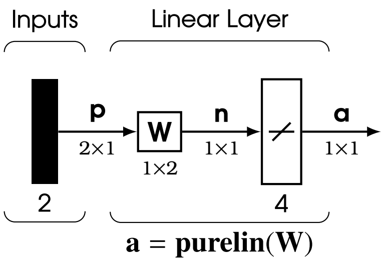
Κεφάλαιο 1

Πρόβλημα 1

1.1 Draw the network diagram for a linear associator network that could be trained on these patterns



1.2 Use the Hebb rule to find the weights of the network

The first step is to create the **P** and **T** matrices

$$\mathbf{P} = \begin{bmatrix} 3 & 6 & -6 \\ 6 & 3 & 3 \end{bmatrix} \tag{1.1}$$

$$\mathbf{T} = \begin{bmatrix} 75 & 75 & -75 \end{bmatrix} \tag{1.2}$$

The we compute the weight

$$\mathbf{W}^h = \mathbf{T} \cdot \mathbf{P}^T = \begin{bmatrix} 75 & 75 & -75 \end{bmatrix} \cdot \begin{bmatrix} 3 & 6 \\ 6 & 3 \\ -6 & 3 \end{bmatrix} = \begin{bmatrix} 1125 & 450 \end{bmatrix} \tag{1.3}$$

Εναλλακτικά

$$\Delta W_{i,j}(t) = a x_i(t) \cdot y_i(t) \Rightarrow (3 \cdot 75 + 6 \cdot 75) + (6 \cdot 75 + 3 \cdot 75) + (-6 \cdot (-75) + 3 \cdot (-75)) = 675 + 675 + 225 = 1575 \tag{1.4}$$

Έπειτα της λύσης της άσκησης παρατήρησα πως τα διανύσματα δεν βρίσκονται σε ορθογώνια κανονική μορφή , συνεπώς διαιρώ με το μέτρο $\sqrt{3^2 + 6^2} = \sqrt{9 + 36} = \sqrt{45}$:

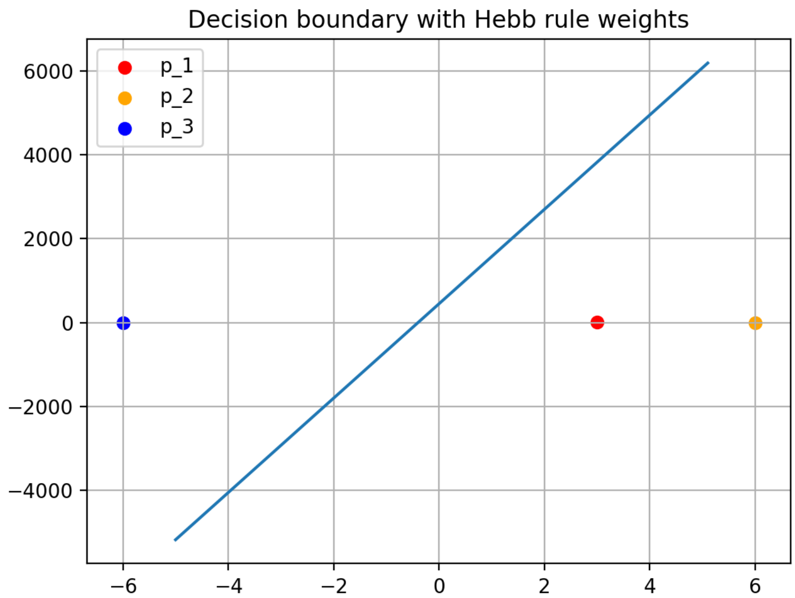
1.
$$p'_1 = \begin{bmatrix} \frac{3}{\sqrt{45}} \\ \frac{6}{\sqrt{45}} \end{bmatrix} \tag{1.5}$$

2.
$$p'_2 = \begin{bmatrix} \frac{6}{\sqrt{45}} \\ \frac{3}{\sqrt{45}} \end{bmatrix} \tag{1.6}$$

3.
$$p'_3 = \begin{bmatrix} \frac{-6}{\sqrt{45}} \\ \frac{3}{\sqrt{45}} \end{bmatrix} \tag{1.7}$$

1.3 Find and sketch the decision boundary for the network with the Hebb rule weights

Note that normalizes weight data and decision boundary is not displayed below . As it is easily seen at image 1.1, the boundary separates correct the patterns .



Εικόνα 1.1: *Problem 1 Decision Boundary*

1.4 Use the pseudo-inverse rule to find the weights of the network

For the pseudo-inverse rule we use equation :

$$\mathbf{W} = \mathbf{T} \cdot \mathbf{P}^{+} \tag{1.8}$$

όπου

computed by software

$$P^{+} = (P^T P)^{-1} P = \begin{bmatrix} -18.11215 & -6.820000 \\ -9.39148 & 5.81377 \\ -4.91934 & -9.39148 \end{bmatrix}$$

Συνεπώς

$$W = \begin{bmatrix} 75 & 75 & -75 \end{bmatrix} \cdot \begin{bmatrix} -18.11215 & -6.820000 \\ -9.39148 & 5.81377 \\ -4.91934 & -9.39148 \end{bmatrix} = \begin{bmatrix} -1693.82149295628 & 628.864118679875 \end{bmatrix} \tag{1.10}$$

1.4.1 Describe the difference between this boundary and the Hebb rule boundary

Παρατηρώ ότι το boundary του Hebb rule διαχωρίζει ορθότερα τα δεδομένα μας .

1.5 Python code

Για να επαληθεύσω τους ανωτέρω υπολογισμούς μου , έγραψα τον κάτωθι κώδικα σε Python , που υπολογίζει τα βάρη του δικτύου με βάση τον κανόνα του Hebb και σχεδιάζει το Decision boundary .

```
# CE418: N e u r o fuzzy Computing
#
# Evangelos Stamos
# 02338
# e s t a m o s @e c e .uth. gr

# P r o b l e m 01
#

import numpy as np

import matplotlib.pyplot as plt

RefPatters = [[3, 6], [6,3], [-6, 3]]
Targets = np.array([75, 75, -75])

P = np.array([[3, 6, -6],[6, 3,  3]])


# B | Use the Hebb rule to find the weights of the network
Weights = np.dot(Targets, P.transpose())

print('Final weights :'  , Weights , '\n')

# C | Find and sketch the decision boundary for the network with the Hebb rule weights
```

```
# Does the boundary separate the patterns? Demonstrate
x = np.linspace(-5,5.1)
y = Weights[0]*x + Weights[1]
plt.title('Decision boundary with Hebb rule weights')
plt.plot(x,y)
plt.scatter(3, 6, color='red', label='p_1')
plt.scatter(6, 3, color='orange', label='p_2')
plt.scatter(-6, 3, color='blue', label='p_3')

plt.grid()
plt.legend()
plt.show()
```