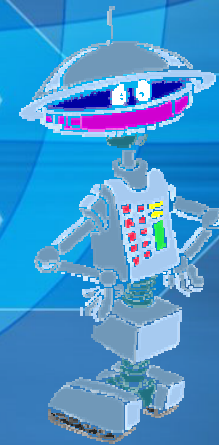
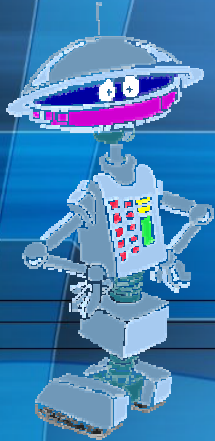


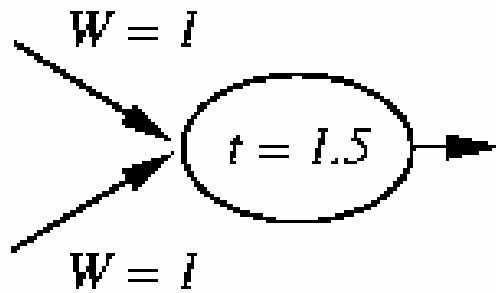
Single Perceptron



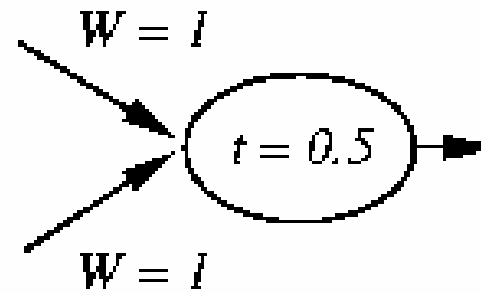
Machine Learning Team
PENS - ITS 2006



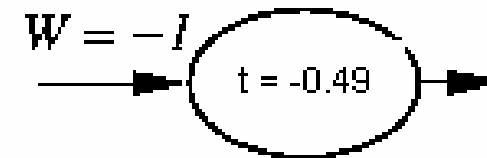
Jaringan sederhana



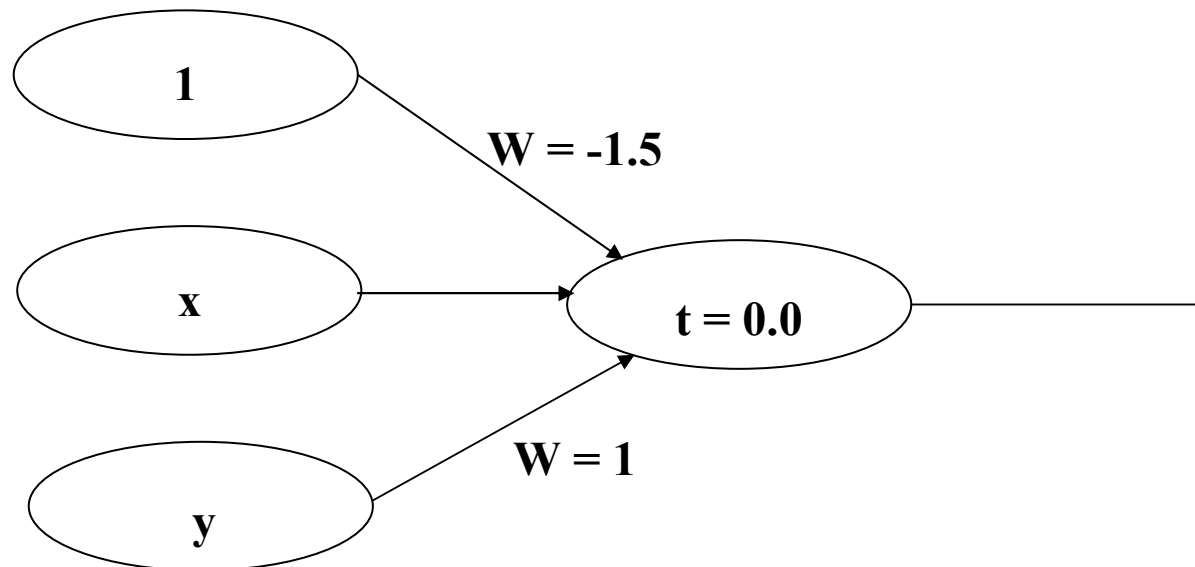
AND



OR

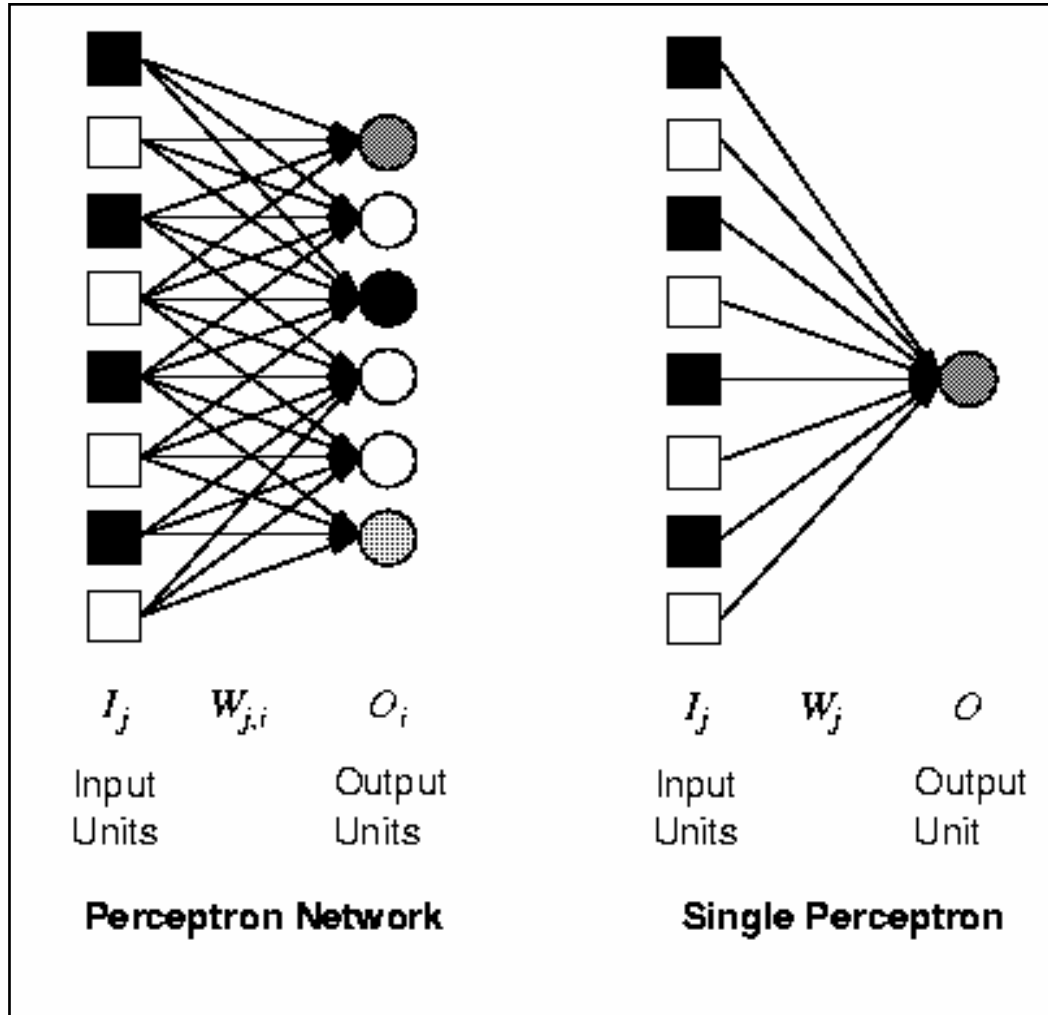


NOT





Perceptron



$$O = Step_0 \sum_j W_j I_j$$



Dapatkah perceptron dipresentasikan?

Input 1

Input 2

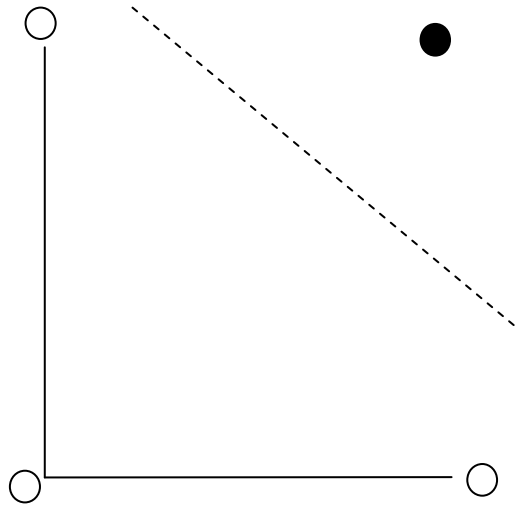
Output

| AND | | | |
|-----|---|---|---|
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 |

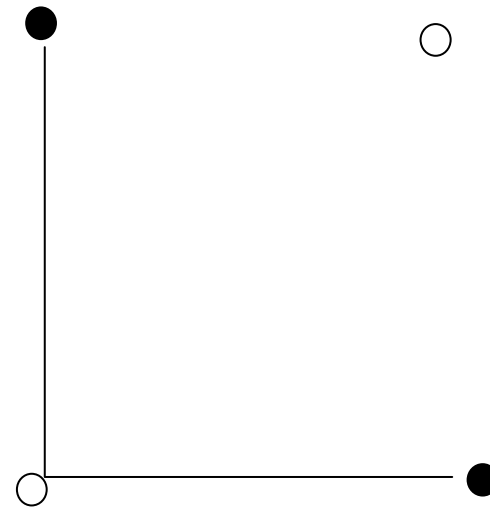
| XOR | | | |
|-----|---|---|---|
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |



Dapatkah perceptron dipresentasikan?



AND

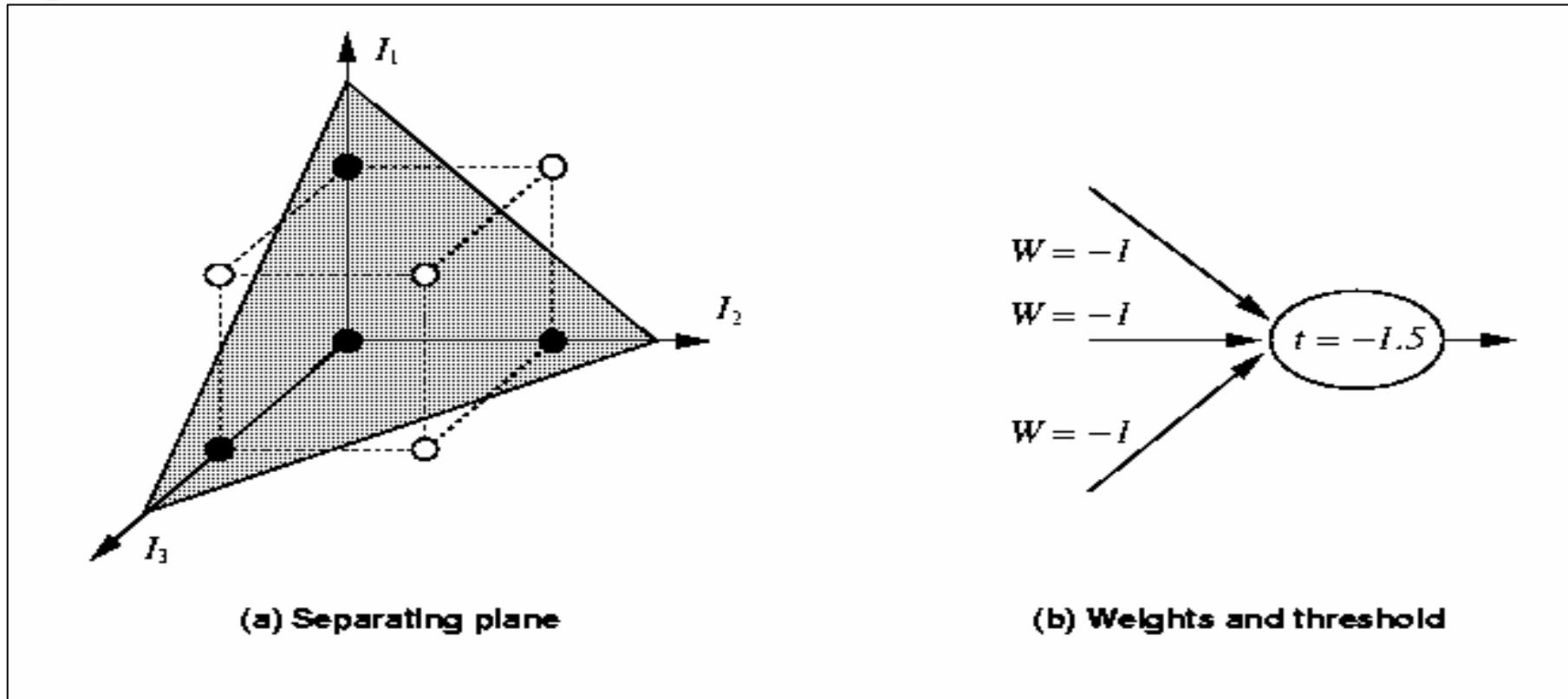


XOR

- Fungsi yang dapat dipisahkan disebut *Linearly Separable*
- Hanya fungsi *Linearly Separable* dapat dipisahkan dengan perceptron



Dapatkah perceptron dipresentasikan?



Fungsi *Linearly Separable* dapat digunakan untuk lebih dari 3 dimensi – tapi sangat sulit digambarkan



Training sebuah perceptron

Input 1

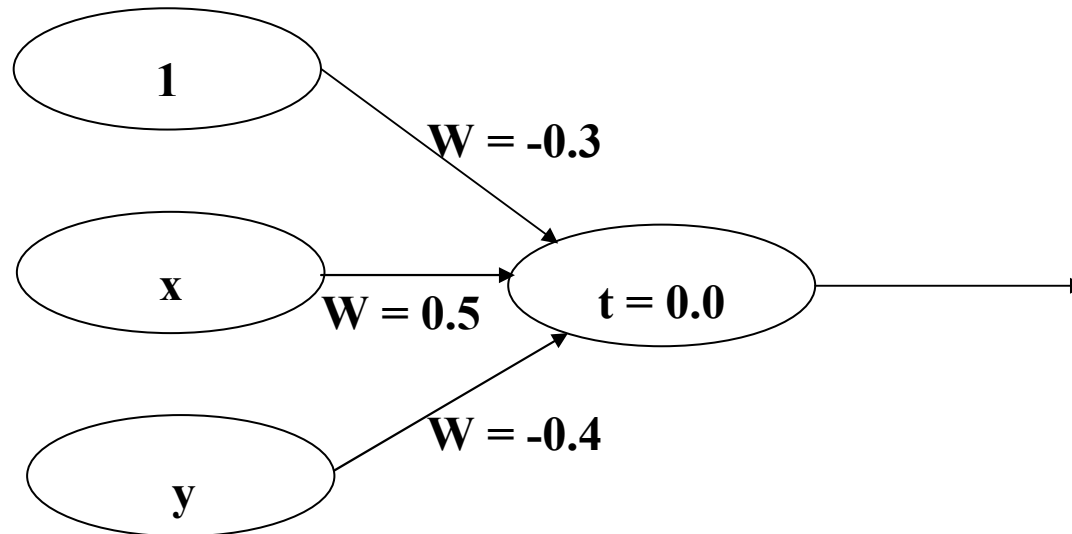
Input 2

Output

| AND | | | |
|-----|---|---|---|
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 |



Training sebuah perceptron



| I_1 | I_2 | I_3 | Penjumlahan | Output |
|-------|-------|-------|--|--------|
| 0 | 0 | 1 | $(0*0.5) + (0*-0.4) + (1*-0.3) = -0.3$ | 0 |
| 0 | 1 | 1 | $(0*0.5) + (1*-0.4) + (1*-0.3) = -0.7$ | 0 |
| 1 | 0 | 1 | $(1*0.5) + (0*-0.4) + (1*-0.3) = 0.2$ | 1 |
| 1 | 1 | 1 | $(1*0.5) + (1*-0.4) + (1*-0.3) = -0.2$ | 0 |



Pembelajaran

While(epoch menghasilkan error)

.....

$$\text{Err} = T - O$$

If Err \neq 0 then

$$W_j = W_j + LR * I_j * \text{Err}$$

End If

End



Pembelajaran

While(epoch menghasilkan error)

...

Err = T – O

If Err \neq 0 then

$W_j = W_j + LR * I_j * Err$

End If

End

Epoch :Set training pada jaringan

KasusAND epoch berisi 4 set dari input

dipresentasikan pada jaringan([0,0], [0,1], [1,0], [1,1])



Pembelajaran

While(epoch menghasilkan error)

...

Err = T – O

If Err <> 0 then

$W_j = W_j + LR * I_j * Err$

End If

End

Nilai Training, T : Training sebuah jaringan

Kita tidak hanya memperhatikan input tetapi juga nilai yang dihasilkan jaringan. Contoh jika kita mempresentasikan jaringan dengan [1,1] untuk fungsi AND nilai training 1



Pembelajaran

While(epoch menghasilkan error)

```
...  
Err = T - O  
If Err <> 0 then  
     $W_j = W_j + LR * I_j * Err$   
End If
```

End

Error, Err : Nilai error

adalah perbedaan nilai output dari jaringan dengan nilai training. Contoh jika kita membutuhkan jaringan untuk training 0 dan output 1, maka $Err = -1$



Pembelajaran

While(epoch menghasilkan error)

...

Err = T – O

If Err <> 0 then

$W_j = W_j + LR * I_j * Err$

End If

End

Output dari Neuron, O :

I_j : Input dari neuron

W_j : Bobot dari input neuron (I_j) ke output neuron

LR : Learning Rate.

Bagaimana membuat jaringan konvergen?

set dari eksperimen biasanya (0.1)

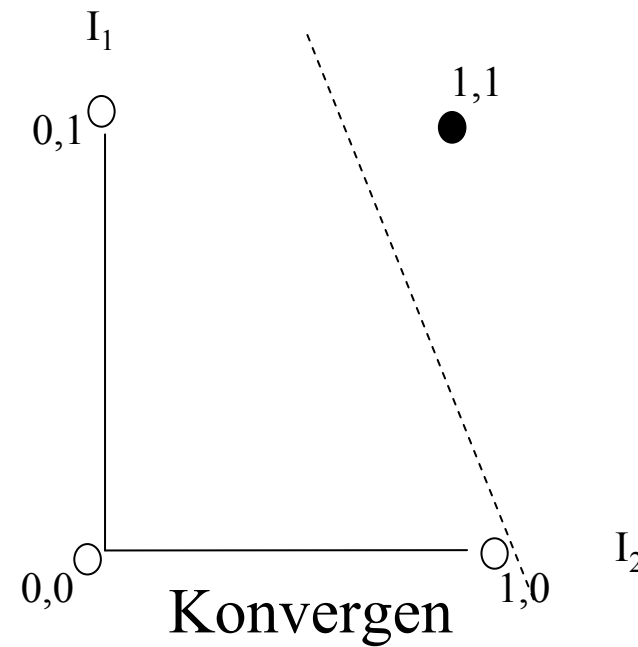
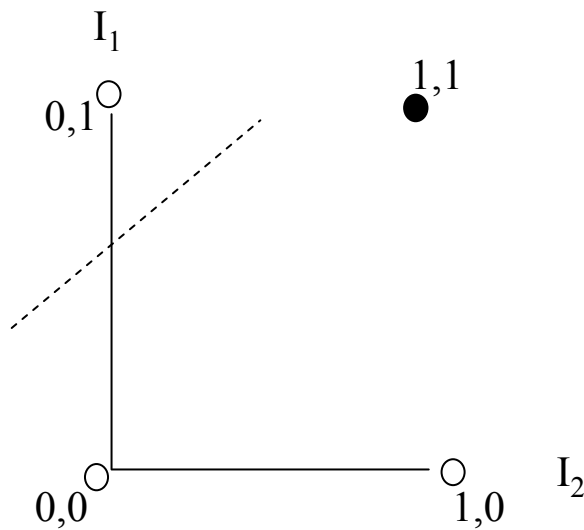


Pembelajaran

Catatan

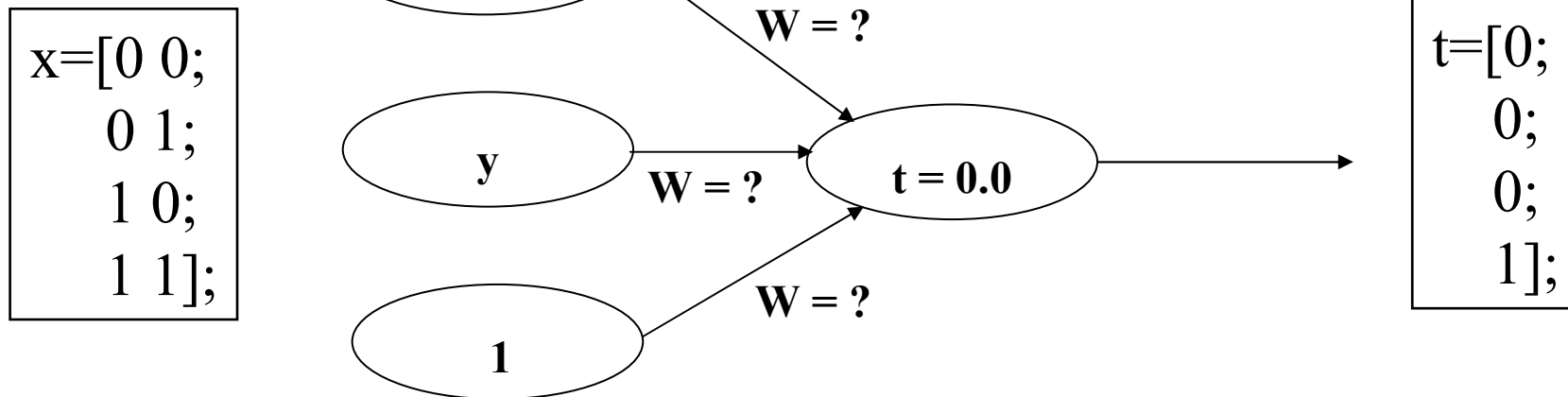
$$I_1 \text{ point} = W_0/W_1$$

$$I_2 \text{ point} = W_0/W_2$$





Input dan target AND





Mencari bobot terbaik?

```
%inisialisasi bobot
w=[];
for j=1:jumlahinput+1
    for i=1:jumlahtarget
        w(j,i)=0.15*(2*rand-1);
    end
end
```

```
%pembelajaran untuk mendapatkan bobot terbaik
for itr=1:jumlahiterasi
    for p=1:jumlahpattern
        for i=1:jumlahtarget
            O(i)=input(p,:)*w(:,i);
            if(O(i)>0.0)
                out(i)=1;
            else
                out(i)=0;
            end
            ERR(i)=target(p,i)-out(i);
            if(ERR(i)~=0)
                for j=1:jumlahinput+1
                    w(j,i)=w(j,i)+LR*input(p,j)*ERR(i);
                end
            end
        end
    end
end
end
```

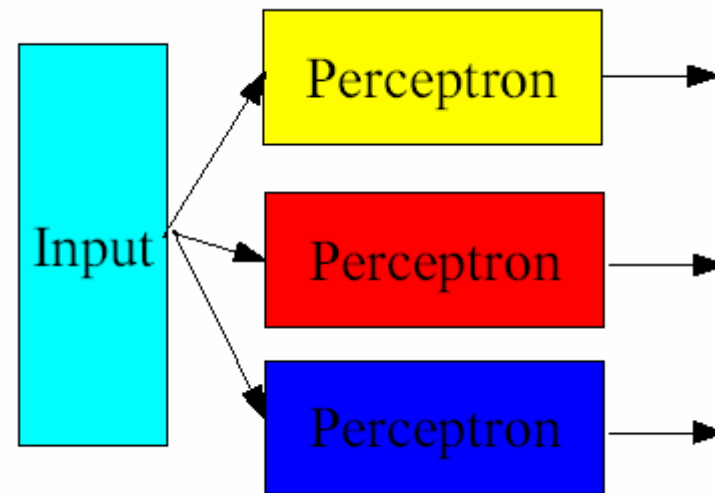
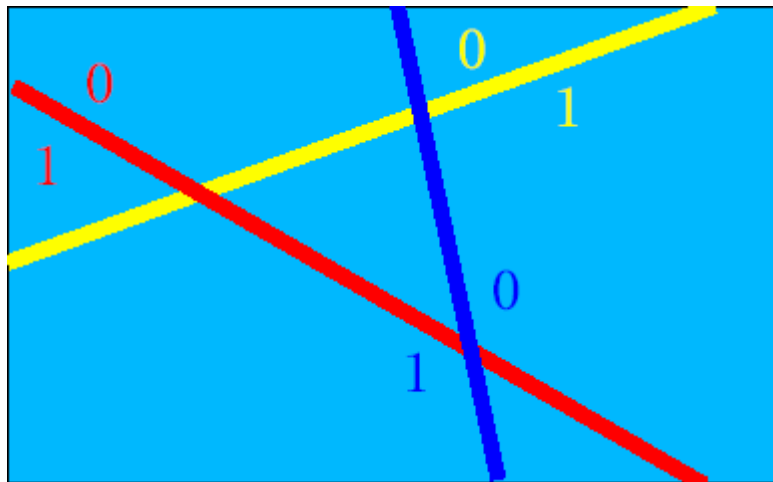



Menghitung keluaran jaringan

```
function output=RunNN(input, w, target)
%menghitung keluaran jaringan
jumlahinput=size(input,2);
jumlahtarget=size(target,2);
input(jumlahinput+1)=1;
output=zeros(jumlahtarget,1);
for i=1:jumlahtarget
    out=input*w(:,i);
    if(out>0.0)
        output(i)=1;
    else
        output(i)=0;
    end
end
end
```



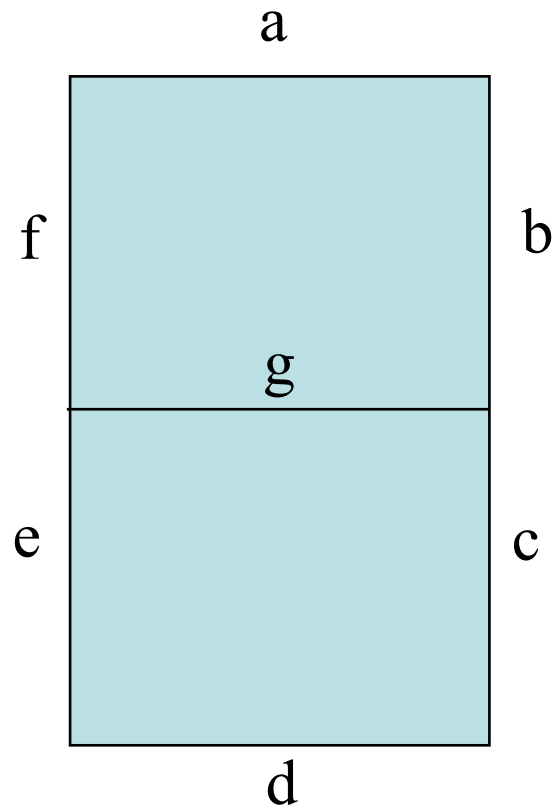
Multi Output Perceptron



Menggunakan single perceptron untuk membagi data input dalam beberapa katagori



Input dan target seven segment



| | a | b | c | d | e | f | g | |
|----|---|---|---|---|---|---|---|---|
| x= | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 2 |
| | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 3 |
| | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 4 |
| | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 5 |
| | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 6 |
| | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 7 |
| | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 8 |
| | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 9 |



Target dari seven segment

0 1 2 3 4 5 6 7 8 9

```
t=[1 0 0 0 0 0 0 0 0 0;  
  0 1 0 0 0 0 0 0 0 0;  
  0 0 1 0 0 0 0 0 0 0;  
  0 0 0 1 0 0 0 0 0 0;  
  0 0 0 0 1 0 0 0 0 0;  
  0 0 0 0 0 1 0 0 0 0;  
  0 0 0 0 0 0 1 0 0 0;  
  0 0 0 0 0 0 0 1 0 0;  
  0 0 0 0 0 0 0 0 1 0;  
  0 0 0 0 0 0 0 0 0 1];
```