

# Sebelumnya...

---

- Pembelajaran Mesin/*Machine Learning*
- Pembelajaran dengan *Decision Tree* (ID3)
- Teori Bayes dalam Pembelajaran

# Kecerdasan Buatan

Pertemuan 11

## Jaringan Syaraf Tiruan *(Artificial Neural Network)*

Kelas 10-S1TI-03, 04, 05

Husni

[Lunix96@gmail.com](mailto:Lunix96@gmail.com)

<http://Komputasi.wordpress.com>

# *Outline*

---

- Pengertian Jaringan Syaraf Tiruan
- Kesamaan Jaringan Syaraf Manusia & Tiruan
- Jenis Jaringan Syaraf
- Perceptron
- Perceptron Banyak Lapisan (Multi-layer)

# Pendahuluan

---

- Jaringan Syaraf Tiruan (*Artificial Neural Networks*, ANN) diinspirasi oleh cara kerja sistem syaraf manusia (biologi), cara otak mengelola informasi.
- Sistem pemrosesan informasi tersusun dari sejumlah besar elemen pemrosesan yang saling terhubung (neuron) yang bekerja sama memecahkan masalah tertentu.
- ANN, seperti manusia, belajar berdasarkan contoh (*learn by example*).
- Pembelajaran ANN termasuk menyesuaikan koneksi synaptic yang ada di antara neuron-neuron.

# Pendahuluan (2)

---

- *Neural networks* belajar berdasarkan contoh informasi.
- Tidak dapat diprogram untuk mengerjakan tugas tertentu secara langsung
- Contoh informasi harus dipilih secara cermat agar tidak boros waktu, bahkan (pada kondisi terburuk) *network* tidak berfungsi dengan benar
- *Network* menemukan bagaimana menyelesaikan masalah dengan sendirinya, sehingga operasinya (sangat mungkin) tidak dapat diprediksi.

# Jaringan Syaraf Manusia

---

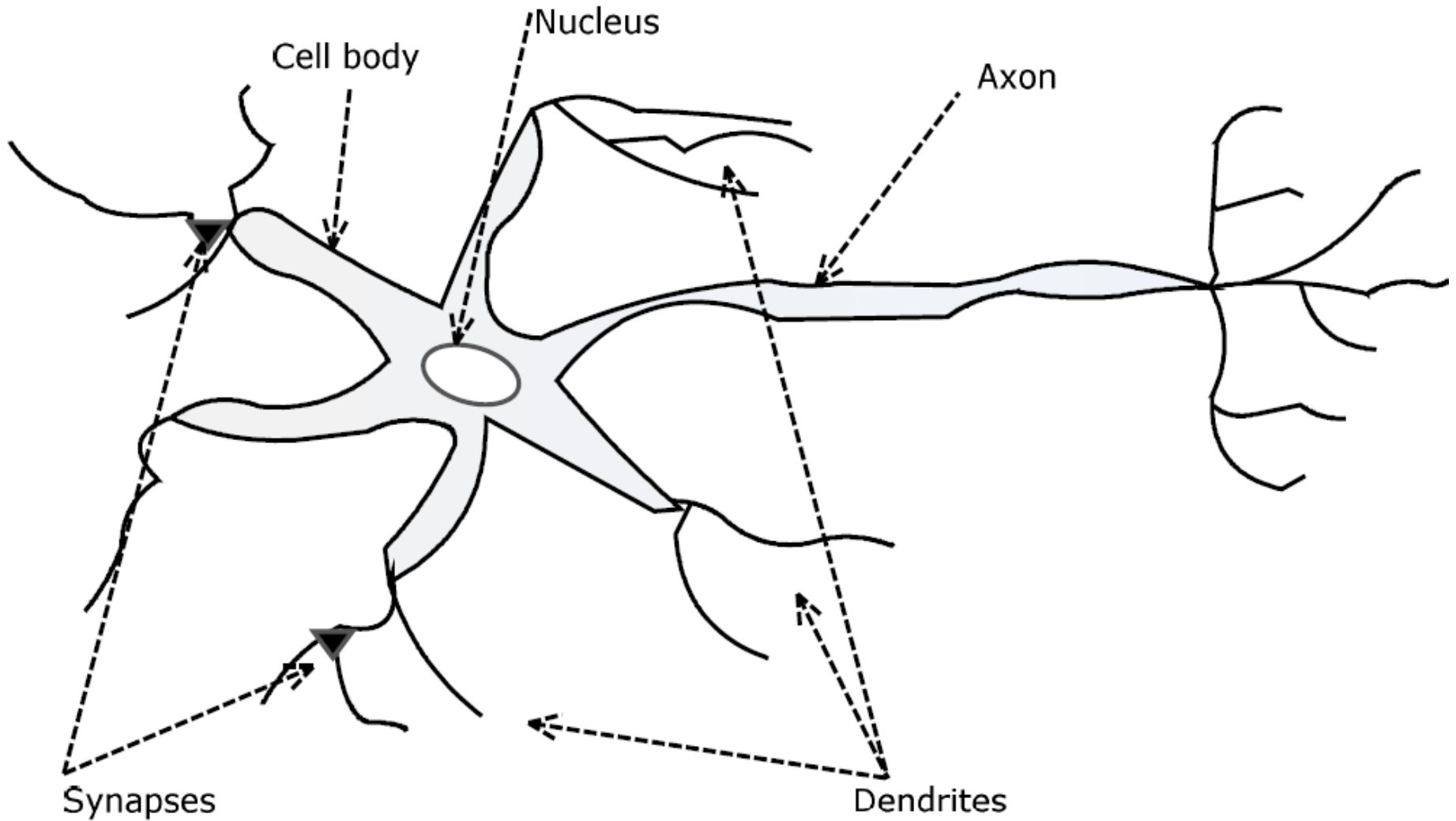
- Terdiri dari sekitar 100 milyar sel syaraf (*nervous*).
- Milyaran neuron saling terhubung membentuk jaringan kompleks.
- Dalam korteks otak besar terdapat sekitar 10 milyar neuron dan dan 60 trilyun sinapsis.
- **Pendapat lain:** Jumlah total neuron 20 milyar, sinapsis 240 trilyun.
- Masa dari suatu *large sensory neuron* sekitar  $10^{-6}$  gram . Jumlah sinapsis dari satu neuron antara 1.000 s.d 10.000.
- Semua fungsi mental dan fisik tergantung pada pembentukan & perawatan jaringan neuron.

# Struktur Neuron

---

- Suatu neuron terdiri dari soma (cell body), axon (long fiber) dan dendrit.
- Axon mengirimkan sinyal dan dendrit menerima sinyal.
- Sinapsis menghubungkan axon ke dendrite.
- Diberikan suatu sinyal, sinapsis dapat menaikkan (membangkitkan) atau menurunkan (menghambat) potensi listrik.
- Neuron aktif/bekerja ketika potensi listriknya mencapai suatu ambang (*threshold*).

# Struktur Neuron

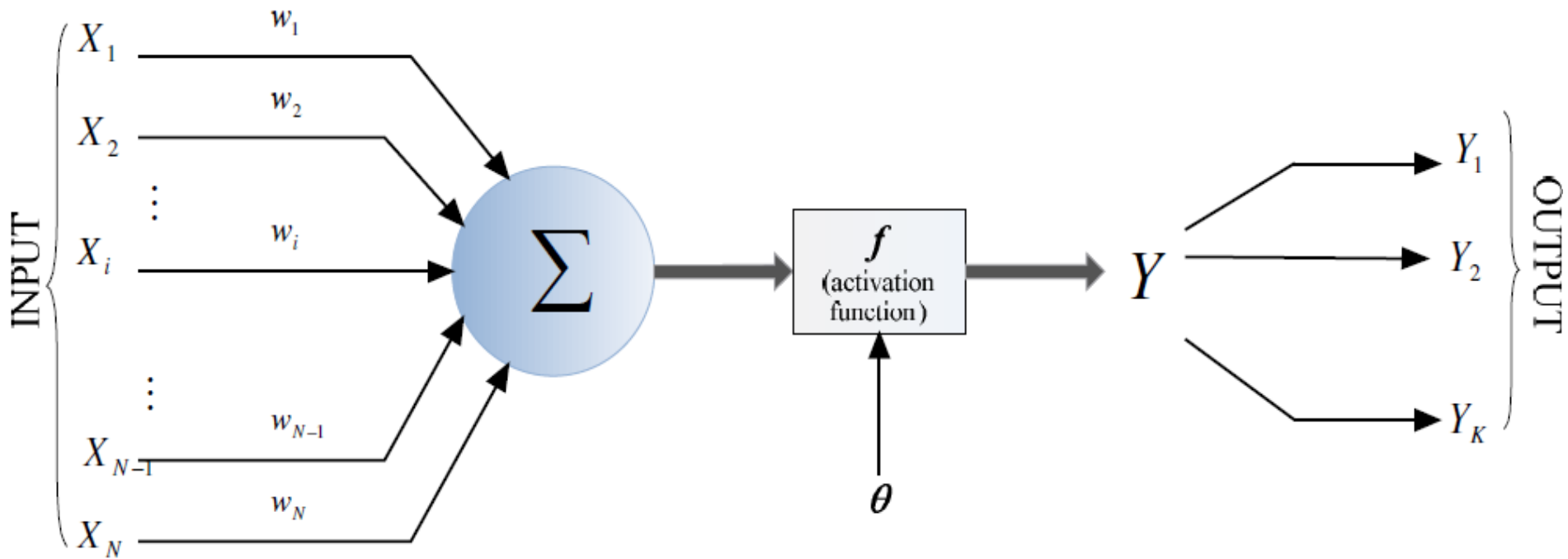




# Jaringan Syaraf Tiruan

---

- Terdiri dari sejumlah neurons (units), disusun dalam lapisan-lapisan (layers), sejumlah koneksi yang dikerjakan oleh link terbobot yang berperan mengirimkan sinyal dari satu neuron ke neuron lainnya.
- Sinyal output dikirimkan melalui *neuron's outgoing connection* (seperti axon).
- *Outgoing connection* dipecah ke dalam sejumlah cabang yang mengirimkan sinyal sama.
- *Outgoing branches* berhenti pada *incoming connections* dari neuron lain dalam jaringan.
- *Inputs* dan *outputs* bernilai numerik.



### Jaringan Syaraf Manusia

Soma (badan sel)

Dendrit, Axon

Sinapsis

Potensi

Sinyal

### Jaringan Syaraf Tiruan

Neuron

Koneksi (input, output)

bobot

Jumlah terbobot

Aktifasi

# Komponen Utama

---

- Sekumpulan unit pengolahan, disebut neuron atau sel;
- Status aktivasi  $Y_i$  untuk setiap unit, ekuivalen dengan output dari unit tersebut;
- Koneksi antar units; tiap koneksi berbobot  $w_{jk}$  (sinyal unit  $j$  ada pada unit  $k$ ).  $w_{jk}$  positif dianggap *excitation* dan negatif sebagai *inhibition*.
- Aturan (rule) Propogasi (perambatan), menentukan inout efektif  $X_i$  dari suatu unit dari input eksternalnya;
- Fungsi aktivasi  $f$ , menentukan level baru dari aktivasi berdasarkan pada input efektif  $X_i(t)$  dan aktivasi terkini  $Y_i(t)$ ;
- Input eksternal (atau bias, offset)  $\theta_i$  untuk setiap unit;
- Metode pemrolehan informasi (learning rule);
- Lingkungan sistem beroperasi, menyediakan sinyal input, mungkin juga sinyal error

# Jenis Unit /Neuron

---

- Input
- Hidden
- Output
- Neurons menerima input dari tetangga atau sumber eksternalnya dan menggunakan ini untuk menghitung sinyal output yang dijalankan ke unit lain.
- Neuron juga bertugas mengatur bobot
- Sistem bersifat paralel, banyak unit dapat mengerjakan komputasi pada waktu yang sama.
- **Tujuan NN:** melatih jaringan untuk mencapai keseimbangan antara kemampuan untuk merespon pola input yang digunakan untuk digunakan selama training dan kemampuan untuk memberikan respon *reasonable* untuk input baru yang serupa (tapi tak identik) dengan pola selama training.

# Jenis Neural Network

---

- Jumlah lapisan pemroses informasi:
  - *single layer network (perceptron)*;
  - *multi-layer network*;
- Geometri koneksi:
  - *feedforward network*;
  - *backpropagation network*;
- Algoritma adaptasi bobot link.

# Fungsi-fungsi Aktifasi

- Tanda (sign):  $Aktifasi(X) = \begin{cases} -1, & \text{jika } X \leq 0 \\ 1, & \text{jika } X > 0 \end{cases}$
- Step  $Aktifasi(X) = \begin{cases} 0, & \text{jika } X \leq 0 \\ 1, & \text{jika } X > 0 \end{cases}$
- Sigmoid/Logistik  $Aktifasi(X) = \frac{1}{1 + e^{-x}}$
- Bipolar  $Aktifasi(X) = \begin{cases} 1, & \text{jika } x > 0 \\ 0, & \text{jika } x = 0 \\ -1, & \text{jika } x < 0 \end{cases}$
- **Fungsi lain:** Identitas, *Than*, *Arctan*, *Exponential*, *Reciprocal*, *Gaussian*, *Sine*, *Cosine*, dan *Elliott*

# Aturan Hebb

---

- Pembelajaran dengan memperbaiki nilai bobot
- Jika dua neuron terhubung, hidup (*on*) bersamaan, bobot keduanya dinaikkan.

$$w_i(\text{baru}) = w_i(\text{lama}) + x_i * y$$

$w_i$  : bobot data input ke-i

$x_i$  : input data ke-i

$y$  : data output

# Algoritma Aturan Hebb

---

- Inisiasi semua bobot:  $w_{ij} = 0$ , dimana  $i = 1, 2, 3, \dots, n$  dan  $j = 1, 2, \dots, m$
- Terhadap setiap pasangan input-output (s-t), lakukan:
  - Set input dengan nilai sama dengan vektor input:  
 $x_i = s_i$
  - Set output dengan nilai sama dengan vektor output:  
 $y_j = t_j$
  - Perbaiki bobot:  
 $w_i(\text{baru}) = w_i(\text{lama}) + x_i * y$

**Catatan:** nilai bias harus selalu 1.

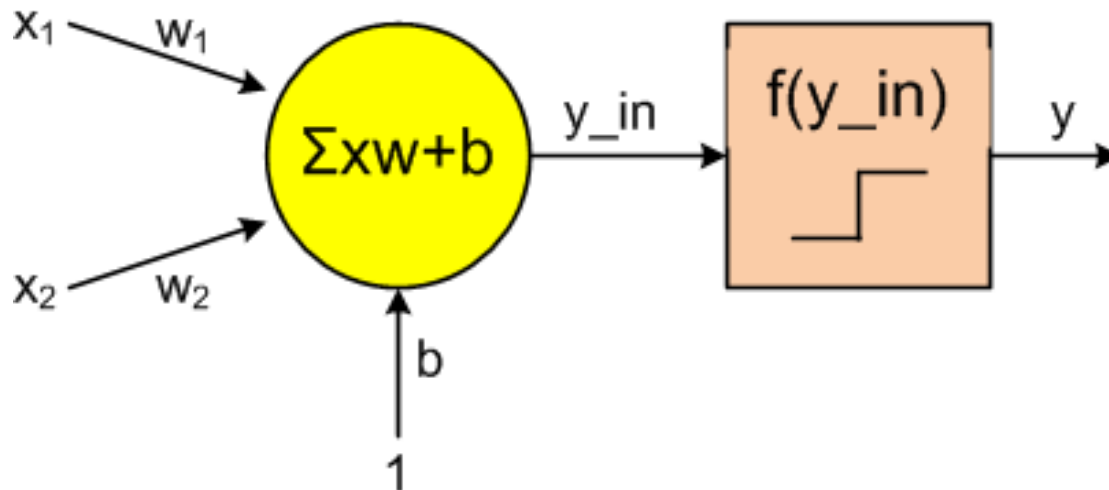


# Contoh: Pembelajaran Fungsi OR

- Representasi  
input: bipolar  
target: bipolar

Input		Bias	Target
-1	-1	1	-1
-1	1	1	1
1	-1	1	1
1	1	1	1

- Bobot awal dan bobot bias 0



# Hebb: Langkah-langkah

---

- Kondisi awal:

$$X = -1,-1; -1,1; 1,-1; 1,1;$$

$$T = -1; 1; 1; 1;$$

$$W = 0; 0$$

$$B = 0$$

## Perubahan Bobot:

- Data ke-1:

$$w_1(\text{baru}) = w_1(\text{lama}) + x_1 * y = 0 + (-1 * -1) = 1$$

$$w_2(\text{baru}) = 0 + 1 = 1$$

$$bb(\text{baru}) = bb(\text{lama}) + b * y = 0 - 1 * 1 = -1$$

# Hebb: Langkah-langkah

- Data ke-2:

$$w_1 = 1 - 1 = 0$$

$$w_2 = 1 + 1 = 2$$

$$b = -1 + 1 = 0$$

- Data ke-3:

$$w_1 = 0 + 1 = 1$$

$$w_2 = 2 - 1 = 1$$

$$b = 0 + 1 = 1$$

- Data ke-4:

$$w_1 = 1 + 1 = 2$$

$$w_2 = 1 + 1 = 2$$

$$b = 1 + 1 = 2$$

- **Pengujian**, misal input  $x = [-1, -1]$

$$y = 2 + (-1 * 2) + (-1 * 2) = -2$$

Karena  $y_{in} = -2$ , maka:

$$y = f(y_{in}) = f(-2) = -1$$

Nilai -1 cocok dengan output yang diharapkan.

**Catatan:** Fungsi aktivasi bipolar

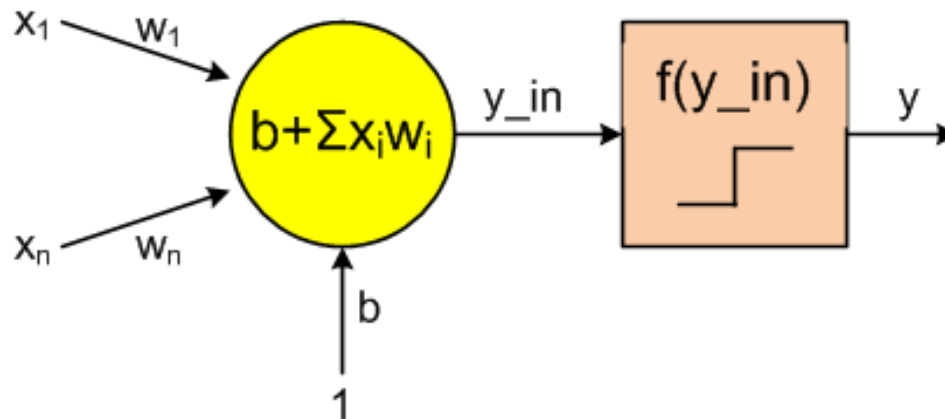
# Pendekatan *Single Neuron*

- Hebb Rule, Perceptron dan Delta Rule sering digunakan. Boleh melibatkan bias (dan bobotnya)
- Perbedaan ada pada cara menentukan bobot.

Hebb rule:  $w_i(\text{baru}) = w_i(\text{lama}) + x_i * y_d$

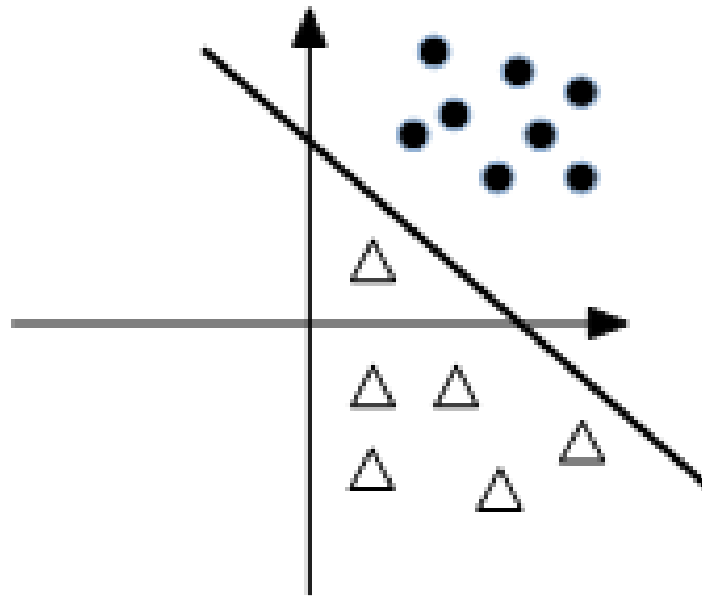
Perceptron:  $w_i(\text{baru}) = w_i(\text{lama}) + \alpha * y_d * x_i$

Delta rule:  $w_i(\text{baru}) = w_i(\text{lama}) + \alpha * (y - y_d) * x_i$



# Perceptron & Delta Rule

- Hasil akhir membentuk *hyperplane* sebagai pembatas keputusan.



$$w_1 X_1 + w_2 X_2 - \theta = 0$$

- Perceptron (dan Delta Rule) dapat belajar dari contoh yang *linearly separable*.

# Contoh: Perceptron untuk Fungsi AND

$X_1$	$X_2$	Y
1	1	1
1	0	0
0	1	0
0	0	0

- Gunakan fungsi aktivasi Step
- Bobot awal:  $w_1 = 0, w_2 = 0$
- Bobot bias:  $bb = 0$
- Learning rate:  $\alpha = 0,8$
- Threshold:  $\theta = 0,5$

# Epoch ke-1

---

- Data ke-1:
- $Y_{in} = (bb + x_1w_1 + x_2w_2) - \theta$   
 $= (0 + 0 + 0) = 0$
- Hasil aktivasi: 0 (karena  $Y_{in}$  tidak lebih dari 0)
- $Y$  yang diharapkan? 1, selisih? 1
- Bobot baru:  
 $w_1 = 0 + 0,8 * 1 * 1 = 0,8$   
 $w_2 = 0 + 0,8 * 1 * 1 = 0,8$   
 $bb = 0 + 0,8 * 1 = 0,8$

# Epoch ke-1

---

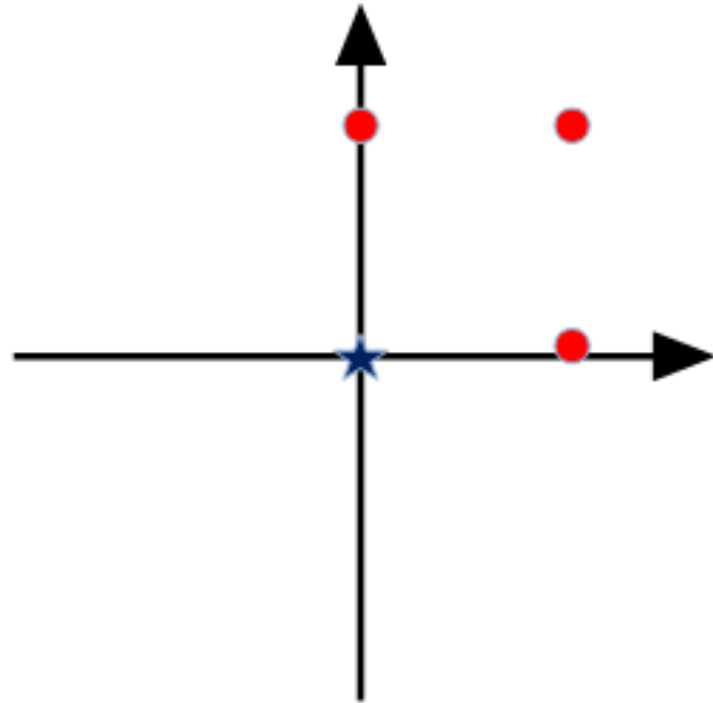
- Data ke-2 ???
- ...
- ...
- Lakukan terus menerus sampai tidak terdapat error, selisih  $Y$  dan  $Y_{in}$  selalu 0 untuk semua Data



# Contoh: Fungsi OR, Delta Rule Tanpa Bias

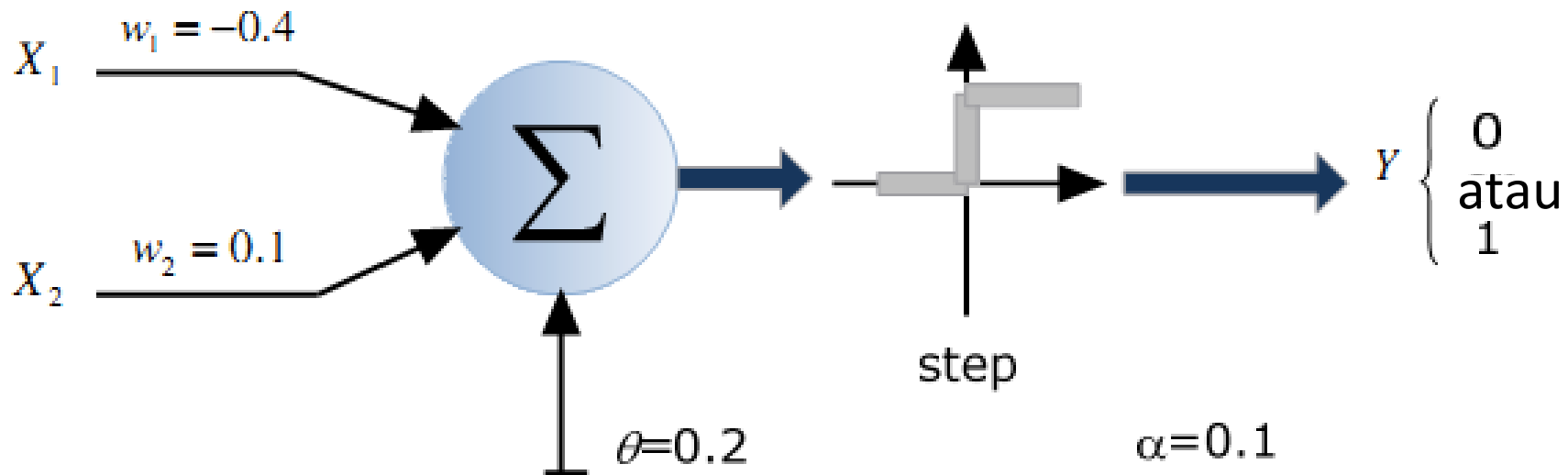
- Tabel kebenaran dan representasi grafisnya

$X_1$	$X_2$	$Y$
0	0	0
1	0	1
0	1	1
1	1	1



# Nilai-nilai Awal

- Bobot  $w_1$  dan  $w_2$ , serta *threshold*  $\theta$  diberikan suatu nilai awal. Ada pula suatu *learning rate*  $\alpha$ .



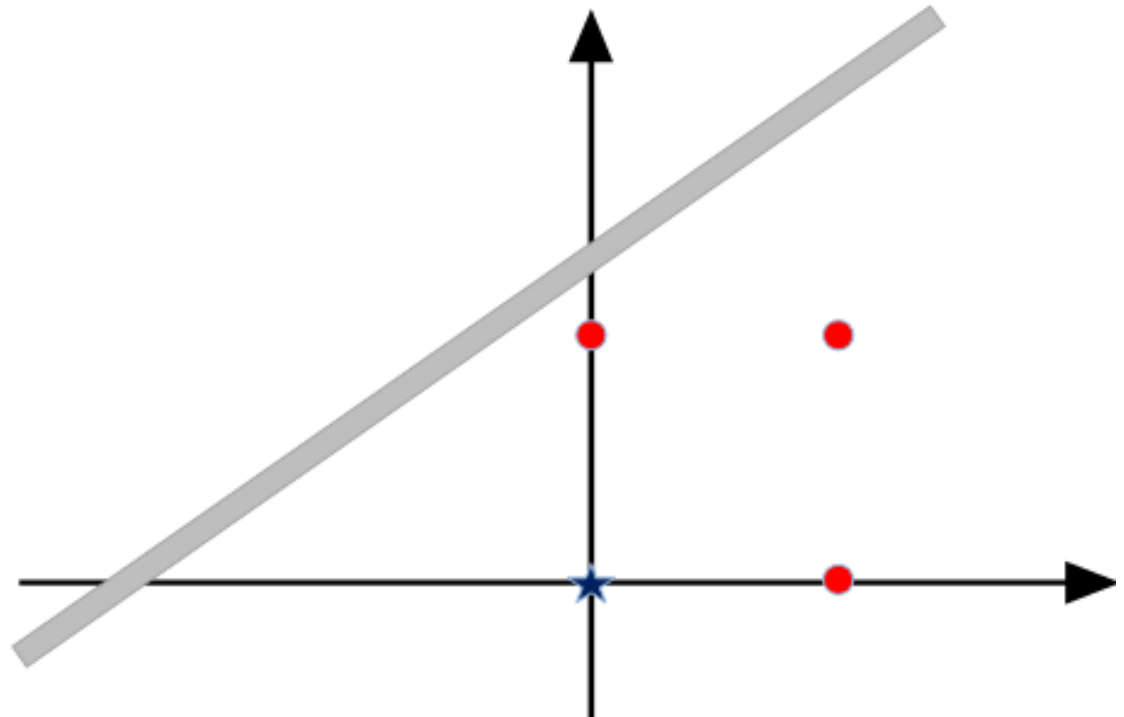
# Persamaan dari *Separator*

- Berdasarkan konfigurasi awal, diperoleh:

$$w_1X_1 + w_2X_2 - \theta = 0$$

$$-0.4X_1 + 0.1 \cdot X_2 - 0.2 = 0$$

$$4X_1 - X_2 + 2 = 0$$



# Langkah-langkah

Input		Desired output	Initial weights		Obtained output	error	Updated weights		Epoch
$X_1$	$X_2$	$Y_d$	$w_1$	$w_2$	$Y$	$e$	$w_1$	$w_2$	
0	0	0	-0.4	0.1	0	0	-0.4	0.1	1
1	0	1	-0.4	0.1	0	1	-0.3	0.1	
0	1	1	-0.3	0.1	0	1	-0.3	0.2	
1	1	1	-0.3	0.1	0	1	-0.2	0.3	
0	0	0	-0.2	0.3	0	0	-0.2	0.3	2
1	0	1	-0.2	0.3	0	1	-0.1	0.3	
0	1	1	-0.1	0.3	1	0	-0.1	0.3	
1	1	1	-0.1	0.3	1	0	-0.1	0.3	
0	0	0	-0.1	0.3	0	0	-0.1	0.3	3
1	0	1	-0.1	0.3	0	1	0	0.3	
0	1	1	0	0.3	1	0	0	0.3	
1	1	1	0	0.3	1	0	0	0.3	

# Langkah-langkah

Input		Desired output	Initial weights		Obtained output	error	Updated weights		Epoch
$X_1$	$X_2$	$Y_d$	$w_1$	$w_2$	$Y$	$e$	$w_1$	$w_2$	
0	0	0	0	0.3	0	0	0	0.3	4
1	0	1	0	0.3	0	1	0.1	0.3	
0	1	1	0.1	0.3	1	0	0.1	0.3	
1	1	1	0.1	0.3	1	0	0.1	0.3	
0	0	0	0.1	0.3	0	0	0.1	0.3	5
1	0	1	0.1	0.3	0	1	0.2	0.3	
0	1	1	0.2	0.3	1	0	0.2	0.3	
1	1	1	0.2	0.3	1	0	0.2	0.3	
0	0	0	0.2	0.3	0	0	0.2	0.3	6
1	0	1	0.2	0.3	1	0	0.2	0.3	
0	1	1	0.2	0.3	1	0	0.2	0.3	
1	1	1	0.2	0.3	1	0	0.2	0.3	

# Langkah-langkah

---

- Data training pertama, nilai 0 untuk X1, X2 dan Yd.
- Bobot awal -0.4 dan 0.1.
- Delta rule diaktifkan dengan fungsi aktivasi Step
- Nilai threshold 0.2.
- Output yang diperoleh:  
$$Y = \text{step}[0 \cdot (-0.4) + 0 \cdot 0.1 - 0.2] = \text{step}[-0.2] = 0$$
- Output yang diperoleh bernilai 0, sama dengan output yang diharapkan.
- Tidak ada perubahan bobot.

# Langkah-langkah

---

- Data training ke-2:  $X_1 = 1$ ,  $X_2 = 0$  dan  $Y_d = 1$ .

- Diperoleh output:

$$Y = \text{step}[1 \cdot (-0.4) + 0 \cdot 0.1 - 0.2] = \text{step}[-0.6] = 0$$

- Output yang diperoleh berbeda dengan output yang diharapkan. Errornya 1 (selisih).

- Bobot harus diupdate:

$$w_1 = w_1 + \alpha \cdot X_1 \cdot e$$

- Dengan learning rate 0.1, diperoleh

$$w_1 = -0.4 + 0.1 \cdot 1 \cdot 1 = -0.3$$

$$w_2 = 0.1 + 0.1 \cdot 0 \cdot 1 = 0.1$$

# Langkah-langkah

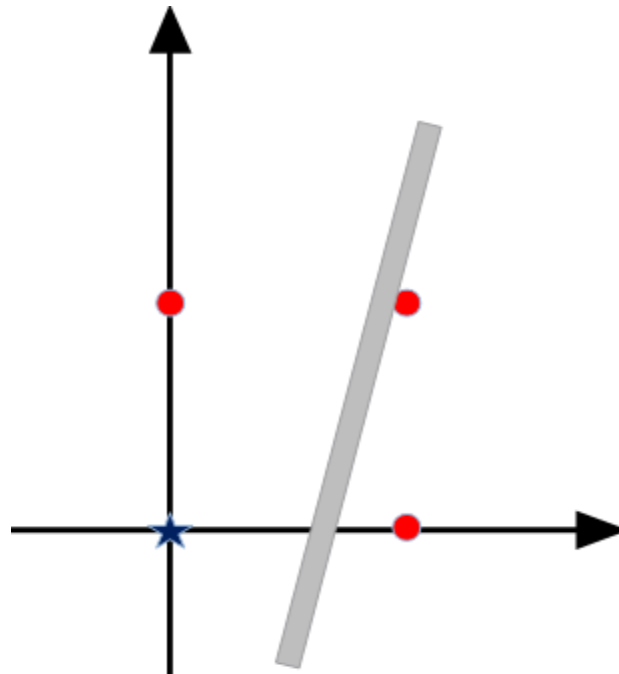
---

- Data training ke-3: ( $X_1 = 0$ ,  $X_2 = 1$  dan  $Y_d = 1$ ) dengan bobot *terupdate*  $w_1 = -0.3$  dan  $w_2 = 0.1$ .
- Diperoleh output:  
$$Y = \text{step}[0 \cdot (-0.3) + 1 \cdot 0.1 - 0.2] = \text{step}[-0.1] = 0$$
- Tidak sama dengan output yang diharapkan, selisihnya 1,  $\text{error} = 1$ .
- Diperoleh bobot baru:  $w_1 = -0.3$  dan  $w_2 = 0.2$ .
- Data training ke-4:  $X_1 = 1$ ,  $X_2 = 1$  dan  $Y_d = 1$ .
- Output yang diperoleh 0 dan yang diharapkan 1. Berarti  $\text{error} = 1$ .
- Update bobot, diperoleh  $w_1 = -0.2$  dan  $w_2 = 0.3$ .



# Langkah-langkah

- Epoch 1 selesai. Data belum terklasifikasi, ada 3 error.
- Lanjut ke epoch ke-2, gunakan bobot yang baru, mulai dari data training pertama.
- Persamaan pemisah di awal epoch 2:



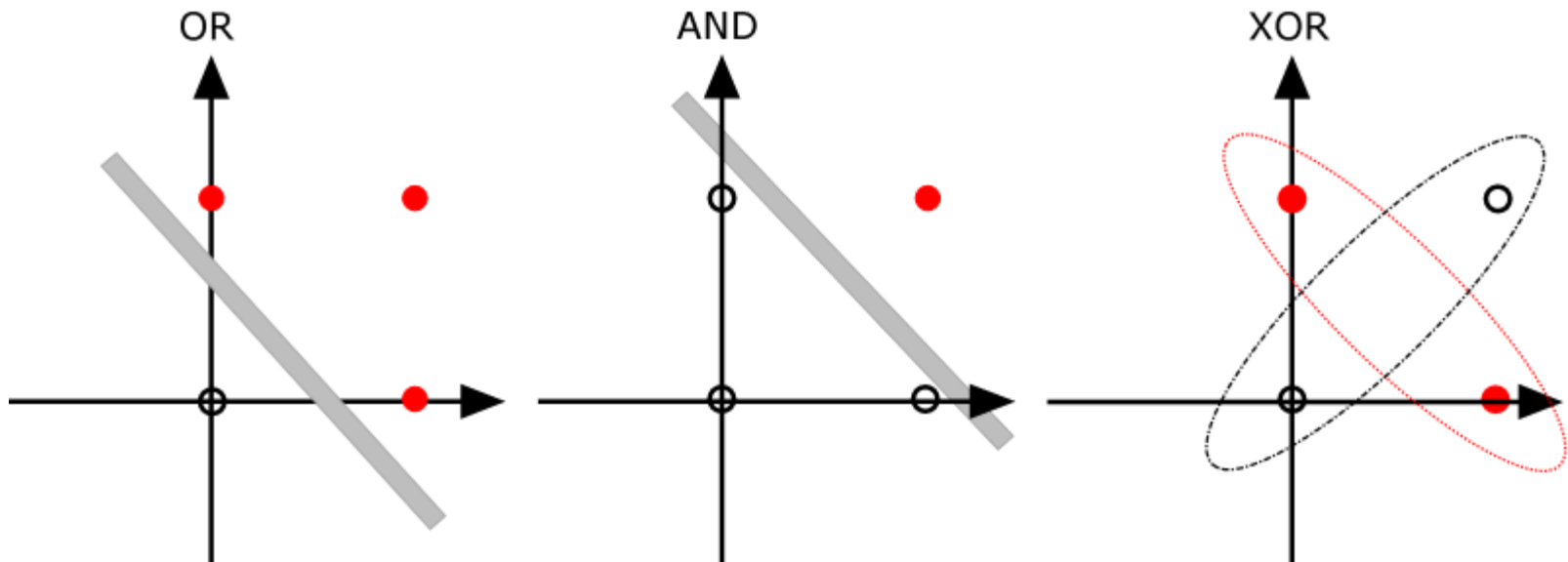
# Kekurangan Perceptron & Delta Rule

---

- Pembelajaran efisien jika bobot tidak terlalu besar;
- Atribut-atribut dibobot secara independen;
- Hanya dapat belajar lines-hyperplanes (tidak dapat belajar exclusive OR, misalnya).

# Jaringan Syaraf Banyak Lapisan

- Representasi grafis dari pemisah pada operasi logika OR, AND dan XOR.



# Backpropagation

---

- Contoh: Jaringan syaraf untuk fungsi XOR.
- Detailnya, lihat di buku Sri Kusumadewi, Artificial Intelligence, Graha Ilmu, Yogyakarta.