MOD 202: Excercise Sheet 1

Ciprian Bangu

February 19 2024

1. Perceptrons

   (a) Yes, the two sets are linearly seperable. This can be demonstrated by the following plot:

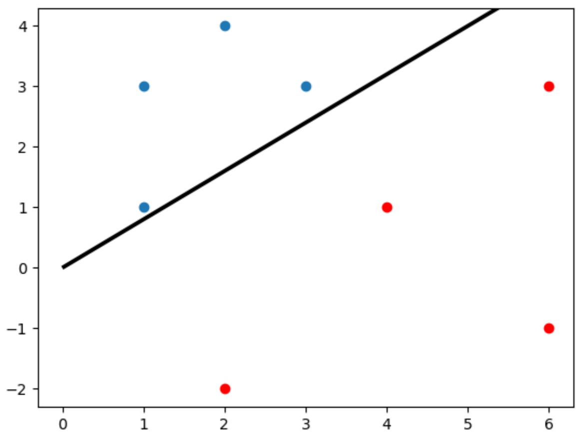

Figure 1: 2D plot of the two sets

Using the Perceptron Rule, we can find a readout weight vector that allows a binary neuron to classify the two sets. The Perceptron Rule is given by first by the computation of the output of the neuron, y_k, for each input vector, $\vec{x}^{(k)}$ (assuming the normalization factor $b = 0$)

$$y_k = H(\sum_{i=1}^{N} w_i x_i^{(k)}) = H(\vec{w}.\vec{x} - b) = H(\vec{w}.\vec{x} - 0) \tag{1}$$

where H is the Heaviside step function, and N is the number of input features. The Perceptron Rule then updates the weight vector, $\vec{w}$, according to the following rule:

$$w_i(t+1) = w_i(t) + (d_k - y_k)x_i^{(k)} \tag{2}$$

$$\vec{w}_{t+1} = \vec{w}_t.r\vec{x}_t \tag{3}$$

Where d_k is the desired output, and y_k is the actual output at the timestep (their difference later denoted by r). Let us also define our patterns. For this excercise, we will classify blue points as 0, and red points as 1. We can then define the following patterns:

$$\left\{\left(\begin{bmatrix}1\\1\end{bmatrix},0\right),\left(\begin{bmatrix}4\\1\end{bmatrix},1\right),\left(\begin{bmatrix}1\\3\end{bmatrix},0\right),\left(\begin{bmatrix}6\\3\end{bmatrix},1\right),\left(\begin{bmatrix}3\\3\end{bmatrix},0\right),\left(\begin{bmatrix}2\\-2\end{bmatrix},1\right),\left(\begin{bmatrix}2\\4\end{bmatrix},0\right),\left(\begin{bmatrix}6\\-1\end{bmatrix},1\right)\right\}$$

Since the pattern is two dimensional, we need to initialize a 2 dimensonal weight vector, $\vec{w}$, to update. We can initialize $\vec{w}$ randomly. To begin, let us take $\vec{w} = \begin{bmatrix}-1\\-1\end{bmatrix}$.

Updating when the $y_k \neq d_k$:

(a) $d_2 = 1$

$$y_2 = H\left(\begin{bmatrix}-1\\-1\end{bmatrix}\cdot\begin{bmatrix}4\\1\end{bmatrix}\right)$$

$$y_2 = H(-4-1) = H(-5) = 0 \neq 1$$

applying the update rule:

$$\vec{w_3} = \left(\begin{bmatrix}-1\\-1\end{bmatrix} + (1)\begin{bmatrix}4\\1\end{bmatrix}\right)$$

$$\vec{w_3} = \begin{bmatrix}3\\0\end{bmatrix}$$

(b) $d_3 = 0$

$$y_3 = H\left(\begin{bmatrix}3\\0\end{bmatrix}\cdot\begin{bmatrix}1\\3\end{bmatrix}\right)$$

$$y_3 = H(3+0) = H(3) = 1 \neq 0$$

applying the update rule:

$$\vec{w_4} = \left(\begin{bmatrix}3\\0\end{bmatrix} + (-1)\begin{bmatrix}1\\3\end{bmatrix}\right)$$

$$\vec{w_4} = \begin{bmatrix}2\\-3\end{bmatrix}$$

Running the computation through the remaining points, we get that $\begin{bmatrix}2\\-3\end{bmatrix}$ is the final weight vector that can classify the two sets.

(b) Yes, these points are linearly seperable. This can be demonstrated by the following plot:

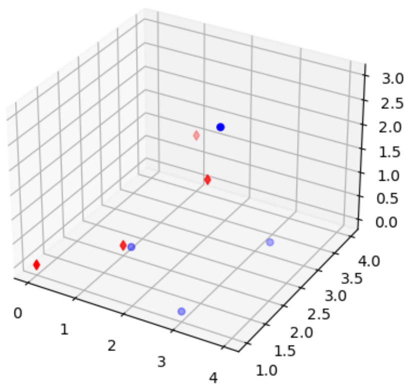

Figure 2: 3D plot of the two sets

It can also be demonstrated by the termination of the algorithm. Classifying the blue points as 0s, and the red points as 1s, we can define the following patterns:

$$\left\{\left(\begin{bmatrix}2\\1\\1\end{bmatrix},0\right),\left(\begin{bmatrix}1\\4\\1\end{bmatrix},1\right),\left(\begin{bmatrix}3\\2\\3\end{bmatrix},0\right),\left(\begin{bmatrix}2\\3\\1\end{bmatrix},1\right),\left(\begin{bmatrix}4\\2\\1\end{bmatrix},0\right),\left(\begin{bmatrix}1\\2\\0\end{bmatrix},1\right),\left(\begin{bmatrix}3\\1\\0\end{bmatrix},0\right),\left(\begin{bmatrix}0\\1\\0\end{bmatrix},1\right)\right\}$$

Since we have 3 dimensional inputs, we need to a 3 dimensional weight vector. Initializing randomly, we can take $\vec{w} = \begin{bmatrix}1\\1\\1\end{bmatrix}$. Taking b = 0, we can then apply the Perceptron Rule to update the weight vector, when $y_k \neq d_k$:

(a) $d_1 = 0$

$$y_1 = H\left(\begin{bmatrix}1\\1\\1\end{bmatrix}.\begin{bmatrix}2\\1\\1\end{bmatrix}\right) = H(4) = 1 \neq 0$$

Applying the update rule:

$$\vec{w}_2 = \begin{bmatrix}1\\1\\1\end{bmatrix} + (-1)\begin{bmatrix}2\\1\\1\end{bmatrix} = \begin{bmatrix}-1\\0\\0\end{bmatrix}$$

(b) $d_2 = 1$

$$y_2 = H\left(\begin{bmatrix}-1\\0\\0\end{bmatrix}.\begin{bmatrix}1\\4\\1\end{bmatrix}\right) = H(-1) = 0 \neq 1$$

updating:

$$\vec{w}_3 = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} + (1) \begin{bmatrix} 1 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 4 \\ 1 \end{bmatrix}$$

(c) $d_3 = 0$

$$y_3 = H\left(\begin{bmatrix} 0 \\ 4 \\ 1 \end{bmatrix} . \begin{bmatrix} 3 \\ 2 \\ 3 \end{bmatrix}\right) = H(11) = 1 \neq 0$$

updating:

$$\vec{w}_4 = \begin{bmatrix} 0 \\ 4 \\ 1 \end{bmatrix} + (-1) \begin{bmatrix} 3 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} -3 \\ 2 \\ -2 \end{bmatrix}$$

(d) $d_4 = 1$

$$y_4 = H\left(\begin{bmatrix} -3 \\ 2 \\ -2 \end{bmatrix} . \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}\right) = H(-2) = 0 \neq 1$$

updating:

$$\vec{w}_5 = \begin{bmatrix} -3 \\ 2 \\ -2 \end{bmatrix} + (1) \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 5 \\ -1 \end{bmatrix}$$

(e) $d_5 = 0$

$$y_5 = H\left(\begin{bmatrix} -1 \\ 5 \\ -1 \end{bmatrix} . \begin{bmatrix} 4 \\ 2 \\ 1 \end{bmatrix}\right) = H(5) = 1 \neq 0$$

updating:

$$\vec{w}_6 = \begin{bmatrix} -1 \\ 5 \\ -1 \end{bmatrix} + (-1) \begin{bmatrix} 4 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} -5 \\ 3 \\ -2 \end{bmatrix}$$

(f) $d_4 = 1$

$$y_4 = H\left(\begin{bmatrix} -5 \\ 3 \\ -2 \end{bmatrix} . \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}\right) = H(-3) = 0 \neq 1$$

updating:

$$\vec{w}_7 = \begin{bmatrix} -5 \\ 3 \\ -2 \end{bmatrix} + (1) \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} -3 \\ 6 \\ -1 \end{bmatrix}$$

(g) $d_3 = 0$

$$y_3 = H\left(\begin{bmatrix} -3 \\ 6 \\ -1 \end{bmatrix} . \begin{bmatrix} 3 \\ 2 \\ 3 \end{bmatrix}\right) = H(0) = 1 \neq 0$$

updating:

$$\vec{w}_8 = \begin{bmatrix} -3 \\ 6 \\ -1 \end{bmatrix} + (-1) \begin{bmatrix} 3 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} -6 \\ 4 \\ -4 \end{bmatrix}$$

(h) $d_4 = 1$

$$y_4 = H\left(\begin{bmatrix} -6 \\ 4 \\ -4 \end{bmatrix} . \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} \right) = H(-4) = 0 \neq 1$$

updating:

$$\vec{w}_9 = \begin{bmatrix} -6 \\ 4 \\ -4 \end{bmatrix} + (1) \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} -4 \\ 7 \\ -3 \end{bmatrix}$$

Running the rest of the computation, we get that the weight vector $\begin{bmatrix} -4 \\ 7 \\ -3 \end{bmatrix}$ is the final weight vector that can classify the two sets.

2. Hoppfield Networks

(a) The Hoppfield rule asserts that the weight between two neurons, i and j, where N is the number of neurons, and P is the number of patterns. The weight matrix, W, is then given by:

$$w_{ij} = \frac{1}{N} \sum_{k=1}^{P} \xi_i^k \xi_j^k \tag{4}$$

In a recurrent network of five binary neurons, the pattern $\xi^* = (1, -1, -1, 1, -1)$ can be stored with the following synaptic weights:

$$W = \frac{1}{5} \begin{bmatrix} (1*1) & (-1*1) & (-1*1) & (1*1) & (-1*1) \\ (1*-1) & (-1*-1) & (-1*-1) & (1*-1) & (-1*-1) \\ (1*-1) & (-1*-1) & (-1*-1) & (1*-1) & (-1*-1) \\ (1*1) & (-1*1) & (-1*1) & (1*1) & (-1*1) \\ (1*-1) & (-1*-1) & (-1*-1) & (1*-1) & (-1*-1) \end{bmatrix}$$

$$W = \frac{1}{5} \begin{bmatrix} (1) & (-1) & (-1) & (1) & (-1) \\ (-1) & (1) & (1) & (-1) & (1) \\ (-1) & (1) & (1) & (-1) & (1) \\ (1) & (-1) & (-1) & (1) & (-1) \\ (-1) & (1) & (1) & (-1) & (1) \end{bmatrix}$$

$$W = \begin{bmatrix} 0.2 & -0.2 & -0.2 & 0.2 & -0.2 \\ -0.2 & 0.2 & 0.2 & -0.2 & 0.2 \\ -0.2 & 0.2 & 0.2 & -0.2 & 0.2 \\ 0.2 & -0.2 & -0.2 & 0.2 & -0.2 \\ -0.2 & 0.2 & 0.2 & -0.2 & 0.2 \end{bmatrix}$$

If the pattern is a fixed point, then the activity of each neuron in the network will not change, regardles of the activity of the other neurons in the network. i.e.,

$$\xi_i = sgn[\sum_{j=1}^{n} w_{ij}\xi_j(t)] \tag{5}$$

Solving the activity equation for the pattern $\xi^* = (1, -1, -1, 1, -1)$:

$$\xi_1 = sgn[(0.2 * 1) + (-0.2 * -1) + (-0.2 * -1) + (0.2 * 1) + (-0.2 * -1)] = sgn[1] = 1$$
$$\xi_2 = sgn[(-0.2 * 1) + (0.2 * -1) + (0.2 * -1) + (-0.2 * 1) + (0.2 * -1)] = sgn[-1] = -1$$
$$\xi_3 = sgn[(-0.2 * 1) + (0.2 * -1) + (0.2 * -1) + (-0.2 * 1) + (0.2 * -1)] = sgn[-1] = -1$$
$$\xi_4 = sgn[(0.2 * 1) + (-0.2 * -1) + (-0.2 * -1) + (0.2 * 1) + (-0.2 * -1)] = sgn[1] = 1$$
$$\xi_5 = sgn[(-0.2 * 1) + (0.2 * -1) + (0.2 * -1) + (-0.2 * 1) + (0.2 * -1)] = sgn[-1] = -1$$

$$(\xi_1, \xi_2, \xi_3, \xi_4, \xi_5) = (1, -1, -1, 1, -1) = \xi^*$$

Therefore, ξ^* is a fixed point of the network.

(b) The output of the network if the dynamics start from the initial state $\xi = (1, -1, -1, 1, -1)$ will be the same as the initial state, $\xi = (1, -1, -1, 1, -1)$, since it is a fixed point of the network. It is also given by the following equation:

$$\vec{y}(t + 1) = sgn[W.\vec{y}(t)] \tag{6}$$

$$sgn\left(\begin{bmatrix} 0.2 & -0.2 & -0.2 & 0.2 & -0.2 \\ -0.2 & 0.2 & 0.2 & -0.2 & 0.2 \\ -0.2 & 0.2 & 0.2 & -0.2 & 0.2 \\ 0.2 & -0.2 & -0.2 & 0.2 & -0.2 \\ -0.2 & 0.2 & 0.2 & -0.2 & 0.2 \end{bmatrix} . \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \\ -1 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \\ -1 \end{bmatrix}$$

(c) The Hoppfield rule to modify the synpatic weights such that $\eta^* = (1, 1, -1, -1, -1)$ is also memorized is the following:

$$w_{ij} = \frac{\xi_i^{(1)}\xi_j^{(1)} + \eta_i^{(1)}\eta_j^{(1)}}{N} \tag{7}$$

$$W = \frac{1}{5}\begin{bmatrix} (1 * 1 + 1 * 1) & ... & ... & ... & (-1 * 1 + -1 * 1) \\ (1 * -1 + 1 * 1) & ... & ... & ... & (-1 * -1 + -1 * 1) \\ (1 * -1 + 1 * -1) & ... & ... & ... & (-1 * -1 + -1 * -1) \\ (1 * 1 + 1 * -1) & ... & ... & ... & (-1 * 1 + -1 * -1) \\ (1 * -1 + 1 * -1) & ... & ... & ... & (-1 * -1 + -1 * -1) \end{bmatrix}$$

$$W = \frac{1}{5}\begin{bmatrix} (2) & ... & ... & ... & (-2) \\ (0) & ... & ... & ... & (0) \\ (-2) & ... & ... & ... & (2) \\ (0) & ... & ... & ... & (0) \\ (-2) & ... & ... & ... & (2) \end{bmatrix}$$

$$W = \begin{bmatrix} 0.4 & 0 & -0.4 & 0 & -0.4 \\ 0 & 0.4 & 0 & -0.4 & 0 \\ -0.4 & 0 & 0.4 & 0 & 0.4 \\ 0 & -0.4 & 0 & 0.4 & 0 \\ -0.4 & 0 & 0.4 & 0 & 0.4 \end{bmatrix}$$

As in the previous case, we can verify that both ξ^* and η^* are fixed points of the network by verifying:

$$\xi_i = sgn[\sum_{j=1}^{n} w_{ij}\xi_j(t)] \tag{8}$$

and

$$\eta_i = sgn[\sum_{j=1}^{n} w_{ij}\eta_j(t)] \tag{9}$$

Regarding ξ^*:

$$\xi_1 = sgn[(0.4 * 1) + (0 * -1) + (-0.4 * -1) + (0 * 1) + (-0.4 * -1)] = sgn[1.2] = 1$$

$$\xi_2 = sgn[(0 * 1) + (0.4 * -1) + (0 * -1) + (-0.4 * 1) + (0 * -1)] = sgn[-0.8] = -1$$

$$\xi_3 = sgn[(-0.4 * 1) + (0 * -1) + (0.4 * -1) + (0 * 1) + (0.4 * -1)] = sgn[-1.2] = -1$$

$$\xi_4 = sgn[(0 * 1) + (-0.4 * -1) + (0 * -1) + (0.4 * 1) + (0 * -1)] = sgn[0.8] = 1$$

$$\xi_5 = sgn[(-0.4 * 1) + (0 * -1) + (0.4 * -1) + (0 * 1) + (0.4 * -1)] = sgn[-1.2] = -1$$

$$(\xi_1, \xi_2, \xi_3, \xi_4, \xi_5) = (1, -1, -1, 1, -1) = \xi^*$$

Therefore, ξ^* is a fixed point.
Regardig η^*:

$$\eta_1 = sgn[(0.4 * 1) + (0 * 1) + (-0.4 * -1) + (0 * -1) + (-0.4 * -1)] = sgn[1.2] = 1$$

$$\eta_2 = sgn[(0 * 1) + (0.4 * 1) + (0 * -1) + (-0.4 * -1) + (0 * -1)] = sgn[0.8] = 1$$

$$\eta_3 = sgn[(-0.4 * 1) + (0 * 1) + (0.4 * -1) + (0 * -1) + (0.4 * -1)] = sgn[-1.2] = -1$$

$$\eta_4 = sgn[(0 * 1) + (-0.4 * 1) + (0 * -1) + (0.4 * -1) + (0 * -1)] = sgn[-0.8] = -1$$

$$\eta_5 = sgn[(-0.4 * 1) + (0 * 1) + (0.4 * -1) + (0 * -1) + (0.4 * -1)] = sgn[-1.2] = -1$$

$$(\eta_1, \eta_2, \eta_3, \eta_4, \eta_5) = (1, 1, -1, -1, -1) = \eta^*$$

Therefore, η^* is a fixed point.

(d) If the dynamics start from $\xi = (1, -1, 1, 1, 1)$, the output of the network is given by $\vec{y}(t+1) = sgn[W.\vec{y}(t)]$:

$$y_1 = \begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$y_2 = sgn\left(\begin{bmatrix} 0.4 & 0 & -0.4 & 0 & -0.4 \\ 0 & 0.4 & 0 & -0.4 & 0 \\ -0.4 & 0 & 0.4 & 0 & 0.4 \\ 0 & -0.4 & 0 & 0.4 & 0 \\ -0.4 & 0 & 0.4 & 0 & 0.4 \end{bmatrix} . \begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$y_3 = sgn\left(\begin{bmatrix} 0.4 & 0 & -0.4 & 0 & -0.4 \\ 0 & 0.4 & 0 & -0.4 & 0 \\ -0.4 & 0 & 0.4 & 0 & 0.4 \\ 0 & -0.4 & 0 & 0.4 & 0 \\ -0.4 & 0 & 0.4 & 0 & 0.4 \end{bmatrix} . \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$y_4 = sgn\left(\begin{bmatrix} 0.4 & 0 & -0.4 & 0 & -0.4 \\ 0 & 0.4 & 0 & -0.4 & 0 \\ -0.4 & 0 & 0.4 & 0 & 0.4 \\ 0 & -0.4 & 0 & 0.4 & 0 \\ -0.4 & 0 & 0.4 & 0 & 0.4 \end{bmatrix} . \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$...$$

$$y_n = sgn\left(\begin{bmatrix} 0.4 & 0 & -0.4 & 0 & -0.4 \\ 0 & 0.4 & 0 & -0.4 & 0 \\ -0.4 & 0 & 0.4 & 0 & 0.4 \\ 0 & -0.4 & 0 & 0.4 & 0 \\ -0.4 & 0 & 0.4 & 0 & 0.4 \end{bmatrix} . \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Thus, $(-1, -1, 1, 1, 1)$ is the resulting fixed point. So the the output of the network when the dynamics start from $\xi = (1, -1, 1, 1, 1)$ is $\xi' = (-1, -1, 1, 1, 1)$.