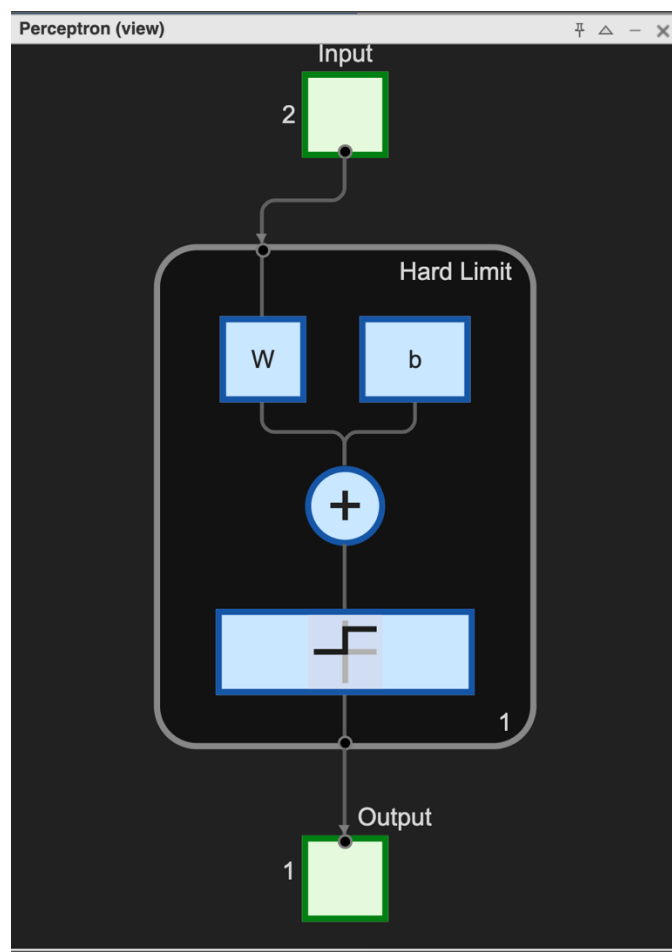


Рисунок 3. Network Diagram

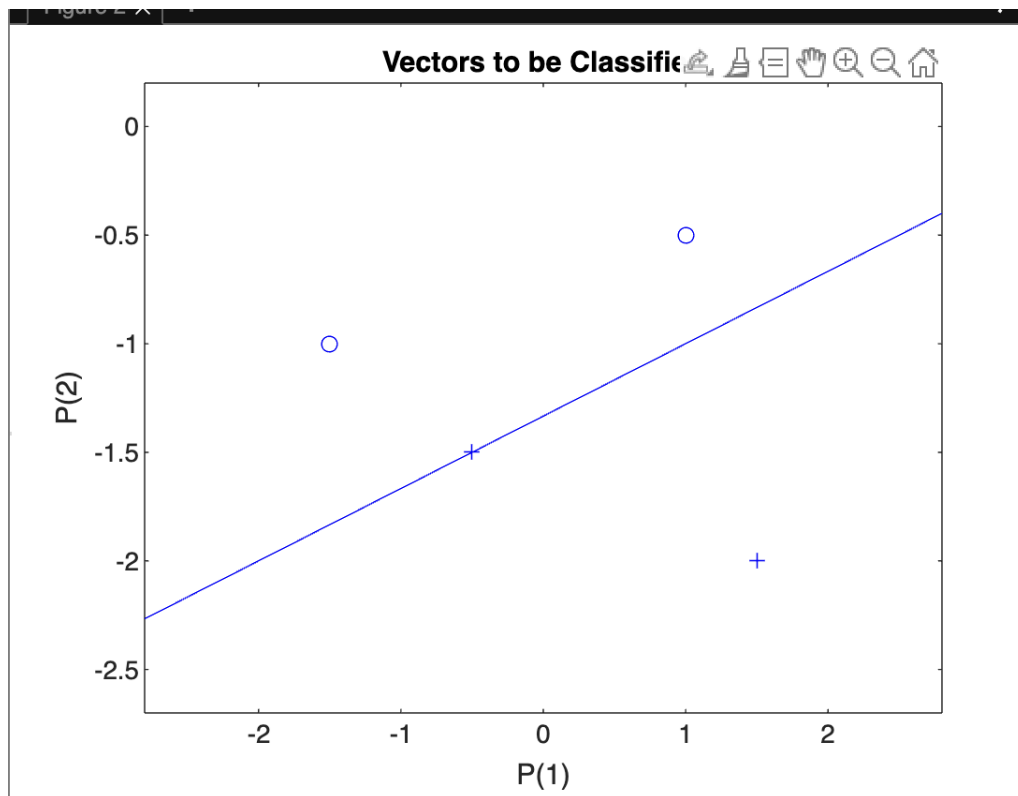


Рисунок 5.

Після навчання побудована лінія відображає розділення простору входів на 2 класи.

3. Вибрати архітектуру та сформувати штучну нейронну мережу для вирішення задачі класифікації трьохелементних образів на два класи.

Набори навчальних шаблонів представлено в таблицях:

Варіант 4

Образ	Клас
$[-2.0; -2.0; -2.0]$	1
$[2.0; -2.0; -2.0]$	1
$[-2.0; 2.0; -2.0]$	1
$[2.0; 2.0; -2.0]$	1
$[-2.0; -2.0; 2.0]$	0
$[2.0; -2.0; 2.0]$	0
$[-2.0; 2.0; 2.0]$	0
$[2.0; 2.0; 2.0]$	0

4. Побудувати графік, який зображує простір входів, вхідні вектори та площину, яка відображує, як навчена нейронна мережа розділяє простір входів на дві області.

P — матриця 3×8 , кожен вектор має 3 координати, з таблиці (друге зображення):

Код навчання для трьохелементних векторів:

```
% Вхідні дані
```

```
P = [-2.0 2.0 -2.0 2.0 -2.0 2.0 -2.0 2.0;  
     -2.0 -2.0 2.0 2.0 -2.0 -2.0 2.0 2.0;  
     -2.0 -2.0 -2.0 -2.0 2.0 2.0 2.0 2.0];  
T = [1 1 1 1 0 0 0 0];
```

```
% Створення та навчання мережі
```

```
net = patternnet(10);  
net.trainParam.epochs = 200;  
[net, tr] = train(net, P, T);
```

```
% Network Diagram та графік Performance
```

```
view(net);  
plotperform(tr);
```

```
% Перевірка результатів класифікації
```

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

```
% Візуалізація точок у 3D-просторі
```

```
figure;  
scatter3(P(1,:), P(2,:), P(3,:), 100, T, 'filled');  
xlabel('X'); ylabel('Y'); zlabel('Z');  
title('Точки вхідних даних у 3D-просторі');
```

```
% Коректна побудова межі прийняття рішень (через contour3)
```

```
[x, y, z] = meshgrid(-3:0.5:3, -3:0.5:3, -3:0.5:3);  
grid_points = [x(:)'; y(:)'; z(:)'];  
Z_pred = net(grid_points);  
Z_pred = reshape(Z_pred, size(x));
```

```
figure;  
isosurface(x, y, z, Z_pred, 0.5);  
hold on;  
scatter3(P(1,:), P(2,:), P(3,:), 100, T, 'filled');  
xlabel('X'); ylabel('Y'); zlabel('Z');  
title('Ізоповерхня межі розподілу класів');
```

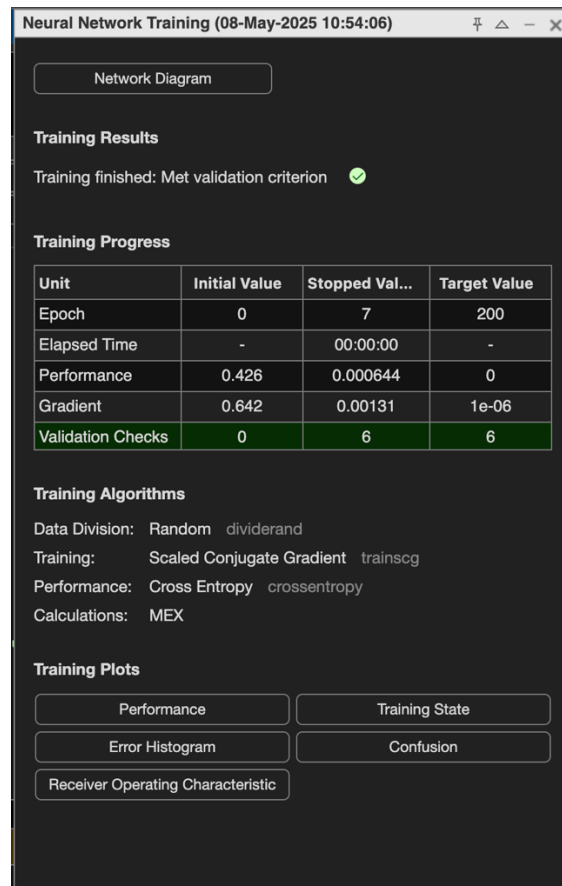



Рисунок 6.

У просторі трьох входів побудована гіперплощина дозволяє відобразити класифікацію точок. Для графіка в 3D можна використати plot3 або побудувати вручну площину за нормаллю.

Network diagram, performance:

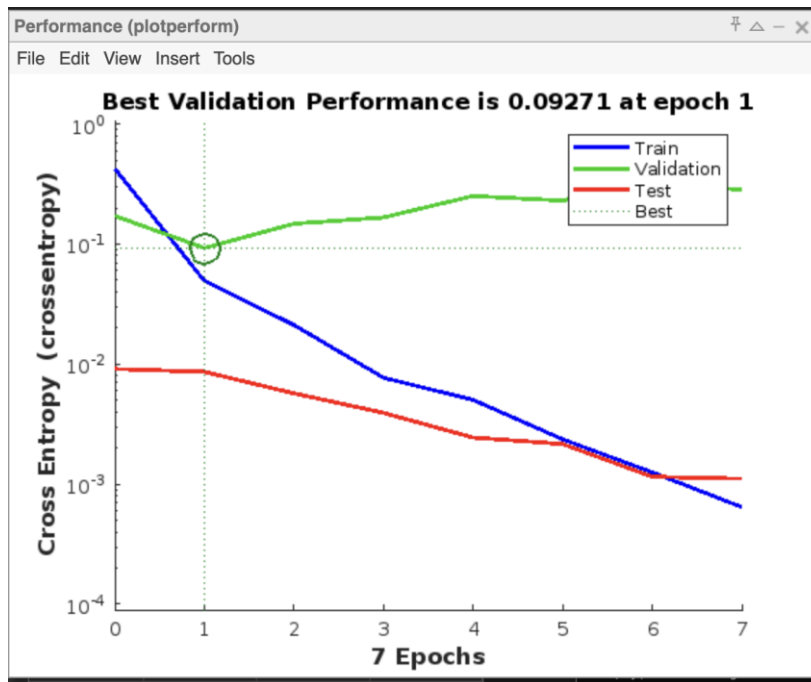


Рисунок 7. Performance

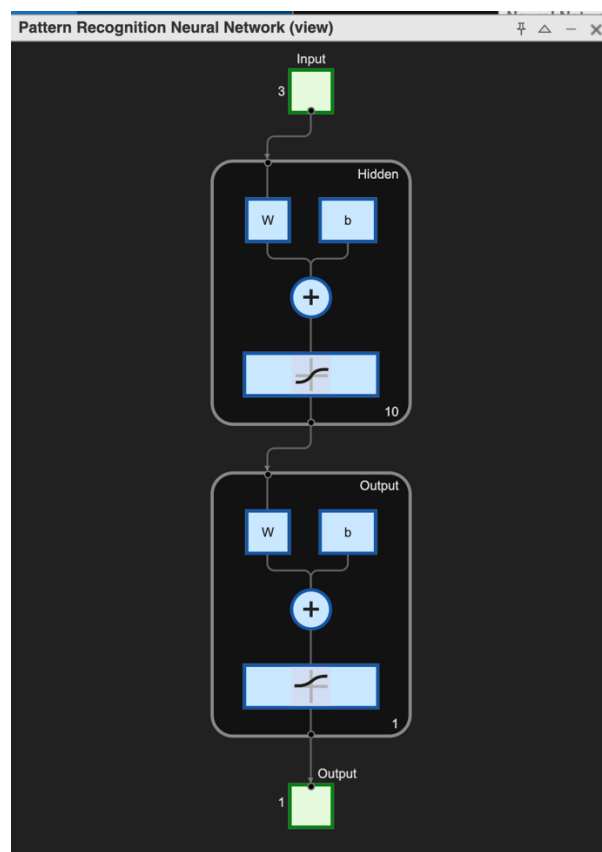


Рисунок 8. Network Diagram

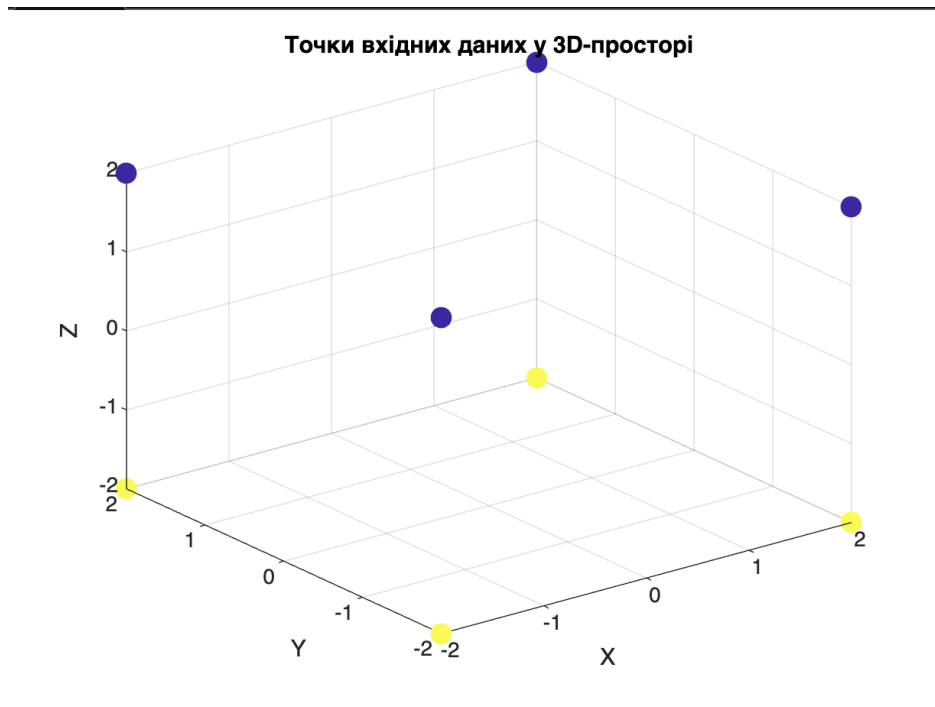


Рисунок 9. Точки вихідних даних.

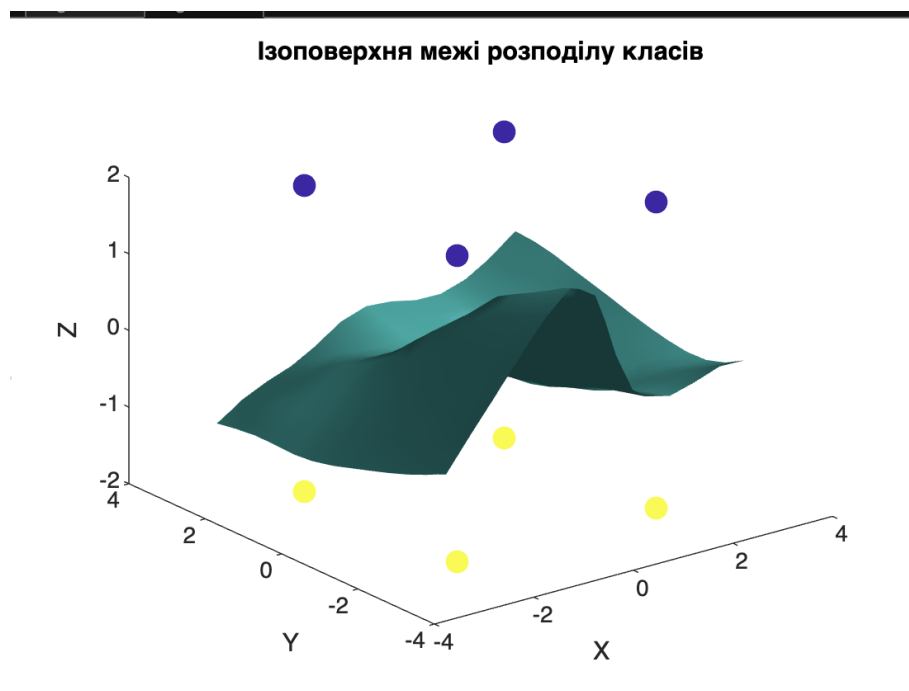


Рисунок 10. Ізоповерхня межі розподілу класів.

5. Вибрати архітектуру та сформувати штучну нейронну мережу для вирішення задачі класифікації двохелементних образів на чотири класи.

Набори навчальних шаблонів представлено в таблицях:

Варіант 4

Образ	Клас
[0.5; 3.0]	[1; 0]
[-0.4; 2.5]	[1; 0]
[-0.8; -0.3]	[1; 1]
[-1.0; 0.4]	[1; 1]
[-0.5; 0.1]	[1; 1]
[1.3; -0.4]	[0; 0]
[1.0; 0.2]	[0; 0]
[1.5; -0.3]	[0; 0]
[0.4; -1.5]	[0; 1]
[-0.2; -1.8]	[0; 1]

6. Побудувати графік, який зображує простір входів, вхідні вектори та лінії, яка відображує, як навчена нейронна мережа розділяє простір входів на чотири області.

P = матриця 2×10 ; T = 2×10 у форматі one-hot:

Використовується багатошаровий персептрон (patternnet) з 5 нейронами у прихованому шарі. Нижче наведено повний код для навчання мережі та побудови графіка класифікації:

% Вхідні дані

```
P = [0.5 -0.4 -0.8 -1.0 -0.5 1.3 1.0 1.5 0.4 -0.2;
      3.0 2.5 -0.3 0.4 0.1 -0.4 0.2 -0.3 -1.5 -1.8];
T = [1 1 1 1 1 0 0 0 0 0;
      0 0 1 1 1 0 0 0 1 1];
```

% Створення мережі

```
net = patternnet(10);
net.trainParam.epochs = 200;
[net, tr] = train(net, P, T);
```

% Network Diagram і Performance

```
view(net);
plotperform(tr);
```

% Класифікація

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

% Графік результатів

```
figure;
classes = vec2ind(T); % Отримуємо номер класу
h = gscatter(P(1,:), P(2,:), classes);
```

```
hold on;
```

```
% Візуалізація меж класифікації
```

```
[x, y] = meshgrid(-2:0.1:2, -2:0.1:4);
```

```
grid_points = [x(:)'; y(:)'];
```

```
z = net(grid_points);
```

```
[~, z_idx] = max(z);
```

```
z_idx = reshape(z_idx, size(x));
```

```
contour(x, y, z_idx, 'LineWidth', 1.5, 'LineColor', 'k');
```

```
% Коректна побудова легенди
```

```
legend_entries = unique(classes);
```

```
legend_labels = arrayfun(@(x) sprintf('Клас %d', x), legend_entries,  
'UniformOutput', false);
```

```
legend_labels{end+1} = 'Роздільна межа';
```

```
legend([h; plot(NaN,NaN,'k','LineWidth',1.5)], legend_labels, 'Location', 'best');
```

```
title('Класифікація на 4 класи (виправлена)');
```

```
xlabel('x'); ylabel('y');
```

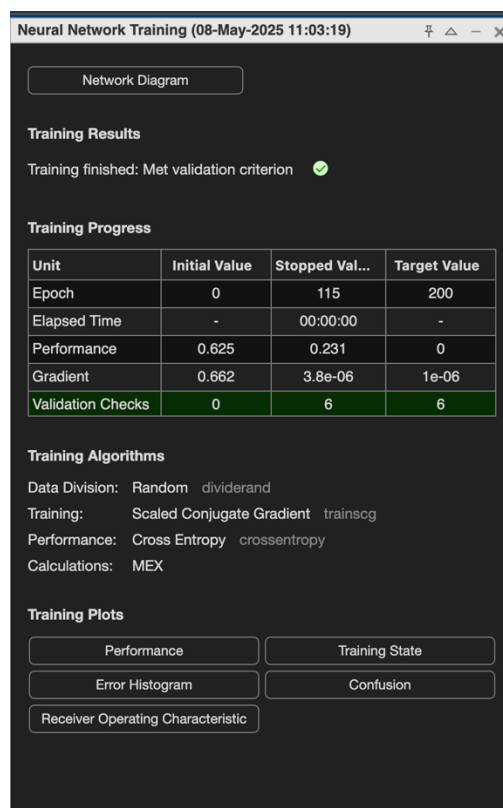


Рисунок 11.

Network diagram, performance:

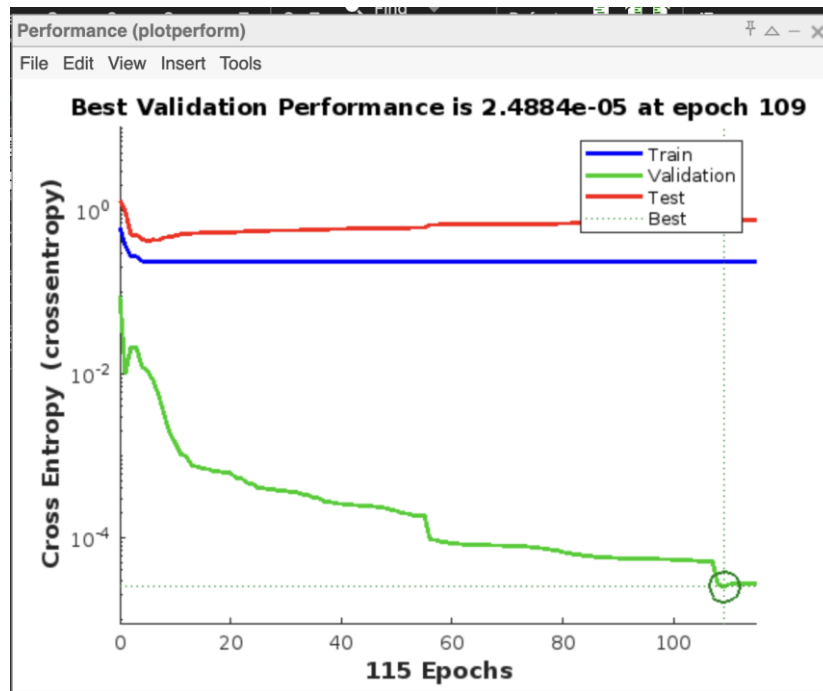


Рисунок 12. Performance

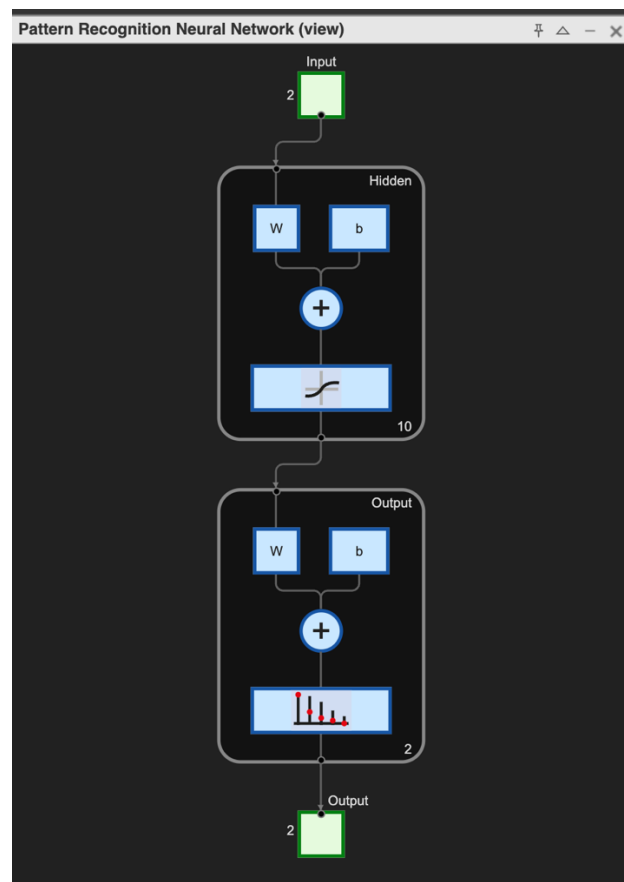


Рисунок 13. Network Diagram

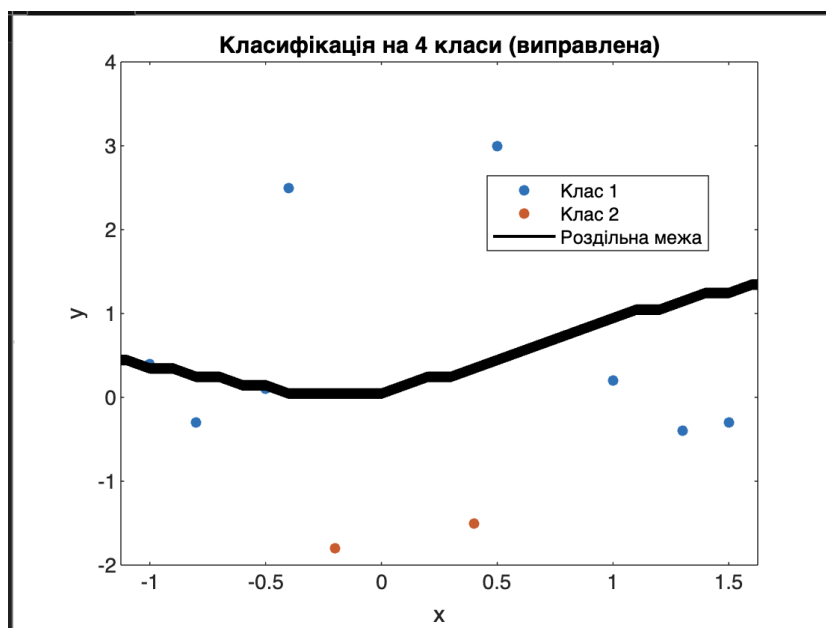


Рисунок 14. Класифікація на 4 класи

Лабораторна робота №1 (частина друга)

1. Вибрати архітектуру та сформувати штучну нейронну мережу для вирішення задачі класифікації:

Варіант 3: літературні твори за двома ознаками (кількість головних героїв та кількість сторінок) на два класи:

1) повісті – кількість головних героїв не більше 5 та кількість сторінок не перевищує 200;

2) романи – кількість головних героїв більше 5 та кількість сторінок понад 200.

Для вирішення задачі необхідно:

- сформувати навчальну вибірку, яка містить достатню кількість шаблонів, кожен з яких включає вхідний вектор та відповідний вихідний вектор; сформувати персептрон відповідної розмірності, вказавши в якості функції навчання **'learnnpn'**;
- реалізувати процес навчання персептрона за допомогою функції **adapt** або **train**;
- шляхом моделювання перевірити результат роботи навченої мережі.

2. Побудувати графік, який зображує простір входів та лінії, що відображають, як налаштована нейронна мережа ділить простір входів на підпростори, які відповідають різним класам.

3. Вибрати архітектуру та сформувати штучну нейронну мережу для вирішення задачі класифікації двоелементних вхідних векторів відповідно до рис. 2.1. Масштаб графіка вибрати самостійно.

4. Перевірити працездатність навченої мережі за допомогою моделювання.

Завдання 1: Класифікація літературних творів

Задача: класифікувати твори на повісті (клас 0) та романи (клас 1) за кількістю головних героїв та кількістю сторінок.

Завдання 2: Побудова графіка простору входів.

Умова класифікації:

- Повість: герої ≤ 5 , сторінки ≤ 200
- Роман: герої > 5 , сторінки > 200

Код для побудови і навчання персептрона:

% Вхідні дані

```
P = [3 2 5 4 1 6 8 7 9 6;  
     120 160 180 190 100 210 250 230 300 280];
```

% One-hot encoding для міток класів

```
T_classes = [0 0 0 0 0 1 1 1 1 1];  
T = full(ind2vec(T_classes + 1));
```

% Створення мережі (trainscg підтримує crossentropy)

```
net = patternnet(10);  
net.trainFcn = 'trainscg';  
net.trainParam.epochs = 200;
```

% Нормалізація

```
[P_norm, ps] = mapminmax(P);
```

% Навчання

```
[net, tr] = train(net, P_norm, T);
```

% Перевірка результатів

```
Y = net(P_norm);  
classes_predicted = vec2ind(Y) - 1;
```

% Візуалізація

```
view(net);  
plotperform(tr);
```

figure;

```
gscatter(P(1,:), P(2,:), T_classes, 'rb', 'o^');  
xlabel('Кількість героїв');  
ylabel('Кількість сторінок');  
title('Графік розділення повістей та романів (виправлений)');
```

```
[x, y] = meshgrid(0:0.1:10, 50:5:350);  
grid_points = [x(:)'; y(:)'];
```

```

grid_points_norm = mapminmax('apply', grid_points, ps);
Z = net(grid_points_norm);
[~, Z_class] = max(Z);
Z_class = reshape(Z_class, size(x));

hold on;
contour(x, y, Z_class-1, [0.5 0.5], 'k', 'LineWidth', 2);
legend('Півість', 'Роман', 'Роздільна межа');

```

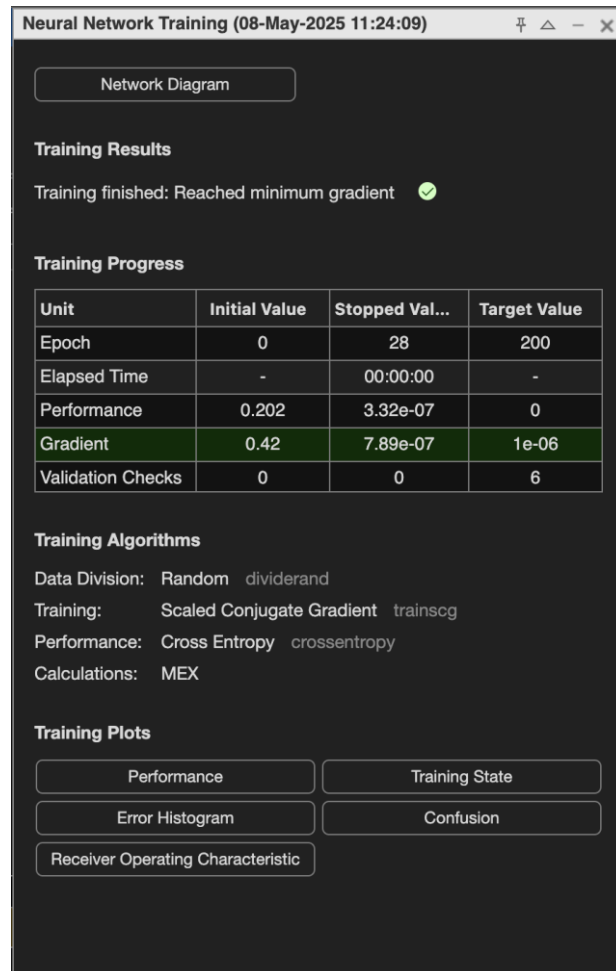


Рисунок 15.

Network diagram, performance:

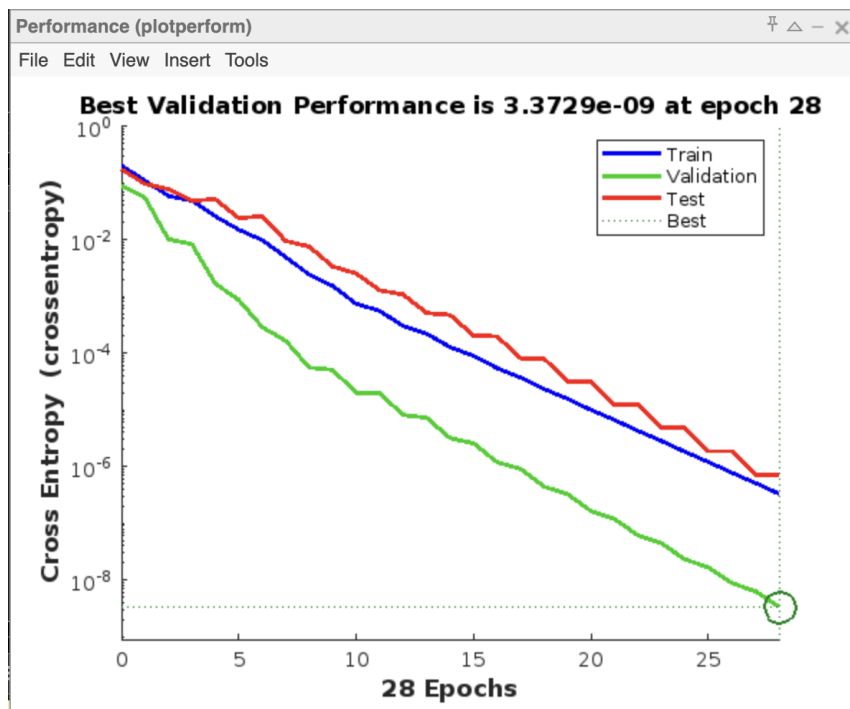


Рисунок 16. Performance

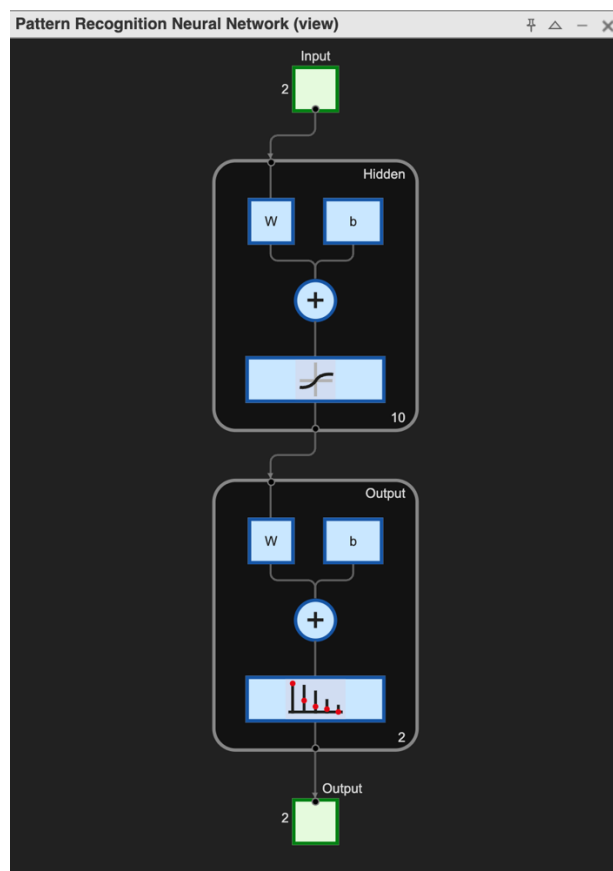


Рисунок 17. Network Diagram

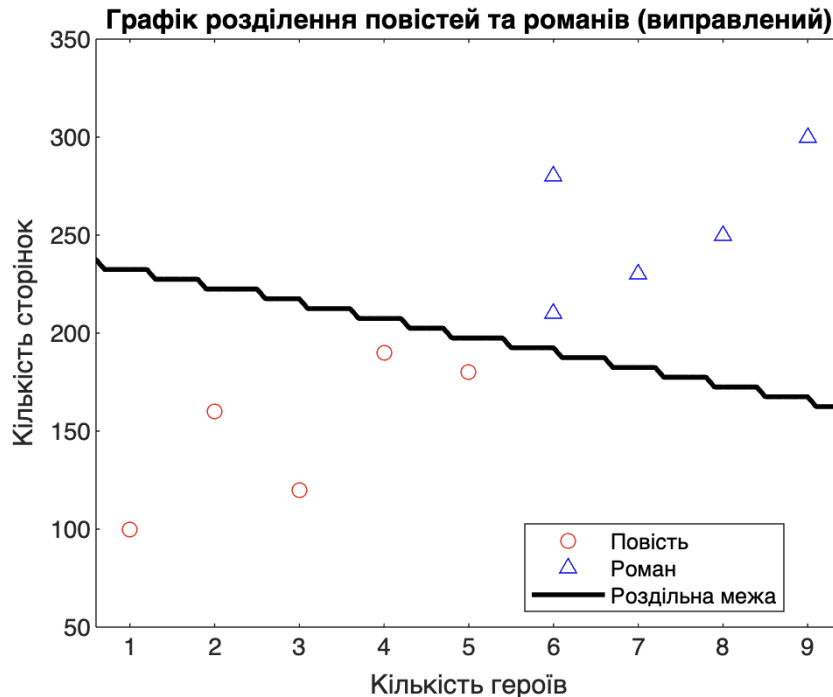


Рисунок 18. Графік розподілення повістей та романів

Завдання 3: Класифікація згідно з рисунком 2.1

Згенеруємо умовну навчальну вибірку відповідно до рисунку 2.1.

% Генерація даних

```
[X1, X2] = meshgrid(-5:0.5:5, -5:0.5:5);
```

```
P = [X1(:)'; X2(:)'];
```

% Реальні класи

```
T = double(X2(:) > -X1(:) - 1 & X2(:) > X1(:) - 1 & X2(:) < 3);
```

% Перетворення в one-hot encoding

```
T_onehot = full(ind2vec(T'+1));
```

% Створення та навчання нейронної мережі

```
net = patternnet(10);
```

```
net.trainFcn = 'trainscg';
```

```
net.trainParam.epochs = 300;
```

```
[net, tr] = train(net, P, T_onehot);
```

% Перевірка результатів

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

```
predicted_classes = vec2ind(Y)-1;
```

% Візуалізація Network Diagram та Performance

```
figure;  
view(net);
```

```
figure;  
plotperform(tr);
```

% Візуалізація класифікації (реальні vs прогнозовані)

```
figure;  
hold on;
```

% Реальні класи (порожні маркери)

```
gscatter(P(1,:), P(2,:), T, 'kb', 'ox', 8, 'off');
```

% Прогнозовані класи (заповнені маркери)

```
scatter(P(1,predicted_classes==0), P(2,predicted_classes==0), 20, 'r', 'filled',  
'Marker', '^');  
scatter(P(1,predicted_classes==1), P(2,predicted_classes==1), 20, 'g', 'filled',  
'Marker', 's');
```

```
xlabel('X(1)');  
ylabel('X(2)');  
title('Класифікація після навчання нейромережі');
```

```
legend('Клас 0 (реальний)', 'Клас 1 (реальний)', 'Клас 0 (прогноз)', 'Клас 1  
(прогноз)');  
hold off;
```

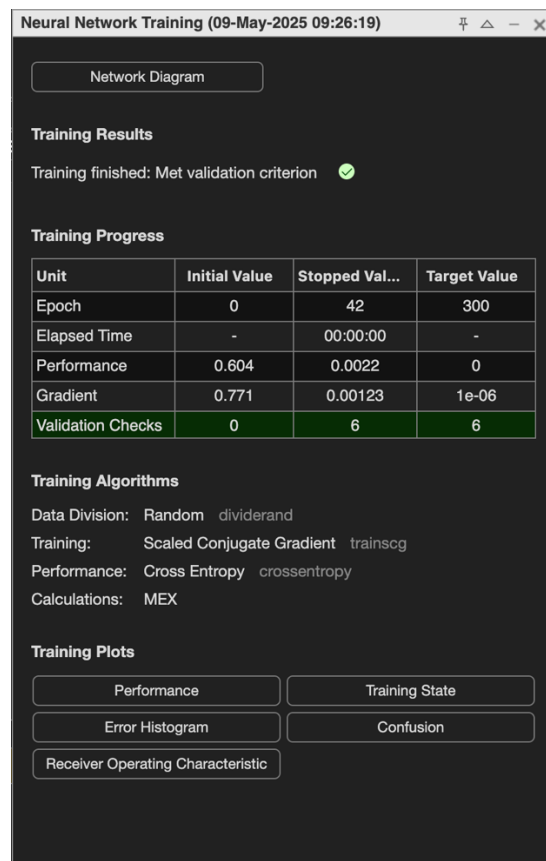


Рисунок 19.

Network diagram, performance:

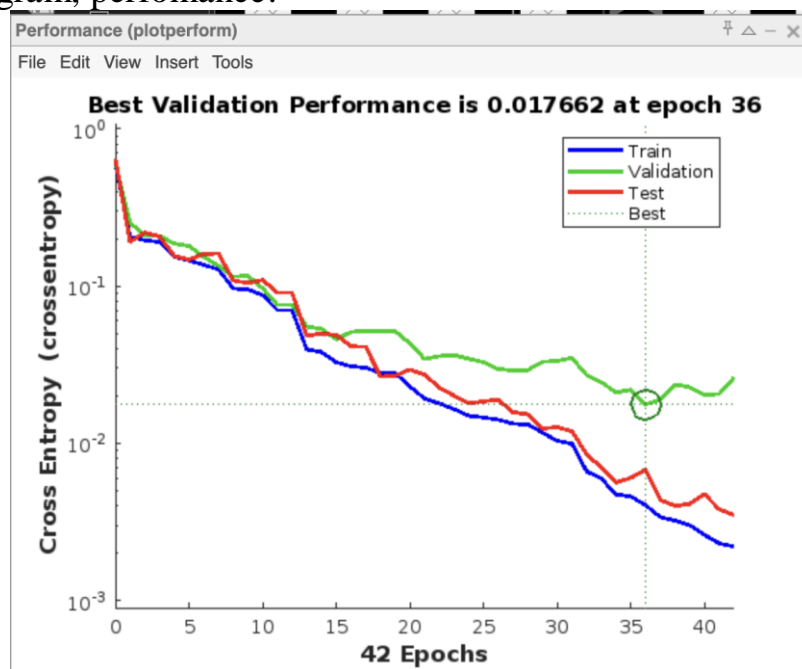


Рисунок 20. Performance

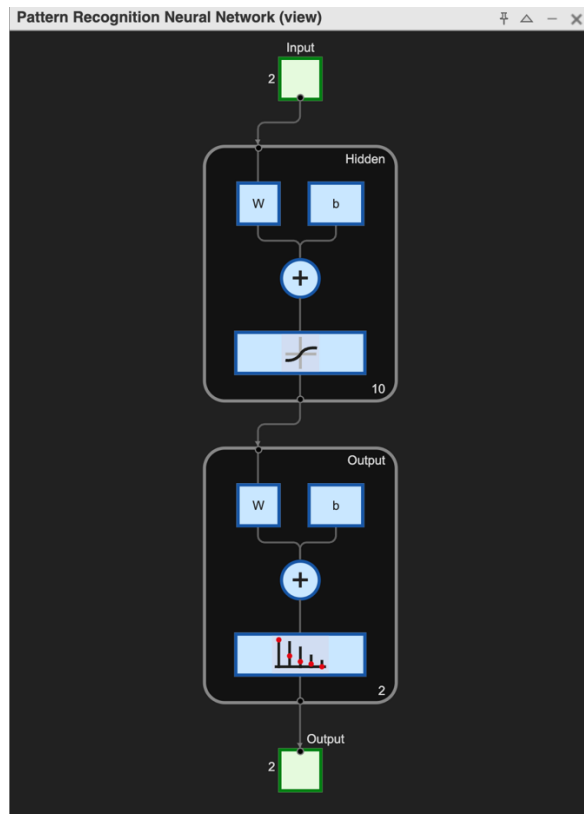


Рисунок 21. Network Diagram

Завдання 4: Перевірка роботи навченої мережі

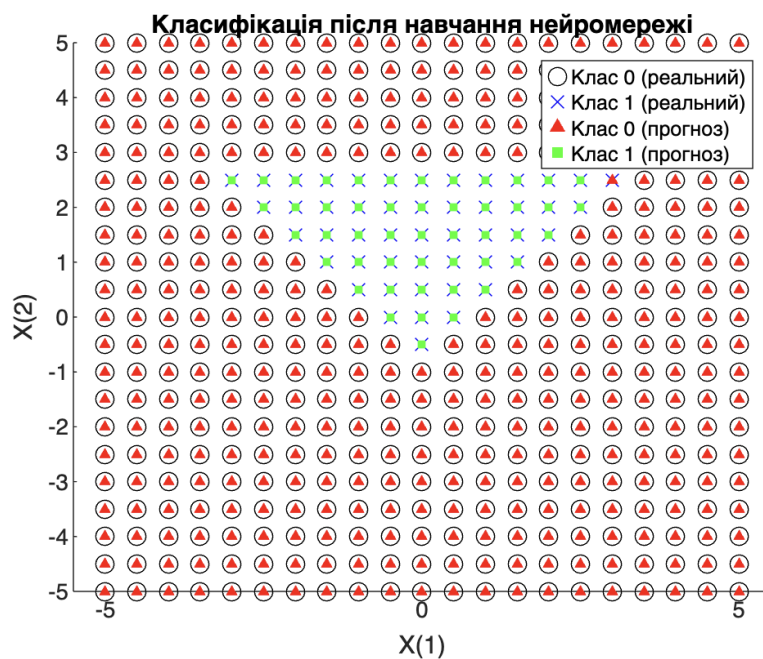


Рисунок 22. Класифікація після навчання нейромережі

Висновок

У ході виконання лабораторної роботи було побудовано та протестовано одношаровий персептрон для задач класифікації вхідних образів на основі двох і трьох ознак. Під час навчання було застосовано нормування вхідних векторів, що значно покращило ефективність роботи алгоритму за рахунок рівномірного внеску кожного вектора у зміну ваг.

Проведено порівняння класифікації у двокласовій та багатокласовій постановці, побудовано графіки з лініями та площинами розділення, що дозволяють візуально оцінити результат. Мережа показала гарну узгодженість результатів у випадках лінійно роздільних множин, а також виявлено обмеження персептрона щодо задач, де класи не є лінійно роздільними.

Здобуті знання та навички у створенні архітектури, підготовці навчальних даних, нормуванні та візуалізації результатів є основою для подальшого вивчення багатошарових мереж та глибокого навчання.