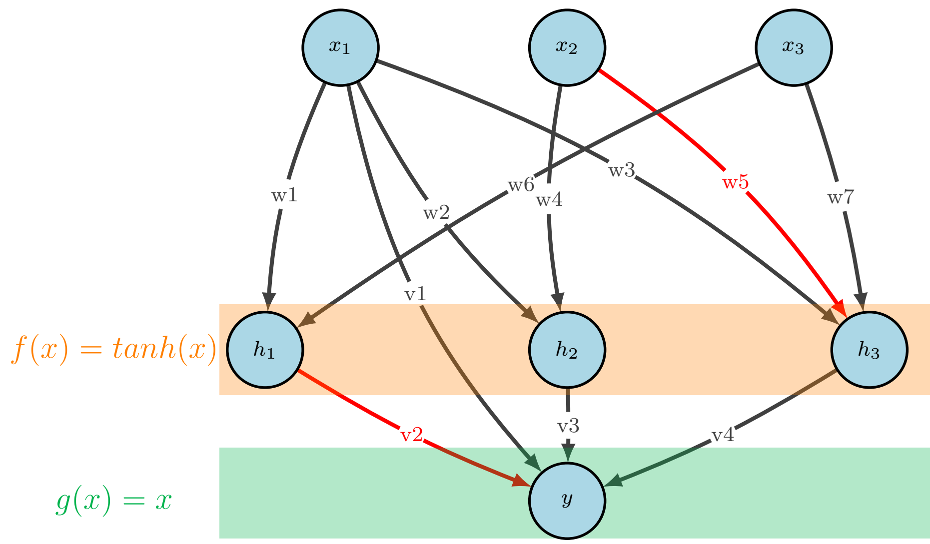
Yaowen Mei

Soft Computing ENGG6570 **Homework-1** 2021/05/28

**Question 1-1**.**Write the feedforward equations of h1, h2, h3 and y, as a function of the input variables x1, x2 and x3. Note that the thresholds must be included:**




**Solution 1-1:**

From the diagram above, we can write:

$$\boxed{h_1 = f\left(w_1 x_1 + w_6 x_3 + \theta_{h1}\right)} \tag{1}$$

$$\boxed{h_2 = f\left(w_2 x_1 + w_4 x_2 + \theta_{h2}\right)} \tag{2}$$

$$\boxed{h_3 = f\left(w_3 x_1 + w_5 x_2 + w_7 x_3 + \theta_{h3}\right)} \tag{3}$$

$$y = g\left(v_1 x_1 + v_2 h_1 + v_3 h_2 + v_4 h_3\right)$$

**Question 1-2**. **Derive a learning algorithm for v2, w5, using LMS (Least Mean Square) method, i.e., by minimizing the output error e = t - y, where t is the target output.**

**Solution 1-2:**

Given that $f(x) = tanh(x)$ and $g(x) = x$, we have $f'$ and $g'$:

$$f' = 1 - f^2$$

$$g' = 1$$

By observation from the diagram, $v_2$ is a directed Edge from Vertex $h_1$ to Vertex $y$; while, $w_5$ is a directed Edge from Vertex $x_2$ to $h_3$.

From the lecture, we can proofed that the LMS Error strategy leads to a uniform learning equation for both the hidden layer and the output layer:

$$\Delta Edge = \eta \times \underbrace{(ActivationFunction)'}_{\text{Activationfunction@Dist}} \times \underbrace{(error)}_{\text{error@Dist}} \times \underbrace{(input)}_{\text{value@Source}}$$

This is to say

$$\boxed{\Delta v_2 = \eta g' e_y h_1 = \Delta w_3 = \eta\,(t-y)\,h_1} \tag{4}$$

$$\boxed{\Delta w_5 = \eta f' e_{h3} x_2 = \Delta w_3 = \eta\left(1-f^2\right)e_{h3}x_2} \tag{5}$$

Where:

$$e_{h3} = g' \times e \times v_4 = g'\,(t-y)\,v_4 = (t-y)\,v_4$$

---

To get equation (4), and equation (5) in detail:

The LMS Error $E$ is defined as:

$$E = \frac{1}{2}(t-y)^2$$

It is trivial to get $\Delta v_2$

$$\Delta v_2 = -\eta\frac{\partial E}{\partial v_2} = -\eta\,(t-y)\left(\frac{\partial\,(t-y)}{\partial v_2}\right) = \eta\,(t-y)\left(\frac{\partial y}{\partial v_2}\right) = \eta\,(t-y)\,(h_1 g') = \eta\,(t-y)\,h_1 \quad \square$$

It is tricky to calculate $\Delta w_5$

$$\Delta w_5 = -\eta\frac{\partial E}{\partial w_5} = -\eta\,(t-y)\left(\frac{\partial\,(t-y)}{\partial w_5}\right) = \eta\,(t-y)\left(\frac{\partial y}{\partial w_5}\right)$$

Re-write $y$ so that $y$ explicitly contains $w_5$:

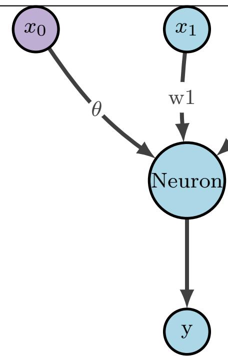
$$y = g\,(v_1x_1 + v_2h_1 + v_3h_2 + v_4h_3) = g\left[(v_1x_1 + v_2h_1 + v_3h_2) + v_4 f\,(w_3x_1 + w_5x_2 + w_7x_3 + \theta_{h3})\right]$$

$$\frac{\partial y}{\partial w_5} = g'\frac{\partial\left[(v_1x_1 + v_2h_1 + v_3h_2) + v_4 f\,(w_3x_1 + w_5x_2 + w_7x_3 + \theta_{h3})\right]}{\partial w_5} = g' v_4 f' x_2 = v_4 f' x_2$$

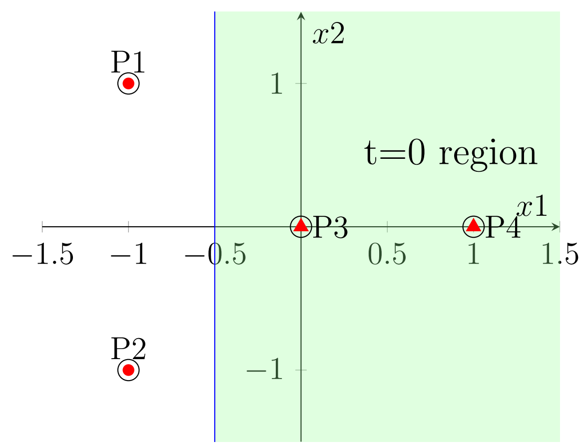
So, we can get:

$$\Delta w_5 = \eta\,(t-y)\left(\frac{\partial y}{\partial w_5}\right) = \eta\underbrace{(t-y)\,v_4}_{e_{h3}} f' x_2 = \eta e_{h3} f' x_2 = \eta\left(1-f^2\right)e_{h3}x_2 \quad \square$$

**Question 2-1. Design a single-neuron perceptron to solve this problem.**




| | **X1** | **X2** | **t** |
|---|---|---|---|
| **P1** | -1 | 1 | 1 |
| **P2** | -1 | -1 | 1 |
| **P3** | 0 | 0 | 0 |
| **P4** | 1 | 0 | 0 |




**Solution2-1:**

For hardlimitor activation function $y = h.l.(\mathbb{W}\mathbb{X} + \theta)$, the decision boundary is determined by:

$$\mathbb{W}\mathbb{X} + \theta = 0 \tag{6}$$

By observation, we are going to choose the line $x_1 = -0.5$ as our boundary. The weight vector that is orthogonal to our decision boundary is:

$$\mathbb{W} = \begin{bmatrix} w_1 & w_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \end{bmatrix}$$

We know this $x_1 = -0.5$ line passing point $(-0.5, 1)$, we can find $\theta$ by

$$\begin{bmatrix} -1 & 0 \end{bmatrix} \begin{bmatrix} -0.5 \\ 1 \end{bmatrix} + \theta = 0.5 + \theta = 0 \Rightarrow \theta = -0.5$$

So we can write our model as:

$$\boxed{y = h.l.\left( \begin{bmatrix} -1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - 0.5 \right)} \tag{7}$$

**Question-2-2: Test your solution with all four input vectors.**

**Solution-2-2:**

Use all 4 input points to test our model:

1. Test for P1:

$$y_1 = h.l.\left(\begin{bmatrix} -1 & 0 \end{bmatrix}\begin{bmatrix} -1 \\ 1 \end{bmatrix} - 0.5\right) = h.l.(1+1) = 1 \qquad ☑$$

2. Test for P2:

$$y_2 = h.l.\left(\begin{bmatrix} -1 & 0 \end{bmatrix}\begin{bmatrix} -1 \\ -1 \end{bmatrix} - 0.5\right) = h.l.(1+1) = 1 \qquad ☑$$

3. Test for P3:

$$y_3 = h.l.\left(\begin{bmatrix} -1 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \end{bmatrix} - 0.5\right) = h.l.(0-0.5) = 0 \qquad ☑$$

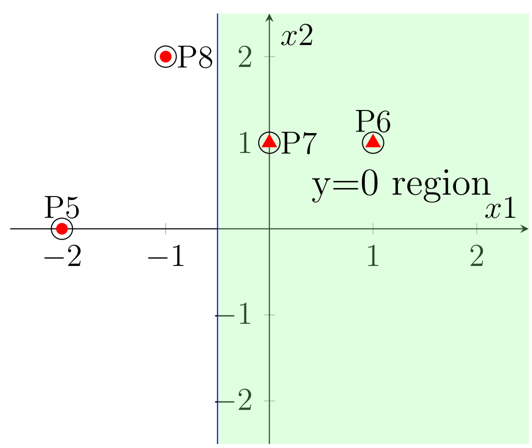
4. Test for P4:

$$y_4 = h.l.\left(\begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \end{bmatrix} - 0.5\right) = h.l.(0-0.5) = 0 \qquad ☑$$

**Question-2-3: Classify the following input vectors with your solution. You can either perform the calculations manually or with Matlab.**

**Solution-2-3:**

| | **X1** | **X2** | **Y** |
|---|---|---|---|
| **P5** | -2 | 0 | 1 |
| **P6** | 1 | 1 | 0 |
| **P7** | 0 | 1 | 0 |
| **P8** | -1 | 2 | 1 |

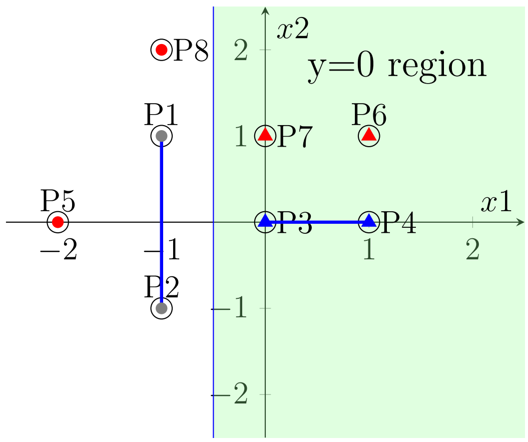



```
% Learning Model
w=[-1,0]
theta=-0.5

% Training DataSet P1-P4 for question2-1 to question2-2
pointset1=[-1,-1,0,1;1,-1,0,0]
result1 = hardlim(w*pointset1+theta)

% DataSet P5--P8 for question2-3
pointset2=[-2,1,0,-1;0,1,1,-2]
result2 = hardlim(w*pointset2+theta)
% result2 =
%      1      0      0      1
```

**Question-2-4: Which of the input vectors in Part (3) is always classified the same way, regardless of the solution values of the weigh $\mathbb{W}$ and the threshold $\theta$? Which may vary depending on the solution? Why?**




**Solution:** We can see from the plot below, no decision boundary shall across the two thick orange lines that respectively connect Class-1 (Point-1 and Point-2) and Class-2 (Point-3 and Point-4).

- Any data points inside of the gray zone (open boundary) are **not** always classified the same way.
- Any data points inside of the white zone (close boundary) are always classified the same way

| |
|---|
| Point 7 and Point 8 may vary depending on the decision boundary |

| |
|---|
| Point 5 and Point 6 can always be classified the same way |