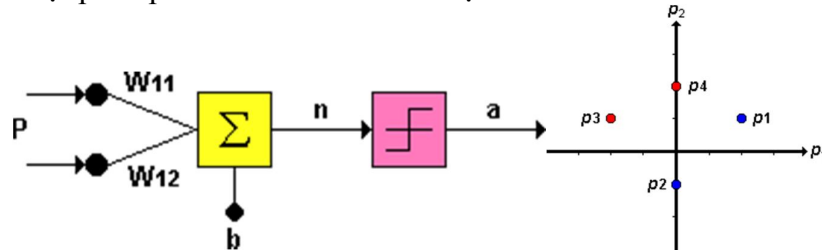


Sinh viên không được sử dụng tài liệu

Câu 1: (5,0đ)

Hãy khảo sát một perceptron hai đầu vào với một neuron như hình vẽ



$$P_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \quad t_1 = 0; \quad P_2 = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \quad t_2 = 1; \quad P_3 = \begin{bmatrix} -2 \\ 2 \end{bmatrix} \quad t_3 = 0; \quad P_4 = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad t_4 = 1$$

$$W(0) = [0 \ 0], \quad b(0) = 0;$$

Câu 2: (2,5đ)

Nêu cơ sở để lập thuật toán học cho mạng Perceptron và các bước lập trình mạng Perceptron

Câu 3: (2,5đ)

Hãy giải thích đoạn code sau, chức năng các câu lệnh trong matlab. Sử dụng mạng gì, khi muốn thay đổi số vòng dạy lên 1000 lần ta làm thế nào?

```
net = newlin([0,10],1); %?????
net.inputWeights{1,1}.delays = [0 1 2]; %?????
net.IW{1,1} = [7 8 9];
net.b{1} = [0]; %
pi = {1 2}; %
p = {3 4 5 6}; %
[a, pf] = sim(net,p,pi) %
a = [46] [70] [94] [118] %
pf = [5] [6] %
T = {10 20 30 40}; %
net.adaptParam.passes = 10; %
[net,y,E pf,af] = adapt(net,p,T,pi); %
wts = net.IW{1,1} %
bias = net.b{1} %
```

Chủ nhiệm bộ môn

Giảng viên ra đề thi

ĐÁP ÁN ĐỀ THI
MÔN ĐIỀU KHIỂN HỆ THỐNG 3

Ngày thi: 09/04/2011

Thời gian: 45 phút

Câu 1: (4,0 đ)	Điểm
<p>a=1 $a = \text{hardlim}(W(0)p_1 + b(0)) = \text{hardlim}([0 \ 0] \begin{bmatrix} 2 \\ 2 \end{bmatrix} + 0) = \text{hardlim}(0) = 1$</p>	
<p> $e = t_1 - a = 0 - 1 = -1$ $\Delta W = ep_1^T = (-1)[2 \ 2] = [-2 \ -2]$ $\Delta b = e = (-1) = -1$ $W^{new} = W^{old} + ep^T = [0 \ 0] + [-2 \ -2] = [-2 \ -2] = W(1)$ $b^{new} = b^{old} + e = 0 + (-1) = -1 = b(1)$ $a = \text{hardlim}(W(1)p_2 + b(1)) = \text{hardlim}([-2 \ -2] \begin{bmatrix} -2 \\ -2 \end{bmatrix} - 1) = \text{hardlim}(1) = 1$ $W(2) = W(1) = [-2 \ -2] \quad b(2) = b(1) = -1$ $W(4) = [-3 \ -1] \quad b(4) = 0$ $W(6) = [-2 \ -3] \quad b(6) = 1$ </p>	
Câu 2: (3,0 đ)	
<p>Cơ sở để ta lập thuật toán học cho mạng perceptron như sau: Ta có 3 trường hợp: Case 1: $a = t \Rightarrow e = 0$: w không thay đổi Case 2: $a = 0, t = 1 \Rightarrow e = t - a = 1$: $w = w + p$ vì lúc đó w sẽ gần p hơn và có khả năng cho kết quả $a = 1$ cao hơn Case 3: $a = 1, t = 0 \Rightarrow e = t - a = -1$: $w = w - p$ vì lúc đó w sẽ xa p hơn và có khả năng cho kết quả $a = 0$ cao hơn. Từ đó suy ra nguyên tắc học thống nhất:</p> $w^{new} = w^{old} + ep = w^{old} + (t - a)p$ $b^{new} = b^{old} + e$ <p>Các bước lập trình mạng Perceptron Bước 1 : Chọn tốc độ học Bước 2 : Khởi động - gán sai số $E=0$ - gán biến chạy $k=1$ - gán các vector trọng số $w_i(k)$ ($i=1, n$) bằng giá trị ngẫu nhiên nhỏ bất kì Bước 3 : Quá trình huấn luyện bắt đầu, tính : $y_i(k) = \text{step}(w_i^T(k)x(k)) = \text{step}\left(\sum_{j=1}^m w_{ij}(k)x_j(k)\right) \quad (i = \overline{1, n})$ Bước 4 : Cập nhật các vector trọng số $w_i(k+1) = w_i(k) + \eta(d_i(k) - y_i(k))x(k) \quad (i = \overline{1, n})$</p>	

<p>Bước 5 : Tính sai số tích lũy</p> $E = E + \frac{1}{2} \ d(k) - y(k)\ ^2$ <p>Bước 6 : Nếu $k < K$ thì gán $k=k+1$ và trở lại Bước 3. Nếu $k=K$ thì tiếp tục Bước 7</p> <p>Bước 7 : Kết thúc một chu kì huấn luyện (epoch).</p> <p>Nếu $E=0$ thì kết thúc quá trình học.</p> <p>Nếu $E \neq 0$ thì gán $E=0$, $k=1$ và trở lại Bước 3 bắt đầu một chu kì huấn luyện mới.</p>	
<p>Câu 3: (3,0 đ)</p>	
<pre> net = newlin([0,10],1); % Khai bao mang net.inputWeights{1,1}.delays = [0 1 2];%Khai bao Tapped Delay net.IW{1,1} = [7 8 9]; net.b{1} = [0]; % Khai bao vector trong so va bias pi = {1 2}; % Gia tri ban dau cho bo delay p = {3 4 5 6}; % Input cho bo Filter theo thoi gian [a, pf] = sim(net,p,pi) %output cua mang theo thong so ban dau a = [46] [70] [94] [118] pf = [5] [6] T = {10 20 30 40}; %output mong muon net.adaptParam.passes = 10; %Chon so vong day la` 10 [net,y,E pf,af] = adapt(net,p,T,pi); %Huan luyen neural voi bo I/O wts = net.IW{1,1} %vector trong so sau huan luyen bias = net.b{1} %bias sau huan luyen </pre>	
<p>Mạng Adaline bộ lọc tuyến tính thích nghi</p>	
<pre> net.adaptParam.passes = 1000; </pre>	