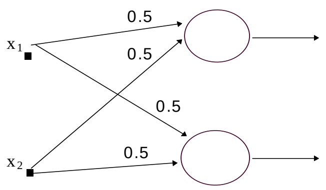
FIT5186 Examples from Lecture 6

Winner-Take-All

Find 2 clusters from the following data points:
(0.8, 0.6), (0.9, 0.5), (0.3, 0.2), (0.2, 0.3), (1, 0.7)



Select the output neuron with the maximum net input as the winner.

Weights are initialised to 0.5, c = 0.5.

$$\text{Input } (0.8, 0.6): \quad net = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix} \begin{pmatrix} 0.8 \\ 0.6 \end{pmatrix} = \begin{pmatrix} 0.7 \\ 0.7 \end{pmatrix}$$

No clear winner - arbitrarily choose neuron 1.

$$\Delta \mathbf{W}_1 = c(\mathbf{x} - \mathbf{W_1}) = 0.5 \times [\begin{pmatrix} 0.8 \\ 0.6 \end{pmatrix} - \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}] = \begin{pmatrix} 0.15 \\ 0.05 \end{pmatrix} \quad \text{so } \mathbf{W} = \begin{pmatrix} 0.65 & 0.55 \\ 0.5 & 0.5 \end{pmatrix}$$

$$\text{Input } (0.9, 0.5): \quad net = \begin{pmatrix} 0.65 & 0.55 \\ 0.5 & 0.5 \end{pmatrix} \begin{pmatrix} 0.9 \\ 0.5 \end{pmatrix} = \begin{pmatrix} 0.86 \\ 0.7 \end{pmatrix} \quad \text{neuron 1 is the winner}$$

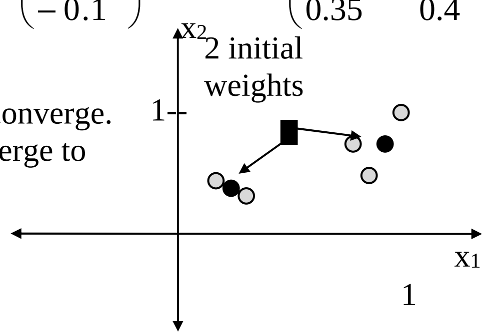
$$\Delta \mathbf{W}_1 = c(\mathbf{x} - \mathbf{W_1}) = 0.5 \times [\begin{pmatrix} 0.9 \\ 0.5 \end{pmatrix} - \begin{pmatrix} 0.65 \\ 0.55 \end{pmatrix}] = \begin{pmatrix} 0.125 \\ -0.025 \end{pmatrix} \quad \text{so } \mathbf{W} = \begin{pmatrix} 0.775 & 0.525 \\ 0.5 & 0.5 \end{pmatrix}$$

$$\text{Input } (0.3, 0.2): \quad net = \begin{pmatrix} 0.3375 \\ 0.25 \end{pmatrix} \quad \text{neuron 1 is the winner}$$

$$\Delta \mathbf{W}_1 = c(\mathbf{x} - \mathbf{W_1}) = 0.5 \times [\begin{pmatrix} 0.3 \\ 0.2 \end{pmatrix} - \begin{pmatrix} 0.775 \\ 0.525 \end{pmatrix}] = \begin{pmatrix} -0.2375 \\ -0.1625 \end{pmatrix} \quad \text{so } \mathbf{W} = \begin{pmatrix} 0.5375 & 0.3625 \\ 0.5 & 0.5 \end{pmatrix}$$

$$\text{Input } (0.2, 0.3): \quad net = \begin{pmatrix} 0.2163 \\ 0.25 \end{pmatrix} \quad \text{neuron 2 is the winner}$$

$$\Delta \mathbf{W}_2 = c(\mathbf{x} - \mathbf{W}_2) = 0.5 \times [\begin{pmatrix} 0.2 \\ 0.3 \end{pmatrix} - \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}] = \begin{pmatrix} -0.15 \\ -0.1 \end{pmatrix} \quad \text{so } \mathbf{W} = \begin{pmatrix} 0.5375 & 0.3625 \\ 0.35 & 0.4 \end{pmatrix}$$



Keep presenting the inputs until the weights converge.
After approximately 30 epochs, weights converge to

$$\mathbf{W} = \begin{pmatrix} \overbrace{0.9 \quad 0.6}^{\mathbf{p}_1} \\ \underbrace{0.25 \quad 0.25}_{\mathbf{p}_2} \end{pmatrix}$$

$\mathbf{p}_1$ and $\mathbf{p}_2$ are “prototype” points for the two apparent clusters.

Self Organising Feature Map

Find clusters from the following data points: (0.8, 0.6), (0.9, 0.5), (0.3, 0.2), (0.2, 0.3), (1, 0.7) – the same as the winner-take-all example.
Suppose we don't know the number of clusters p.
Choose a 1-dimensional array of 3 neurons (trying to find $p \leq 3$ clusters).
Initialise weights to random values (around 0.5 to be unbiased).
Initialise neighbourhood size, and other parameters so that:

$$\mathbf{W} = \begin{pmatrix} 0.48 & 0.51 \\ 0.50 & 0.47 \\ 0.51 & 0.50 \end{pmatrix};\ N_m(0) = 2;\ \alpha = 0.5;\ \sigma^2 = 1 \text{ for all time t}$$

Present input (0.8,0.6): $d_1 = \sqrt{(0.8-0.48)^2 + (0.6-0.51)^2} = 0.332$

$d_2 = \sqrt{(0.8-0.5)^2 + (0.6-0.47)^2} = 0.327$ $\quad d_3 = \sqrt{(0.8-0.51)^2 + (0.6-0.5)^2} = 0.307$

So the winning neuron (minimum distance) is neuron 3 (m = 3)

$$\alpha(\mathrm{N}_1(t)) = \alpha\exp(-\|r_1 - r_3\|/\sigma^2) = \alpha\exp(-2/1) = 0.07$$

$$\alpha(\mathrm{N}_2(t)) = \alpha\exp(-\|r_2 - r_3\|/\sigma^2) = \alpha\exp(-1/1) = 0.18$$

$$\alpha(\mathrm{N}_3(t)) = \alpha\exp(-\|r_3 - r_3\|/\sigma^2) = \alpha\exp(-0/1) = 0.5$$

$$\Delta\mathbf{W_1} = \alpha(\mathrm{N}_1(t))[\mathbf{x} - \mathbf{W_1}] = 0.07 \times [\begin{pmatrix} 0.8 \\ 0.6 \end{pmatrix} - \begin{pmatrix} 0.48 \\ 0.51 \end{pmatrix}] = \begin{pmatrix} 0.02 \\ 0.01 \end{pmatrix}$$

$$\Delta\mathbf{W}_2 = 0.18 \times [\begin{pmatrix} 0.8 \\ 0.6 \end{pmatrix} - \begin{pmatrix} 0.5 \\ 0.47 \end{pmatrix}] = \begin{pmatrix} 0.05 \\ 0.02 \end{pmatrix} \qquad \Delta\mathbf{W}_3 = 0.5 \times [\begin{pmatrix} 0.8 \\ 0.6 \end{pmatrix} - \begin{pmatrix} 0.51 \\ 0.5 \end{pmatrix}] = \begin{pmatrix} 0.15 \\ 0.05 \end{pmatrix}$$

So $\mathbf{W} = \begin{pmatrix} 0.5 & 0.52 \\ 0.55 & 0.49 \\ 0.66 & 0.55 \end{pmatrix}$ after 1st input. Check the rest of the weight updates yourself:

Present input (0.3,0.2): winner is m = 1

$$\mathbf{W} = \begin{pmatrix} 0.4 & 0.36 \\ 0.5 & 0.44 \\ 0.63 & 0.53 \end{pmatrix}$$

Present input (1,0.7): winner is m = 3

$$\mathbf{W} = \begin{pmatrix} 0.44 & 0.38 \\ 0.59 & 0.49 \\ 0.82 & 0.62 \end{pmatrix}$$

Present input (0.9,0.5): winner is m = 3

$$\mathbf{W} = \begin{pmatrix} 0.47 & 0.39 \\ 0.65 & 0.49 \\ 0.86 & 0.56 \end{pmatrix}$$

Present input (0.2,0.3): winner is m = 1

$$\mathbf{W} = \begin{pmatrix} 0.33 & 0.34 \\ 0.57 & 0.46 \\ 0.81 & 0.54 \end{pmatrix}$$

This is the end of 1 epoch; Repeat for say 50 epochs; Shrink $\mathrm{N_m}(t) = 1$; Repeat for 50 epochs; Shrink $\mathrm{N_m}(t) = 0$ (only the winning neuron gets updated).

Using NeuroShell2 to Complete the learning process, the final weights are:

$$\mathrm{W} = \begin{pmatrix} 0.19 & 0.22 \\ 0.55 & 0.52 \\ 0.83 & 0.76 \end{pmatrix}$$

When applying this trained network to the original data points (using a winner -take-all classification), the output is:

0	0	1	(0.8, 0.6)
0	0	1	(0.9, 0.5)
1	0	0	(0.3, 0.2)
1	0	0	(0.2, 0.3)
0	0	1	(1, 0.7)

Cluster 2 is never used

The SOFM has detected p = 2 clusters.