



# MODEL JARINGAN PERCEPTRON

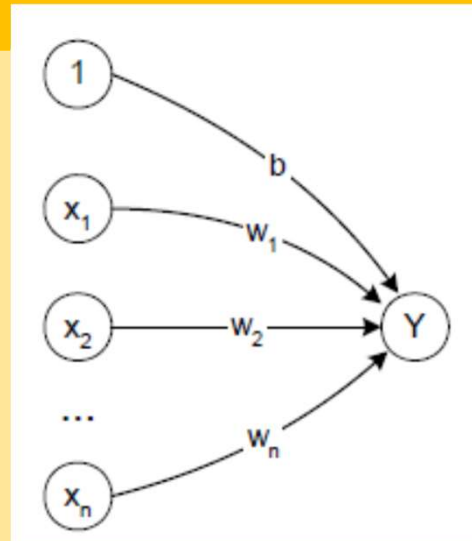


# PERCEPTRON

- Model jaringan perceptron ditemukan oleh Rosenblatt (1962) dan Minsky – Papert (1969).
- Model tersebut merupakan model yang memiliki aplikasi dan pelatihan yang paling baik pada era tersebut.
- Metode pembelajaran Perceptron lebih kuat dari pembelajaran JST Hebb.
- Prosedur pelatihan dari Perceptron terutama dalam proses iterasinya dapat menghasilkan bobot yang *konvergen*. Dan bobot ini bila ditest terhadap pola input dapat menghasilkan output yang sesuai target.



# Arsitektur Perceptron



- Arsitektur jaringan perceptron mirip dengan arsitektur jaringan Hebb.
- Jaringan terdiri dari beberapa unit masukan (ditambah sebuah bias), dan memiliki sebuah unit keluaran. Hanya saja fungsi aktivasi bukan merupakan fungsi biner (atau bipolar), tetapi memiliki kemungkinan nilai -1, 0 atau 1.



# Fungsi Aktivasi

- Untuk suatu harga *threshold*  $\theta$  yang ditentukan :

$$f(\text{net}) = \begin{cases} 1 & \text{jika } \text{net} > \theta \\ 0 & \text{jika } -\theta \leq \text{net} \leq \theta \\ -1 & \text{jika } \text{net} < -\theta \end{cases}$$

- Fungsi aktivasi yg dipakai akan mengatur batas antara daerah positif, negatif dan nol.
- Daerah nol adalah daerah yg terletak antara daerah positif dan negatif. Lebar daerah ini adalah  $\theta$
- Garis pemisah antara daerah positif dan daerah nol memiliki pertidaksamaan :

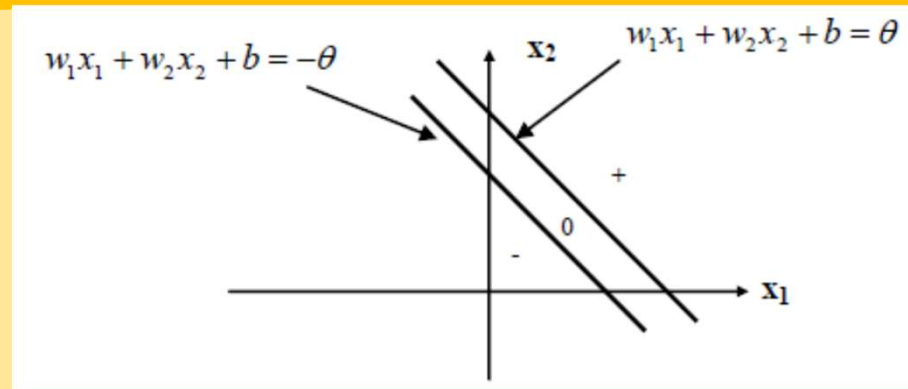
$$x_1 w_1 + \dots + x_n w_n + b = \theta$$

- Garis pemisah antara daerah negatif dan daerah nol memiliki pertidaksamaan :

$$x_1 w_1 + \dots + x_n w_n + b = -\theta$$



# Fungsi Aktivasi



Batas Keputusan Perceptron

- Shg daerah positif dinyatakan dgn pertidaksamaan :
$$w_1x_1 + w_2x_2 + b > \theta$$
- Dan daerah negatif dinyatakan dgn pertidaksamaan :
$$w_1x_1 + w_2x_2 + b < -\theta$$



# Algoritma Pelatihan Perceptron

- Misalkan :
  - s vektor masukan dan t adalah target keluaran
  - $\alpha$  adalah laju pembelajaran/pemahaman (learning rate) yang ditentukan
  - $\theta$  adalah threshold yang ditentukan
- Maka algoritma perceptronnya adalah sebagai berikut



0. Inisialisasi semua bobot dan bias (umumnya  $w_i = b = 0$  )

Set laju pembelajaran  $\alpha$  ( $0 < \alpha \leq 1$ ) (untuk penyederhanaan set  $\alpha = 1$ )

1. Selama kondisi berhenti bernilai FALSE atau selama ada elemen vektor masukan yang respon unit keluarannya tidak sama dengan target ( $y \neq t$ ), lakukan langkah-langkah 2 – 6

2. Untuk setiap pasangan (s,t), kerjakan langkah 3 – 5:

3. Set aktivasi unit masukan  $x_i = s_i$  ( $i = 1, \dots, n$ )

4. Hitung respon untuk unit output:

$$\text{net} = \sum_i x_i w_i + b$$

$$f(\text{net}) = \begin{cases} 1 & \text{jika } \text{net} > \theta \\ 0 & \text{jika } -\theta \leq \text{net} \leq \theta \\ -1 & \text{jika } \text{net} < -\theta \end{cases}$$

5. Perbaiki bobot dan bias pola jika terjadi kesalahan

Jika  $y \neq t$ , maka:

$$w_i (\text{baru}) = w_i (\text{lama}) + \Delta w_i \quad (i = 1 \dots n); \text{ dengan } \Delta w_i = \alpha t x_i$$

$$b (\text{baru}) = b (\text{lama}) + \Delta b \text{ dengan } \Delta b = \alpha t$$

Jika  $y = t$  maka:

$$w_i (\text{baru}) = w_i (\text{lama})$$

$$b (\text{baru}) = b (\text{lama})$$

6. Test kondisi berhenti: jika tidak terjadi perubahan bobot pada (i) maka kondisi berhenti TRUE, namun jika masih terjadi perubahan maka kondisi berhenti FALSE



Ada beberapa hal yang perlu diperhatikan dalam algoritma tersebut:

- a. Iterasi terus dilakukan hingga semua pola memiliki keluaran jaringan yang sama dengan targetnya (jaringan sudah memahami pola). Iterasi TIDAK berhenti setelah semua pola dimasukan seperti yang terjadi pada model Hebbian
- b. Pada langkah 5, perubahan bobot hanya dilakukan pada pola yang mengandung kesalahan (keluaran jaringan  $\neq$  target). Perubahan tersebut merupakan hasil kali unit masukan dengan target dan laju pembelajaran. Perubahan bobot hanya akan terjadi jika unit masukan  $\neq 0$
- c. Kecepatan iterasi ditentukan oleh laju pembelajaran  $\alpha$  ( dengan  $0 < \alpha \leq 1$ ) yang dipakai. Semakin besar  $\alpha$ , semakin sedikit iterasi yang diperlukan. Akan tetapi jika  $\alpha$  terlalu besar, maka akan merusak pola yang sudah benar sehingga pembelajaran menjadi lambat





# Algoritma pelatihan perseptron lebih baik dibandingkan model Hebbian karena:

- a. Setiap kali pola dimasukkan, hasil keluaran jaringan dibandingkan dengan target yang sesungguhnya. Jika terdapat perbedaan, maka bobot akan dimodifikasi. Jadi tidak semua bobot selalu dimodifikasi dalam setiap iterasinya
- b. Modifikasi bobot tidak hanya ditentukan oleh perkalian antara target dan masukan, tetapi juga melibatkan suatu laju pembelajaran yang besar dapat diatur.
- c. Pelatihan dilakukan berulang-ulang untuk semua kemungkinan pola yang ada hingga jaringan dapat mengerti polanya (ditandai dengan samanya semua keluaran jaringan dengan target keluaran yang diinginkan). Satu siklus pelatihan yang melibatkan semua pola disebut **Epoch**. Dalam jaringan Hebbian, pelatihan hanya dilakukan dalam satu Epoch saja.
- d. Teorema konvergensi perseptron menyatakan bahwa apabila ada bobot yang tepat, maka proses pelatihan akan konvergen ke bobot yang tepat tersebut



## Contoh 1:

- Buatlah perseptron untuk mengenali fungsi logika AND dengan masukan dan keluaran bipolar. Untuk inisialisasi, gunakan bobot dan bias awal = 0,  $\alpha = 1$ , dan threshold  $\theta = 0$

### Penyelesaian

- Tabel masukan dan target fungsi logika AND dengan masukan dan keluaran bipolar

Masukan			Target
$x_1$	$x_2$	1	$t$
1	1	1	1
1	-1	1	-1
-1	1	1	-1
-1	-1	1	-1



- Untuk threshold  $\theta = 0$ , maka fungsi aktivasinya:

$$f(net) = \begin{cases} 1 & \text{jika } net > 0 \\ 0 & \text{jika } net = 0 \\ -1 & \text{jika } net < 0 \end{cases}$$

- Iterasi untuk seluruh pola yang ada disebut EPOCH. Tabel berikut menunjukkan hasil pada **epoch pertama**

Masukan	target			Perubahan bobot	Bobot Baru
( $x_1$ $x_2$ 1)	t	net	y = f(net)	( $\Delta w_1$ $\Delta w_2$ $\Delta b$ )	( $w_1$ $w_2$ b)
Inisialisasi					(0 0 0)
(1 1 1)	1	0	0	(1 1 1)	(1 1 1)
(1 -1 1)	-1	1	1	(-1 1 -1)	(0 2 0)
(-1 1 1)	-1	2	1	(1 -1 -1)	(1 1 -1)
(-1 -1 1)	-1	-3	-1	(0 0 0)	(1 1 -1)

$$\Delta w = \alpha \cdot t \cdot xi$$

$$\Delta b = \alpha \cdot t$$



- Garis pemisah pola terbentuk dari persamaan:  
 $x_1w_1 + x_2w_2 + b = 0$  (karena  $\theta = 0$ ), jadi terbentuk sebuah garis saja
- Persamaan garis pemisah untuk setiap pola hasil iterasi:

Masukan	Bobot baru	Persamaan garis
$(x_1 \ x_2 \ 1)$	$(w_1 \ w_2 \ b)$	
$(1 \ 1 \ 1)$	$(1 \ 1 \ 1)$	$x_1 + x_2 = -1$
$(1 \ -1 \ 1)$	$(0 \ 2 \ 0)$	$x_2 = 0$
$(-1 \ 1 \ 1)$	$(1 \ 1 \ -1)$	$x_1 + x_2 = 1$
$(-1 \ -1 \ 1)$	$(1 \ 1 \ -1)$	$x_1 + x_2 = 1$

- Mengingat tidak semua  $f(\text{net})$  pada tabel di atas sama dengan  $t$ , maka iterasi dilanjutkan pada **epoch kedua**. Semua pola kembali dimasukkan ke jaringan dengan menggunakan bobot terakhir yang diperoleh sebagai bobot awalnya.



- Diperoleh hasil iterasi pada tabel di bawah berikut:

Masukan	target			Perubahan bobot	Bobot Baru
$(x_1 \ x_2 \ 1)$	$t$	net	$y = f(\text{net})$	$(\Delta w_1 \ \Delta w_2 \ \Delta b)$	$(w_1 \ w_2 \ b)$
Bobot yang diperoleh dari epoch pertama					$(1 \ 1 \ -1)$
$(1 \ 1 \ 1)$	1	1	1	$(0 \ 0 \ 0)$	$(1 \ 1 \ -1)$
$(1 \ -1 \ 1)$	-1	-1	-1	$(0 \ 0 \ 0)$	$(1 \ 1 \ -1)$
$(-1 \ 1 \ 1)$	-1	-1	-1	$(0 \ 0 \ 0)$	$(1 \ 1 \ -1)$
$(-1 \ -1 \ 1)$	-1	-3	-1	$(0 \ 0 \ 0)$	$(1 \ 1 \ -1)$

- Dalam iterasi di atas, untuk semua pola nilai  $f(\text{net}) = t$ , sehingga tidak dilakukan perubahan terhadap bobot. Karena  $f(\text{net}) = t$  untuk semua pola maka jaringan sudah mengenal semua pola sehingga iterasi dihentikan



Ulangi contoh di atas, tapi menggunakan masukan biner dan keluaran bipolar. Gunakan  $\alpha = 1$  dan threshold  $\theta = 0,2$ .

Penyelesaian:

Dengan *threshold* = 0.2, maka fungsi aktivasi menjadi :

$$y = f(\text{net}) = \begin{cases} 1 & \text{jika } \text{net} > 0.2 \\ 0 & \text{jika } -0.2 \leq \text{net} \leq 0.2 \\ -1 & \text{jika } \text{net} < -0.2 \end{cases}$$

Hasil iterasi epoch pertama

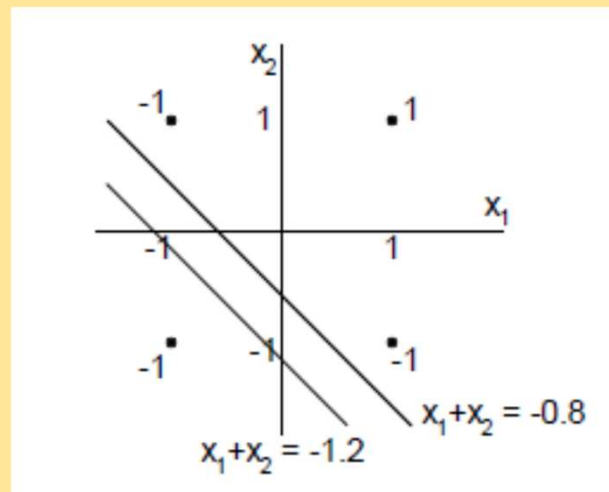
Masukan ( $x_1 \ x_2 \ 1$ )	Target $t$	$y$ net $= f(\text{net})$	Perubahan Bobot ( $\Delta w_1 \ \Delta w_2 \ \Delta b$ )	Bobot Baru ( $w_1 \ w_2 \ b$ )
inisialisasi				(0 0 0)
(1 1 1)	1	0	(1 1 1)	(1 1 1)
(1 0 1)	-1	2	(-1 0 -1)	(0 1 0)
(0 1 1)	-1	1	(0 -1 -1)	(0 0 -1)
(0 0 1)	-1	-1	(0 0 0)	(0 0 -1)



- Perhatikan bahwa dengan *threshold* yang  $\neq 0$ , maka akan terbentuk 2 buah garis pemisah pola. Dalam contoh ini, persamaan garis yang terbentuk adalah  $w_1x_1 + w_2x_2 + b = 0.2$  dan
- Sebagai contoh, setelah pola pertama dimasukkan, maka garis yang terbentuk memiliki persamaan

$$x_1 + x_2 + 1 = 0.2 \quad (x_1 + x_2 = -0.8)$$

$$x_1 + x_2 + 1 = -0.2 \quad (x_1 + x_2 = -1.2)$$





Masukan ( $x_1$ $x_2$ 1)	Target t	net = f (net)	y	Perubahan Bobot ( $\Delta w_1$ $\Delta w_2$ $\Delta b$ )			Bobot Baru ( $w_1$ $w_2$ b)		
epoch 2 :		Bobot dari epoch - 1					(0 0 -1)		
(1 1 1)	1	-1	-1	(1	1	1)	(1	1	0)
(1 0 1)	-1	1	1	(-1	0	-1)	(0	1	-1)
(0 1 1)	-1	0	0	(0	-1	-1)	(0	0	-2)
(0 0 1)	-1	-2	-1	(0	0	0)	(0	0	-2)
epoch 3 :									
(1 1 1)	1	-2	-1	(1	1	1)	(1	1	-1)
(1 0 1)	-1	0	0	(-1	0	-1)	(0	1	-2)
(0 1 1)	-1	-1	-1	(0	0	0)	(0	1	-2)
(0 0 1)	-1	-2	-1	(0	0	0)	(0	1	-2)
epoch 4 :									
(1 1 1)	1	-1	-1	(1	1	1)	(1	2	-1)
(1 0 1)	-1	0	0	(-1	0	-1)	(0	2	-2)
(0 1 1)	-1	0	0	(0	-1	-1)	(0	1	-3)
(0 0 1)	-1	-3	-1	(0	0	0)	(0	1	-3)





### epoch 5 :

(1 1 1)	1	-2	-1	(1 1 1)	(1 2 -2)
(1 0 1)	-1	-1	-1	(0 0 0)	(1 2 -2)
(0 1 1)	-1	0	0	(0 -1 -1)	(1 1 -3)
(0 0 1)	-1	-3	-1	(0 0 0)	(1 1 -3)

### epoch 6 :

(1 1 1)	1	-1	-1	(1 1 1)	(2 2 -2)
(1 0 1)	-1	0	0	(-1 0 -1)	(1 2 -3)
(0 1 1)	-1	-1	-1	(0 0 0)	(1 2 -3)
(0 0 1)	-1	-3	-1	(0 0 0)	(1 2 -3)

### epoch 7 :

(1 1 1)	1	0	0	(1 1 1)	(2 3 -2)
(1 0 1)	-1	0	0	(-1 0 -1)	(1 3 -3)
(0 1 1)	-1	0	0	(0 -1 -1)	(1 2 -4)
(0 0 1)	-1	-4	-1	(0 0 0)	(1 2 -4)



**epoch 8 :**

(1 1 1)	1	-1	-1	(1 1 1)	(2 3 -3)
(1 0 1)	-1	-1	-1	(0 0 0)	(2 3 -3)
(0 1 1)	-1	0	0	(0 -1 -1)	(2 2 -4)
(0 0 1)	-1	-4	-1	(0 0 0)	(2 2 -4)

**epoch 9 :**

(1 1 1)	1	0	0	(1 1 1)	(3 3 -3)
(1 0 1)	-1	0	0	(-1 0 -1)	(2 3 -4)
(0 1 1)	-1	-1	-1	(0 0 0)	(2 3 -4)
(0 0 1)	-1	-4	-1	(0 0 0)	(2 3 -4)

**epoch 10 :**

(1 1 1)	1	1	1	(0 0 0)	(2 3 -4)
(1 0 1)	-1	-2	-1	(0 0 0)	(2 3 -4)
(0 1 1)	-1	-1	-1	(0 0 0)	(2 3 -4)
(0 0 1)	-1	-4	-1	(0 0 0)	(2 3 -4)



- Setelah 10 epoch, semua  $f(\text{net}) = t$  sehingga jaringan telah mengenal pola dan iterasi dihentikan.
- Persamaan garisnya adalah :  
$$2x_1 + 3x_2 - 4 = 0.2 \text{ (atau } 2x_1 + 3x_2 = 4.2\text{)}$$
$$\text{dan } 2x_1 + 3x_2 - 4 = -0.2 \text{ (atau } 2x_1 + 3x_2 = 3.8\text{)}$$



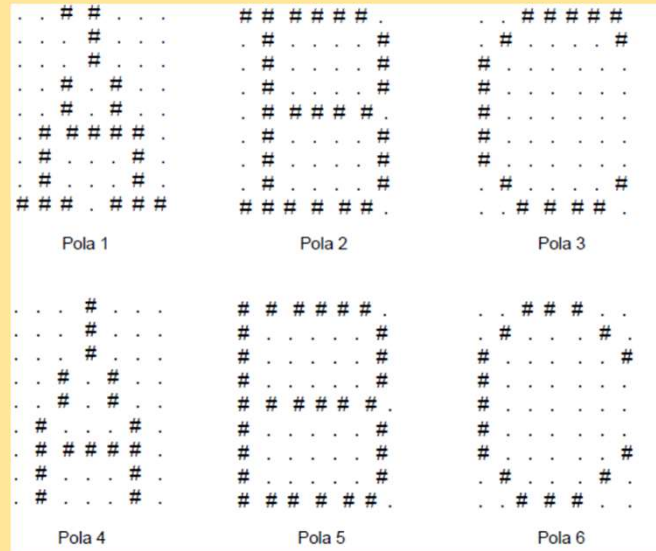
# Pengenalan Pola Karakter

- Perceptron dapat pula dipakai untuk mengenali pola karakter. Dengan berbagai pola masukan yang menyerupai huruf-huruf alphabeth, perceptron dapat dilatih untuk mengenalinya.
- Algoritma untuk mengenali apakah pola masukan yang diberikan menyerupai sebuah karakter tertentu (misal mirip huruf "A") atau tidak adalah sebagai berikut :
  1. Nyatakan tiap pola masukan sebagai vektor bipolar yang elemennya adalah tiap titik dalam pola tersebut.
  2. Berikan nilai target = +1 jika pola masukan menyerupai huruf yang diinginkan. Jika sebaliknya, berikan nilai target = -1
  3. Berikan inisialisasi bobot, bias, laju pemahaman dan **threshold**
  4. Lakukan proses pelatihan perceptron



# Contoh Pengenalan Pola

- Diketahui 6 buah pola masukan seperti gambar



- Buatlah model perceptron untuk mengenali pola “A”.



- **Penyelesaian**

- Untuk menentukan vektor masukan, tiap titik dalam pola diambil sebagai komponen vektor. Jadi tiap vektor masukan memiliki  $9 \times 7 = 63$  komponen. Titik dalam pola yang bertanda “#” diberi nilai = +1 dan titik bertanda “.” diberi nilai -1. Pembacaan pola dilakukan dari kiri ke kanan, dimulai dari baris paling atas.

Vektor masukan pola 1 adalah

(-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 1 -1 -1 -1 1 -1    1 1 1 -1 1 1 1)

Vektor masukan pola 2 adalah

(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 1 -1 -1 -1 -1 1    1 1 1 1 1 1 -1)

Vektor masukan pola 3 adalah

(-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 1 -1 -1 -1 -1 1    -1 -1 1 1 1 1 -1)



Vektor masukan pola 4 adalah

(-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 1 -1 -1 -1 1 -1      -1 1 -1 -1 -1 1 -1)

Vektor masukan pola 5 adalah

(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 -1 -1 -1 -1 -1 1      1 1 1 1 1 1 -1)

Vektor masukan pola 6 adalah

(-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 1 -1 -1 -1 1 -1      -1 -1 1 1 1 -1 -1)

- Target bernilai = +1 bila pola masukan menyerupai huruf “A”. Jika tidak, maka target bernilai = -1. Pola yang menyerupai huruf “A” adalah pola 1 dan pola 4. Pasangan pola dan targetnya tampak pada tabel



Pola Masukan	Target
Pola 1	1
Pola 2	-1
Pola 3	-1
Pola 4	1
Pola 5	-1
Pola 6	-1

- Misalkan bobot awal diambil = 0 untuk semua bobot maupun bias, laju pemahaman diambil  $\alpha = 1$  dan *threshold* = 0.5
- Pelatihan dilakukan dengan cara memasukkan 63 unit masukan (atau sebuah pola huruf).
- Dihitung net, dimana : 
$$\text{net} = \sum_{i=1}^{63} x_i w_i + b$$





- Berikutnya, fungsi aktivasi dihitung menggunakan persamaan

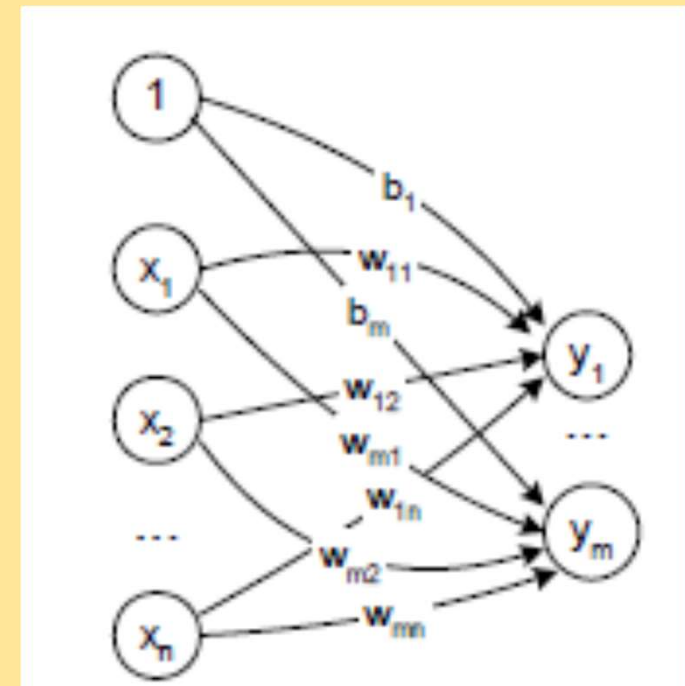
$$y = f(\text{net}) = \begin{cases} 1 & \text{jika } \text{net} > 0.5 \\ 0 & \text{jika } -0.5 \leq \text{net} \leq 0.5 \\ -1 & \text{jika } \text{net} < -0.5 \end{cases}$$

- Apabila  $f(\text{net}) \neq \text{target}$ , maka bobot dan bias diubah
- Proses pelatihan dilakukan terus hingga semua keluaran jaringan sama dengan targetnya.



# Pengenalan Beberapa Pola Karakter

- Pengenalan beberapa pola karakter sekaligus (misal “A” atau bukan “A”, “B” atau bukan “B”, dan seterusnya) dilakukan dengan cara menggabungkan beberapa model perceptron. Jadi ada beberapa unit keluaran sekaligus
- Setiap unit masukan dihubungkan dengan setiap unit target. Bobot penghubung dari unit  $x_i$  ke  $y_j$  adalah  $w_{ji}$  (perhatikan bagaimana indeks dibuat). Demikian juga bias dihubungkan dengan semua unit keluaran dengan bobot masing-masing  $b_1, b_2, \dots, b_m$





Algoritma pelatihannya sbb:

1. Nyatakan tiap pola masukan sebagai vektor bipolar yang elemennya adalah tiap titik dalam pola tersebut
2. Berikan nilai target  $t_j = 1$  jika pola masukan menyerupai huruf yang diinginkan. Jika sebaliknya, berikan nilai target  $t_j = -1$  ( $j = 1, \dots, m$ )
3. Berikan inisialisasi bobot, bias, laju pembelajaran dan threshold
4. Lakukan proses pelatihan perseptron seperti contoh 1 di atas

a. Hitung repson unit keluaran ke-j :  $net_j = \sum_{i=1}^{63} x_i w_{ji} + b_j$

$$y_j = f(net_j) = \begin{cases} 1 & \text{jika } net > \theta \\ 0 & \text{jika } -\theta \leq net \leq \theta \\ -1 & \text{jika } net < -\theta \end{cases}$$

b. Perbaiki bobot pola yang mengandung kesalahan ( $y_j \neq t_j$ ) menurut persamaan:

$$w_{ji} (\text{baru}) = w_{ji} (\text{lama}) + \alpha t_j x_i$$

$$b_j (\text{baru}) = b_j (\text{lama}) + \alpha t_j$$

c. Lakukan langkah 4a – b terus menerus hingga  $y_j = t_j$  ( $j = 1, \dots, m$ )



- **Contoh :**

- Diketahui 6 buah pola masukan seperti pada gambar sebelumnya. Buatlah model perseptron untuk mengenali pola "A", "B" dan "C".

- **Penyelesaian:**

- Mula-mula dibuat 6 buah vektor masukan seperti contoh di atas. Ada 3 buah vektor keluaran yang masing-masing menyatakan bahwa pola menyerupai huruf "A", "B" dan "C". Vektor target tampak pada tabel berikut

Pola masukan	$t_1$	$t_2$	$t_3$
Pola 1	1	-1	-1
Pola 2	-1	1	-1
Pola 3	-1	-1	1
Pola 4	1	-1	-1
Pola 5	-1	1	-1
Pola 6	-1	-1	1



Selanjutnya iterasi dapat dibuat seperti pada contoh-contoh di atas, dengan beberapa perubahan sbb:

1. Vektor masukan terdiri dari 63 elemen  $x_1 \dots x_{63}$  dan sebuah bias
2. Ada 3 target  $t_1, t_2$  dan  $t_3$
3. Ada 3 kolom net yaitu  $net_1, net_2, net_3$  masing-masing merupakan hasil kali bobot dengan vektor masukan
4. Ada 3 buah  $y$  yang merupakan fungsi aktivasi ketiga kolom net.  $y_1 = f(net_1)$ ,  $y_2 = f(net_2)$ ,  $y_3 = f(net_3)$
5. Ada  $3 * 63$  buah kolom perubahan bobot yaitu  
 $\Delta W_{1,1}, \dots \Delta W_{1,63}; \Delta W_{2,1}, \dots \Delta W_{2,63}; \Delta W_{3,1}, \dots \Delta W_{3,63}$
6. Ada  $3*63$  bobot  
 $W_{1,1}, \dots W_{1,63}; W_{2,1}, \dots W_{2,63}; W_{3,1}, \dots W_{3,63}$

