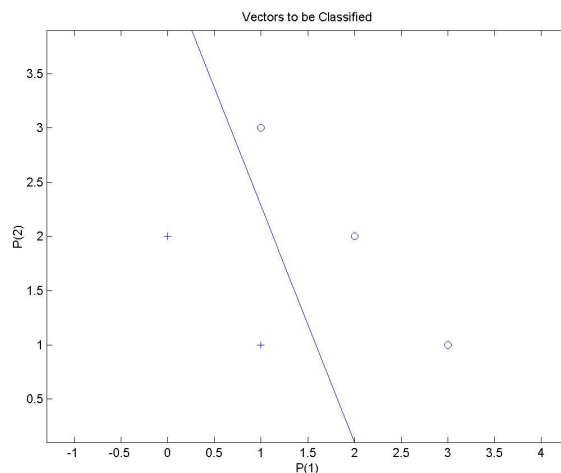


Uniwersytet Zielonogórski	Wykonali:	Grupa:	Nr ćwiczenia: 1	Ocena:
Laboratorium systemów obliczeń inteligentnych				
Temat ćwiczenia: Perceptyony proste.		Prowadzący:	Data wyk. ćw.	Data odd. spr.

Zadanie 1

Korzystając z reguły uczenia perceptronu wyznaczyć wagi perceptronu tak, aby rozpoznawał on elementy dwóch zbiorów określonych następująco: $A = \{(0, 2); (1, 1)\}$ i $B = \{(1, 3); (2, 2); (3, 1)\}$. Elementy zbioru A mają być klasyfikowane jako jedynki, zaś elementy zbioru B jako zera. Zbadać jak wpływa wartość kroku uczenia n na liczbę kroków procesu uczenia perceptronu. Po zakończeniu uczenia dokonać weryfikacji klasyfikatora. Sprawdzić, czy następujące elementy zbiorów są poprawnie klasyfikowane: $A = \{(-1, -1); (1, -1); (2.5, 0)\}$, $B = \{(2, 3); (2.5, -0.5); (3, 0)\}$.



a) krok uczenia $n = 0.1$

Liczba kroków uczących: 12

Parametry sieci:

$W = -0.2997 \quad -0.1377$

$B = 0.6137$

Zadany wzorzec

$u = 0$

1

$y = 1$

System rozpoznał poprawnie wzorzec

b) krok uczenia $n = 0.5$

Liczba kroków uczących: 20

Parametry sieci:

$W = -2.0280 \quad -1.2174$

$B = 4.0242$

Zadany wzorzec

$u = 0$
1

$y = 1$

System rozpoznał poprawnie wzorzec

c) krok uczenia $n = 0.05$

Liczba kroków uczących: 9

Parametry sieci:

$W = -0.2871 \quad -0.3630$

$B = 0.9428$

Zadany wzorzec

$u = 0$
1

$y = 1$

System rozpoznał poprawnie wzorzec

Weryfikacja klasyfikatora:

dla $A = (-1, -1) \quad y = 1$

dla $A = (1, -1) \quad y = 1$

dla $A = (2.5, 0) \quad y = 1$

dla $A = (2, 3) \quad y = 0$

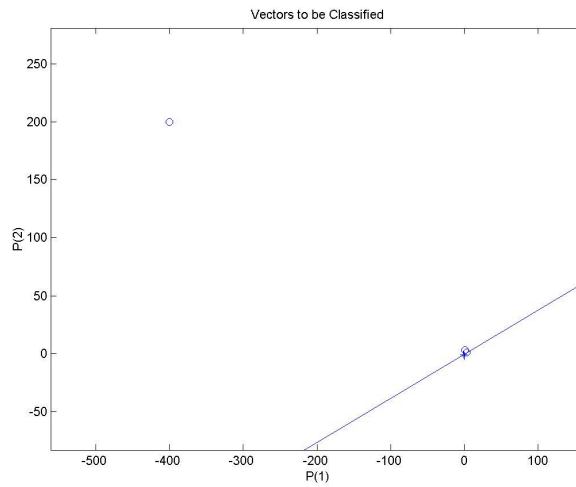
dla $A = (2.5, -0.5) \quad y = 0$

dla $A = (3, 0) \quad y = 0$

Zadanie 2

Przeanalizować problem uczenia się perceptronu prostego w przypadku, gdy jeden z klasyfikowanych elementów jest znacznie odległy od pozostałych. Przyjąć następujące dane: $A = \{(0, -2); (-1, -1)\}$ i $B = \{(1, 3); (-400, 200); (3, 1)\}$. Jak zachowuje się proces uczenia w takiej sytuacji? Co jest przyczyną takiego zachowania? Do tego samego problemu zastosować uczenie perceptronu przy pomocy znormalizowanej reguły uczenia perceptronu. Porównać wyniki obu eksperymentów.

a) reguła uczenia perceptronu



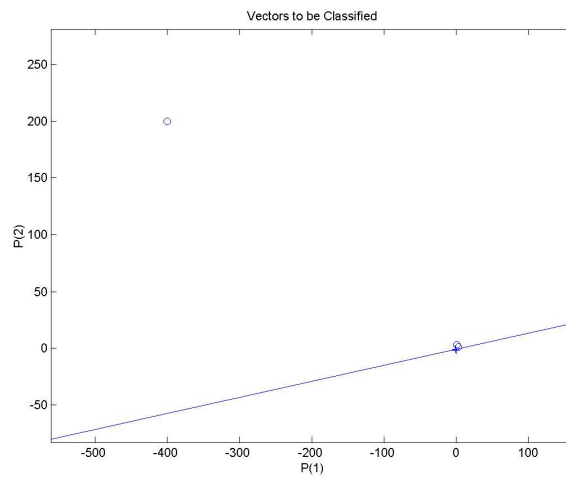
Liczba kroków uczących: 84

Parametry sieci:

$W = 11.9894 \quad -31.4691$

$B = -4.6161$

b) znormalizowana reguła uczenia perceptronu



Liczba kroków uczących: 8

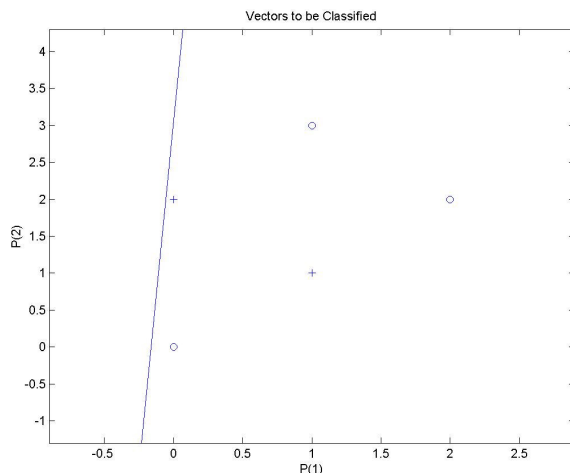
Parametry sieci:

$W = 0.0751 \quad -0.5293$

$B = -0.3646$

Zadanie 3

Przeprowadzić uczenie perceptronu prostego dla poniższych danych: $A = \{(0, 2); (1, 1)\}$ i $B = \{(1, 3); (2, 2); (0, 0)\}$. Dlaczego perceptron nie może dostroić swoich parametrów do tak zdefiniowanych danych uczących? W jaki sposób zmodyfikować sieć (w najmniejszym możliwym stopniu), aby problem można było rozwiązać?



Liczba kroków uczących: 100

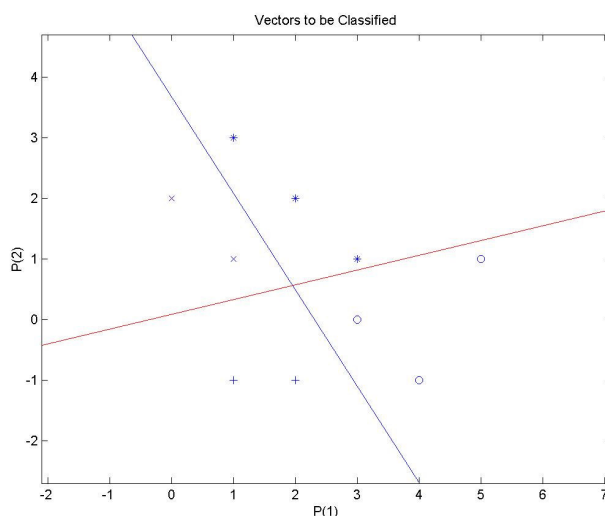
Parametry sieci:

$W = -0.4943 \quad 0.0263$

$B = -0.0803$

Zadanie 4

Dany jest problem klasyfikacji czterech zbiorów określonych na płaszczyźnie: $A = \{(0, 1); (1, 1)\}$, $B = \{(1, 3); (2, 2); (3, 1)\}$, $C = \{(2, -1); (1, -1)\}$ i $D = \{(5, 1); (4, -1); (3, 0)\}$. Jakiej struktury perceptronu prostego należy użyć w tym przypadku? W jakiej postaci może być przedstawione wyjście klasyfikatora?



Liczba kroków uczących: 10

Parametry sieci:

W = -0.3222 -0.2026
 -0.2945 1.2076

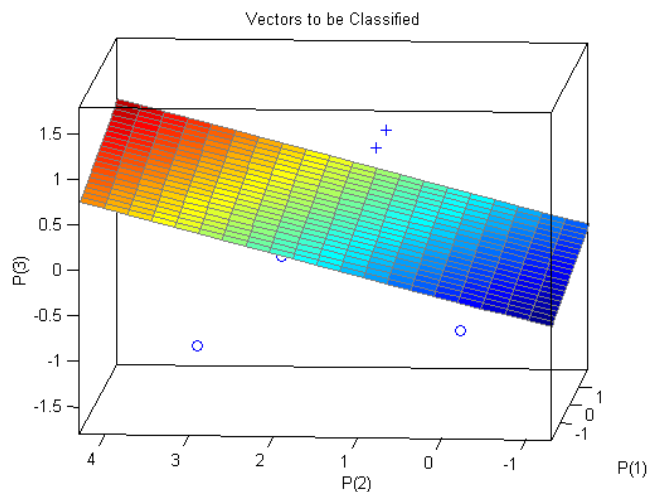
B = 0.7444
 -0.1024

Należało użyć struktury składającej się z dwóch wejść i dwóch wyjść

Wyjście klasyfikatora może być przedstawione w postaci 00, 01, 10, 11

Zadanie 5

Dany jest problem klasyfikacji dwóch zbiorów określonych w przestrzeni trójwymiarowej $A = \{(0, 1, 1); (1, 1, 1)\}$ i $B = \{(-1, 3, -1); (-1, 2, 0); (0, 0, -1)\}$. Określić liczbę wejść i wyjść perceptronu prostego zdolnego rozwiązać ten problem? Jaka postać przyjmuje granica decyzyjna?



Liczba kroków uczących: 2

Parametry sieci:

W = -0.0680 -0.1627 0.6924

B = 0.0503

Granica decyzyjna przyjmuje postać hiperpłaszczyzny

Zadanie 6

Zaprojektować system zdolny do wykonania operacji logicznych NOT, OR i AND. Wziąć pod uwagę większą niż 2 liczbę wejść. Wszystkie operacje logiczne powinny być wykonywane przez jedna sieć.

L.P.	WEJŚCIA	NOT	OR	AND
1	000	1	0	0
2	001	1	1	0
3	010	1	1	0
4	011	1	1	0
5	100	0	1	0
6	101	0	1	0
7	110	0	1	0
8	111	0	1	1

Skrypt:

```
% Zdefiniowanie trzy-elementowych wejściowych wzorców uczących
U = [ 0 0 0 0 1 1 1 1;
      0 0 1 1 0 0 1 1;
      0 1 0 1 0 1 0 1];
```

```
% Zdefiniowanie wyjściowych wzorców uczących
Y = [1 1 1 1 0 0 0 0;
      0 1 1 1 1 1 1 1;
      0 0 0 0 0 0 0 1];
```

```
% Położenie wzorców na płaszczyźnie
plotpv(U,Y),pause(1)
```

```
% Zdefiniowanie sieci neuronowej. Sieć posiada
% dwa wejścia i jedno wyjście
```

```
% Dobór parametrów początkowych sieci
[W,B] = initp(U,Y);
```

```
% Początkowa granica decyzyjna
H=plotpc(W,B);pause(1)
```

```
%Warunki początkowe algorytmu
E=[1 1 1 1 1];
licznik=0;
```

```
% Krok uczenia
n=0.1;
```

```
while licznik<100,
```

```
    licznik=licznik+1; % Licznik iteracji
```

```
    % Warunek jakościowy wyjścia z pętli
    if any(E)==0, break, end
```

```
    fi=W*U;          % Obliczenie potencjału membranowego
    Ys=hardlim(fi,B); % Obliczenie wyjścia
```

```
    % Porównanie wyjść sieci z wartościami wzorcowymi
    E=Y-Ys;
```

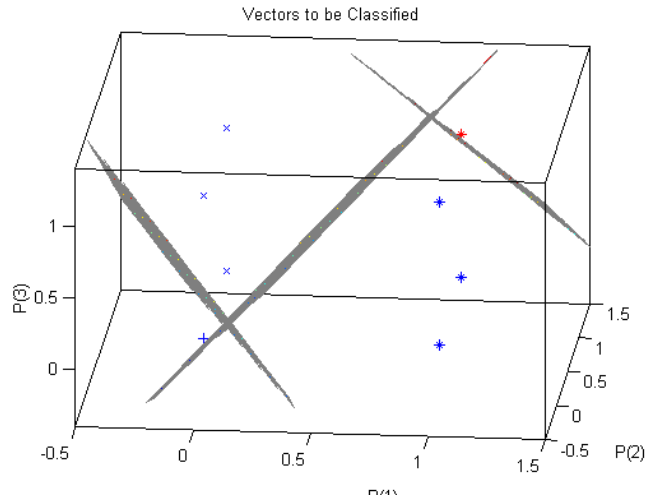
```
    % Faza uczenia
    [dW,dB] = learnp(U,E);
```

```
    % Korekta parametrów sieci
    W=W+n*dW; B=B+n*dB;
```

```
    % Rysowanie prostej decyzyjnej
    H=plotpc(W,B,H);pause2(0.1)
```

end

```
clc
disp(['Liczba kroków uczących: ', num2str(licznik)])
disp('Parametry sieci:');
W
B
```



Liczba kroków uczących: 18

Parametry sieci:

W =

-0.8571	0.1746	0.4878
0.2047	0.0730	0.0983
0.2515	0.1235	0.1974

B =

0.0229
-0.0312
-0.5616