



Chapter 3

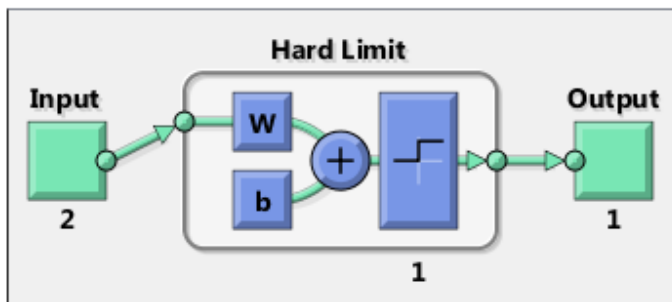
perceptron algorithm

(1)

Example: Use a perceptron to solve a simple classification logical-OR problem.

Sol:

```
x = [0 0 1 1; 0 1 0 1];  
t = [0 1 1 1];  
net = perceptron;  
net = train(net,x,t);  
view(net)  
y = net(x);
```





perceptron algorithm

(2)

Rosenblatt's Perceptron

Training Phase :

```
function [w,b,pass]=PerceptronTrn(x,y);
% %Rosenblatt's Perceptron
tic
[l,p]=size(x);
w=zeros(p,1); % initialize weights
b=0;          % initialize bias
ier=1;        % initialize a misclassification indicator
pass=0;       % number of iterations
n=0.5;        % learning rate
r=max(sqrt(sum(x))); % max norm
while ier==1, %repeat until no error
    ier=0;
    e=0; % number of training errors
    for i=1:l % a pass through x
        xx=x(i,:);
        ey=xx*w+b; % estimated y
        if ey>=0.5
            ey=1;
        else
            ey=0;
        end
        if y(i)~=ey;
            er=y(i)-ey; % error difference
            w=w'+(er*n)*x(i,:); % the only rule works for
me
            %w=w'+(n*y(i))*(x(i,:)); % don't know why
itdoes npt work
            %b=b-n*y(i)*(r^2); % don't know why
itdoes npt work
            e=e+1 ; % number of training errors
            w=w';
        end;
    end;
    ee=e; % number of training errors
```



```
-----  
    if ee>0 % cuntinue if there is still errors  
        ier=1;  
    end  
    pass=pass+1; % stop after 10000 iterations  
    if pass==10000  
        ier=0;  
        pass=0;  
    end;  
end;  
disp(['Training_Errors=' num2str(e) ' Training data  
Size=' num2str(l)])  
toc
```

Testing Phase :

```
function e=PerecptronTst(x,y,w,b);  
%=====   
% Testing phase  
%=====   
tic  
[l,p]=size(x);  
e=0; % number of test errors  
for i=1:l  
    xx=x(i,:); % take one row  
    ey=xx*w+b; % apply the perceptron classification rule  
    if ey>=0.5  
        ey=1;  
    else  
        ey=0;  
    end  
    if y(i)~=ey;  
        e=e+1;  
    end;  
end  
toc
```