

# **MODEL NEURON DAN ARSITEKTUR JARINGAN**

Mengadopsi esensi dasar dari system syaraf biologi, syaraf tiruan digambarkan sebagai berikut :

- Menerima input atau masukan (baik dari data yang dimasukkan atau dari output sel syaraf pada jaringan syaraf. Setiap input datang melalui suatu koneksi atau hubungan yang mempunyai sebuah bobot (*weight*).
- *Setiap sel syaraf mempunyai* sebuah nilai ambang. Jumlah bobot dari input dan dikurangi dengan nilai ambang kemudian akan mendapatkan suatu aktivasi dari sel syaraf (*post synaptic potential, PSP*, dari sel syaraf). Signal aktivasi kemudian menjadi fungsi aktivasi / fungsi transfer untuk menghasilkan output dari sel syaraf.

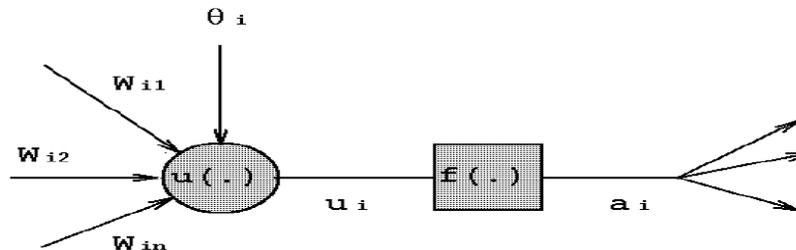
Neuron adalah unit pemroses informasi yang menjadi dasar dalam pengoperasian JST. Neuron terdiri dari 3 elemen:

- Himpunan unit2 yang dihubungkan dengan jalur koneksi. Jalur tersebut memiliki bobot yang berbeda-beda. Bobot yang bernilai positif akan memperkuat sinyal dan yang bernilai negatif akan memperlemah sinyal yang dibawanya. Jumlah, struktur dan pola hubungan antar unit2 tersebut akan menentukan "ARSISTEKTUR JARINGAN" (dan juga model jaringan yang terbentuk).

## 1

# Model Neuron

- Suatu unit penjumlah yang akan menjumlahkan input2 sinyal yang sudah dikalikan dengan bobot. Misalkan  $x_1, x_2, \dots, x_m$  adalah unit2 input dan  $w_{j1}, w_{j2}, \dots, w_{jm}$  adalah bobot penghubung dari unit2 tsb ke unit keluaran  $Y_j$ , maka unit penjumlah akan memberikan keluaran sebesar  $u_j = x_1 w_{j1} + x_2 w_{j2} + \dots + x_m w_{jm}$
- Fungsi aktivasi yang akan menentukan apakah sinyal dari input neuron akan diteruskan ke neuron lain atautkah tidak.
- Jika tahapan fungsi aktivasi digunakan (output sel syaraf = 0 jika input  $< 0$  dan 1 jika input  $\geq 0$ ) maka tindakan sel syaraf sama dengan sel syaraf biologi yang dijelaskan diatas (pengurangan nilai ambang dari jumlah bobot dan membandingkan dengan 0 adalah sama dengan membandingkan jumlah bobot dengan nilai ambang).
- Biasanya tahapan fungsi jarang digunakan dalam Jaringan Syaraf Tiruan. Fungsi aktivasi ( $f(\cdot)$ ) dapat dilihat pada Gambar berikut.



- Bagaimana sel syaraf saling berhubungan?
- Jika suatu jaringan ingin digunakan untuk berbagai keperluan maka harus memiliki input (akan membawa nilai dari suatu variabel dari luar) dan output (dari prediksi atau signal kontrol).
- Input dan output sesuai dengan sensor dan syaraf motorik seperti signal datang dari mata kemudian diteruskan ke tangan, Dalam hal ini terdapat sel syaraf atau neuron pada lapisan tersembunyi berperan pada jaringan ini.
- Input, lapisan tersembunyi dan output sel syaraf diperlukan untuk saling terhubung satu sama lain

Berdasarkan dari arsitektur (pola koneksi), Jaringan Syaraf Tiruan dapat dibagi ke dalam dua katagori :

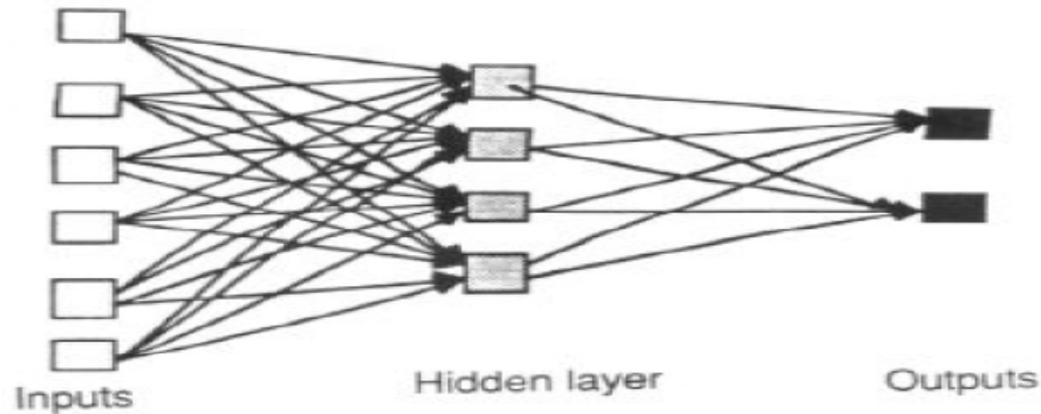
## 1. Struktur *feedforward*

- Sebuah jaringan yang sederhana mempunyai struktur *feedforward* dimana signal bergerak dari input kemudian melewati lapisan tersembunyi dan akhirnya mencapai unit output (mempunyai struktur perilaku yang stabil).
- Tipe jaringan *feedforward* mempunyai sel syaraf yang tersusun dari beberapa lapisan. Lapisan input bukan merupakan sel syaraf. Lapisan ini hanya memberi pelayanan dengan mengenalkan suatu nilai dari suatu variabel.

# 1

## Model Neuron

- Lapisan tersembunyi dan lapisan output sel syaraf terhubung satu sama lain dengan lapisan sebelumnya. Kemungkinan yang timbul adalah adanya hubungan dengan beberapa unit dari lapisan sebelumnya atau terhubung semuanya (lebih baik).



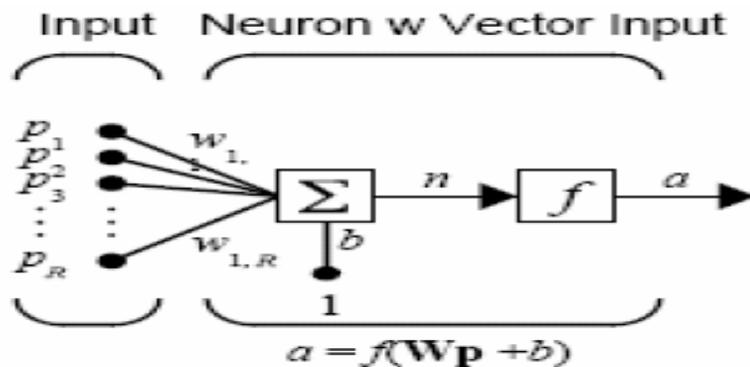
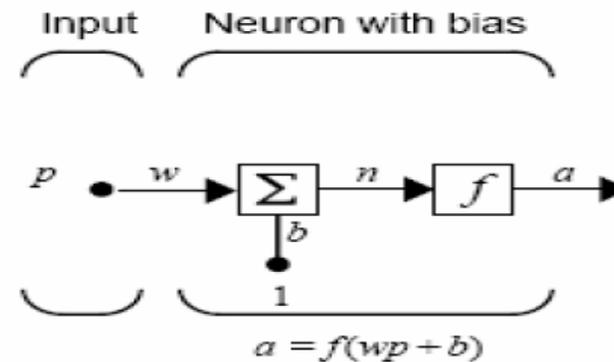
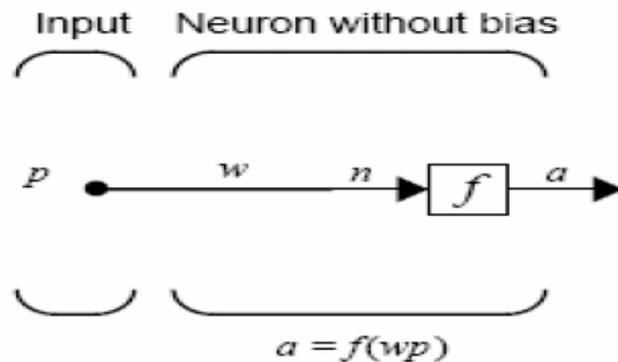
Yang termasuk dalam struktur *feedforward* :

- Single-layer perceptron
- Multilayer perceptron
- Radial-basis function networks
- Higher-order networks
- Polynomial learning networks

# Model Neuron

Contoh:

## Single-Input Neuron



Input jaringan ( $n$ ) dihitung sbb :

$$n = w_{1,1}p_1 + w_{1,2}p_2 + \dots + w_{1,R}p_R + b$$

atau

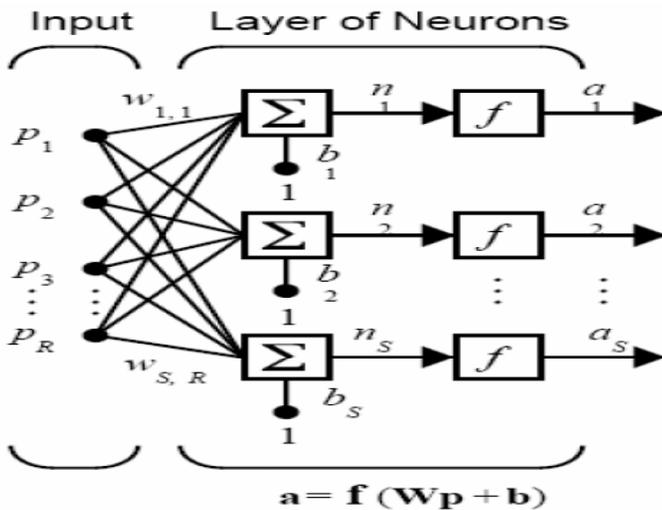
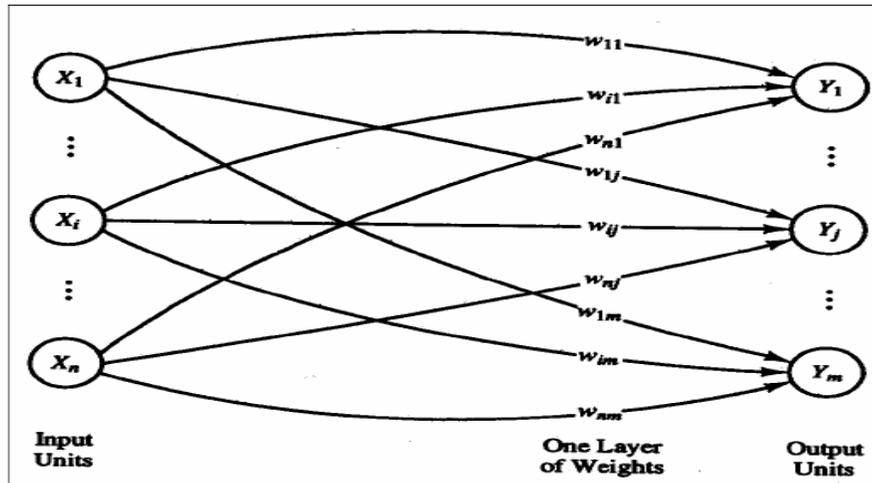
$$n = \mathbf{Wp} + b$$

Output neuron ( $a$ ) :

$$a = f(\mathbf{Wp} + b)$$

# Arsitektur Jaringan

## 1. Neuron selapis



Dengan  
 $R$  sebagai jumlah elemen vektor input dan  
 $S$  sebagai jumlah neuron dalam lapisan

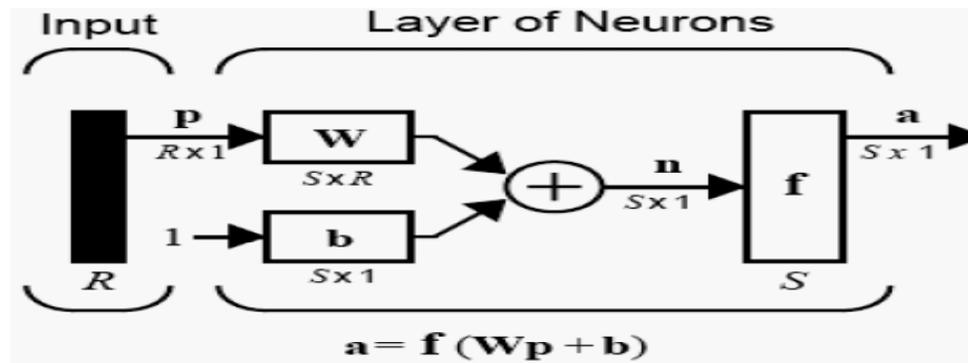
## 1

# Arsitektur Jaringan

- Elemen vektor input jaringan dipresentasikan dalam bentuk matriks :

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \dots & \dots & \dots & \dots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix}$$

- Selapis neuron dapat digambarkan dengan notasi sbb. :

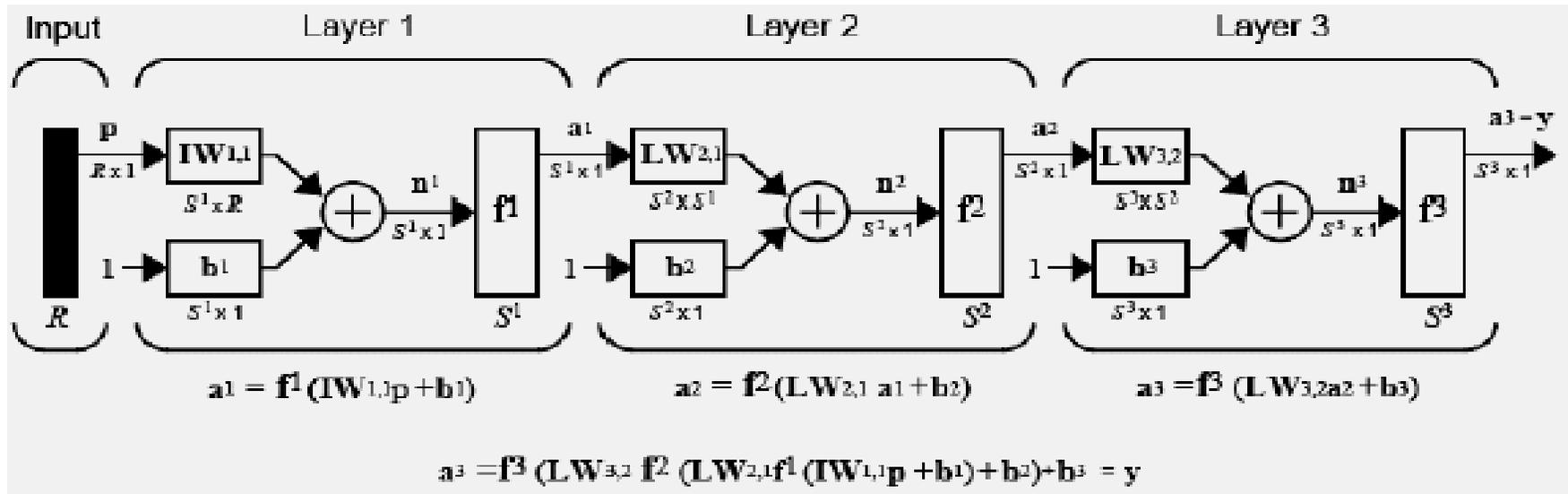


Dengan

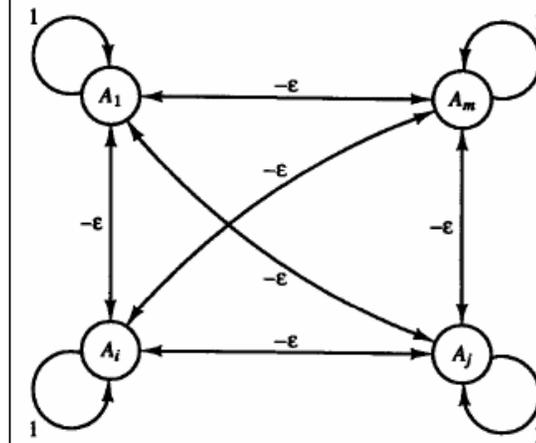
- R sebagai jumlah elemen vektor input dan
- S sebagai jumlah neuron dalam lapisan

# 1

# Arsitektur Jaringan



## 3. Jaringan Kompetitif



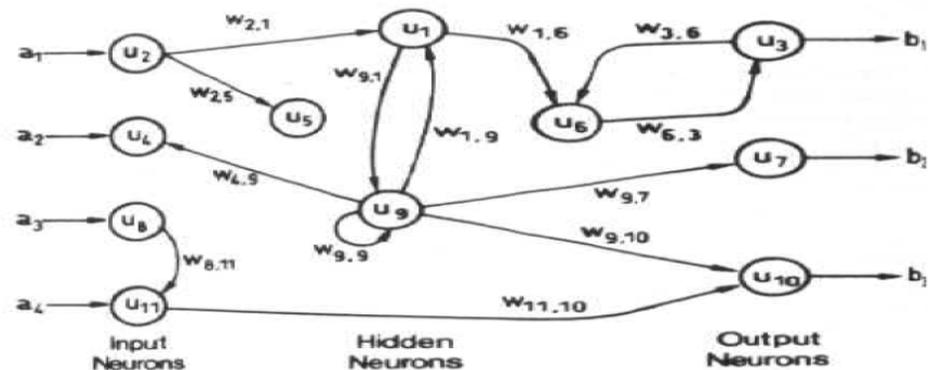
# Arsitektur Jaringan

## 4. Jaringan *recurrent*

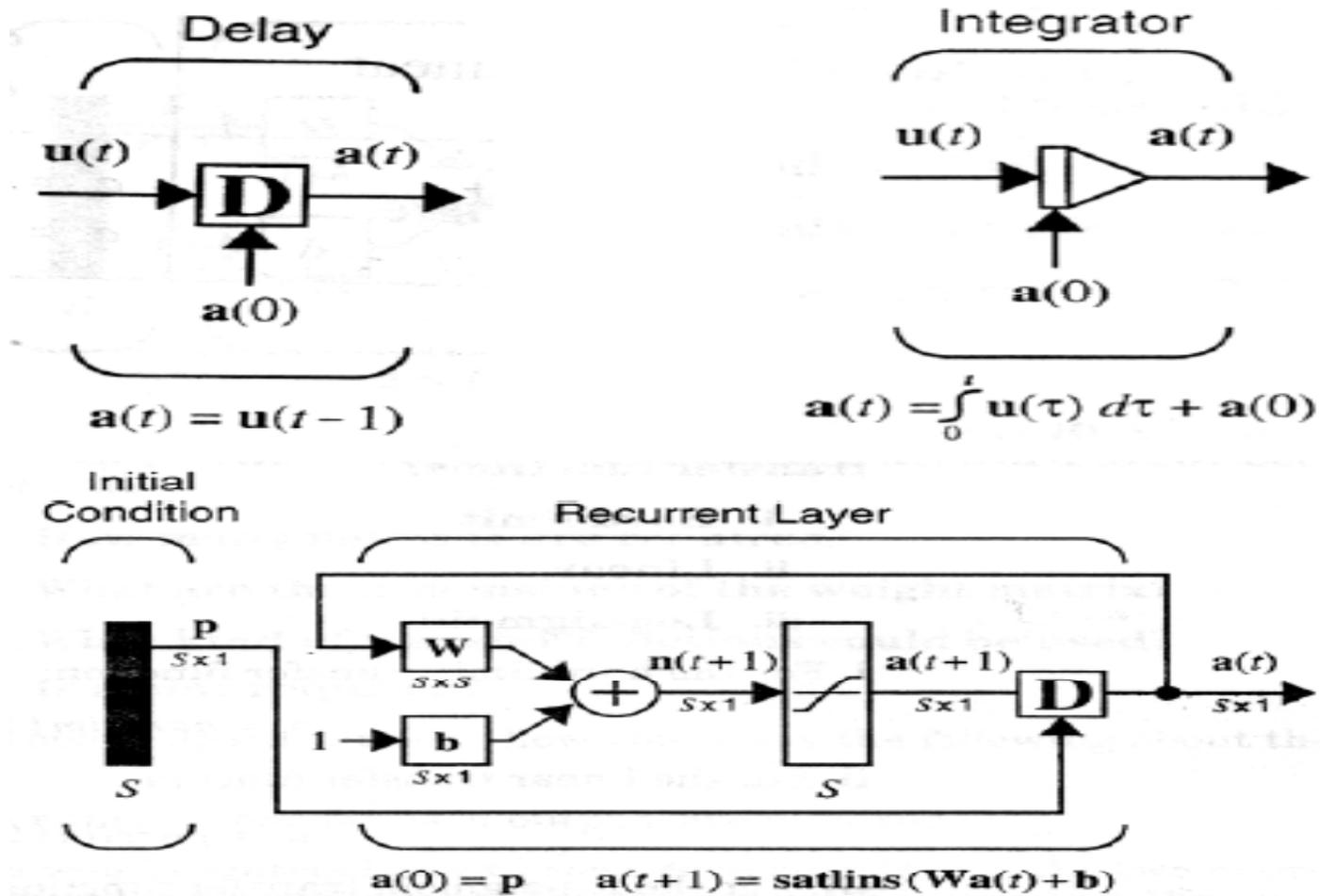
Jika suatu jaringan berulang (mempunyai koneksi kembali dari output ke input) akan menimbulkan ketidakstabilan dan akan menghasilkan dinamika yang sangat kompleks. Jaringan yang berulang sangat menarik untuk diteliti dalam Jaringan Syaraf Tiruan, namun sejauh ini structure *feedforward* sangat berguna untuk memecahkan masalah.

Yang termasuk dalam stuktur *recurrent (feedback)* :

- Competitive networks
- Self-organizing maps
- Hopfield networks
- Adaptive-resonance theory models



# Arsitektur Jaringan

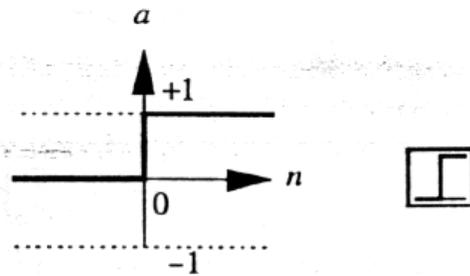


Gambar 4 . Jaringan Syaraf Tiruan *Recurrent Layer*

# Arsitektur Jaringan

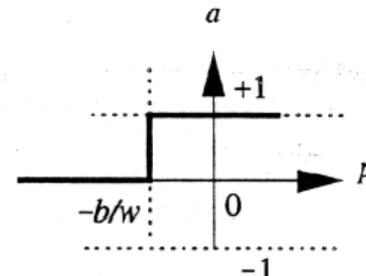
- Ketika sebuah Jaringan Syaraf digunakan, Input dari nilai suatu variabel ditempatkan dalam suatu input unit kemudian unit lapisan tersembunyi dan lapisan output menjalankannya.
- Setiap lapisan tersebut menghitung nilai aktivasi dengan mengambil jumlah bobot output dari setiap unit dari lapisan sebelumnya dan kemudian dikurangi dengan nilai ambang.
- Nilai aktifasi kemudian melalui fungsi aktifasi untuk menghasilkan output dari sel syaraf. Ketika semua unit pada Jaringan Syaraf telah dijalankan maka aksi dari lapisan output merupakan output dari seluruh jaringan syaraf.

# Fungsi Aktivasi



$$a = \text{hardlim}(n)$$

Hard Limit Transfer Function

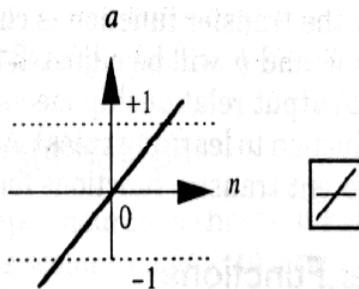


$$a = \text{hardlim}(wp + b)$$

Single-Input *hardlim* Neuron

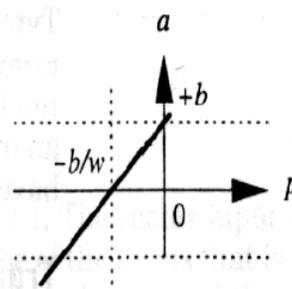
Hardlimit disebut juga fungsi threshold.

Symetric Hardlimit disebut juga fungsi threshold bipolar.



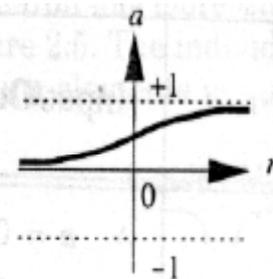
$$a = \text{purelin}(n)$$

Linear Transfer Function



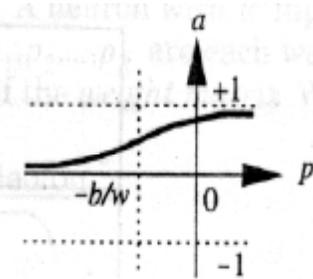
$$a = \text{purelin}(wp + b)$$

Single-Input *purelin* Neuron



$$a = \text{logsig}(n)$$

Log-Sigmoid Transfer Function



$$a = \text{logsig}(wp + b)$$

Single-Input *logsig* Neuron

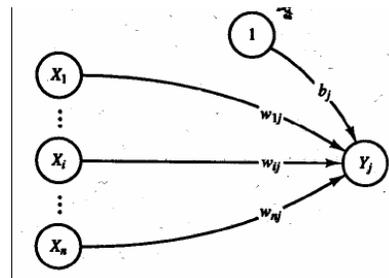
# Fungsi Aktivasi

Name	Input/Output Relation	Icon	MATLAB Function
Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$		hardlim
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$		hardlims
Linear	$a = n$		purelin
Saturating Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$		satlin
Symmetric Saturating Linear	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = 1 \quad n > 1$		satlins
Log-Sigmoid	$a = \frac{1}{1 + e^{-n}}$		logsig
Hyperbolic Tangent Sigmoid	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$		tansig
Positive Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$		poslin
Competitive	$a = 1$ neuron with max $n$ $a = 0$ all other neurons		compet

# 1

## Bias dan Threshold

- Kadang dalam jaringan ditambahkan sebuah unit masukan yang nilainya selalu = 1. Unit yang demikian disebut *bias*. Bias dapat dipandang sebagai sebuah input yang nilainya = 1. Bias berfungsi untuk mengubah nilai *threshold* menjadi = 0 (bukan = a).



- Jika melibatkan bias, maka keluaran unit penjumlah adalah  $net = b + \sum_i x_i w_i$
- Fungsi aktivasi threshold menjadi:
 
$$f(net) = \begin{cases} 1 & \text{jika } net \geq 0 \\ -1 & \text{jika } net < 0 \end{cases}$$

- Contoh:**

Suatu jaringan layer tunggal seperti gambar di atas terdiri dari 2 input  $x_1 = 0,7$  dan  $x_2 = 2,1$  dan memiliki bias. Bobot  $w_1 = 0,5$  dan  $w_2 = -0,3$  dan bobot bias  $b = 1,2$ . Tentukan keluaran neuron Y jika fungsi aktivasi adalah threshold bipolar

- Penyelesaian:**  $net = b + \sum_i x_i w_i = 1,2 + (0,7 * 0,5) + (2,1 * (-0,3)) = 0,92$

Karena  $net > 0$  maka keluaran dari jaringan  $y = f(net) = 1$

## 1

# Klasifikasi JST berdasarkan pelatihan umum

- Berdasarkan cara memodifikasi/encoding/decoding JST diklasifikasikan sbb:

		Decoding	
		Feedforward	Feedback
Encoding	Supervised	I	IV
	Unsupervised	II	III

- Supervised-Feedforward:** JST dibimbing dalam hal penyimpanan pengetahuannya serta sinyal masuk akan diteruskan tanpa umpan balik
- Unsupervised-Feedforward:** JST tidak dibimbing dalam hal penyimpanan pengetahuannya serta sinyal masuk akan diteruskan tanpa umpan balik
- Unsupervised-Feedback:** JST tidak dibimbing dalam hal penyimpanan pengetahuannya serta sinyal masuk akan diteruskan dan memberikan umpan balik
- Supervised-Feedback:** JST dibimbing dalam hal penyimpanan pengetahuannya serta sinyal masuk akan diteruskan dan memberikan umpan balik

# 1

## Klasifikasi JST berdasarkan pelatihan umum

- Konsep JST yang dibimbing (supervised): JST diberi masukan tertentu dan keluarannya ditentukan oleh pengajarnya. Dalam proses tsb, JST akan menyesuaikan bobot sinapsisnya.
- Konsep JST tanpa dibimbing (unsupervised): kebalikan dari supervised, JST secara mandiri akan mengatur keluarannya sesuai aturan yang dimiliki. Konsep JST feedforward: hasil outputnya sudah dapat diketahui sebelumnya. Konsep JST feedback: lebih bersifat dinamis, dalam hal ini kondisi jaringan akan selalu berubah sampai diperoleh keseimbangan tertentu.
- Hingga saat ini terdapat lebih dari 20 model JST. Masing2 model menggunakan arsitektur, fungsi aktivasi dan algoritma yang berbeda-beda dalam prosesnya. Taksonomi JST didasarkan pada metode pembelajaran, aplikasi dan jenis arsitekturnya
- Berdasarkan strategi pembelajaran, model JST dibagi menjadi:
  - a. Pelatihan dengan supervisi. Contoh: model Hebbian, Perceptron, Delta, ADALINE, Backpropagation, Heteroassociative Memory, Bidirectional Associative Memory (BAM)
  - b. Pelatihan tanpa supervisi. Contoh: model Hebbian, competitive, Kohonen, Learning Vector Quantization (LVQ), Hopfield

# Taksonomi JST

Berdasarkan arsitektur JST:

- a. Jaringan Layar Tunggal. Contoh: ADALINE, Hopfield, Perceptron, LVQ
- b. Jaringan Layar Jamak, Contoh: MADALINE, Backpropagation, Neocognitron
- c. Recurrent. Contoh: BAM, Hopfield, Boltzman Machine

Spesifikasi masalah dapat digunakan untuk menolong dalam penentuan arsitektur jaringan, sbb. :

1. Jumlah input jaringan = jumlah input masalah
2. Jumlah neuron dalam lapisan output = jumlah output masalah
3. Fungsi transfer lapisan output dipilih sedemikian rupa sesuai dengan spesifikasi output masalah

Aplikasi yang sudah ditemukan

- a. Klasifikasi. Model yang digunakan: ADALINE, LVQ, Backpropagation
- b. Pengenalaan Pola. Model yang digunakan: Adaptive Resonance Theory (ART), LVQ, Backpropagation
- c. Peramalan. Model yang digunakan: ADALINE, MADALINE, Backpropagation
- d. Optimisasi. Model yang digunakan: ADALINE, Hopfield, Backpropagation

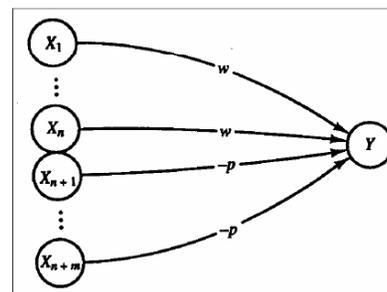
## 1

# Neuron McCulloch-Pitts (McP)

Model JST yang digunakan oleh McP merupakan model yang pertama ditemukan.

Model neuron McP memiliki karakteristik sbb:

- Fungsi aktivasinya biner
- Semua garis yang memperkuat sinyal (bobot positif) ke arah suatu neuron memiliki kekuatan (besar bobot) yang sama. Hal yang sama untuk garis yang memperlemah sinyal (bobot negatif) ke arah neuron tertentu
- Setiap neuron memiliki batas ambang (threshold) yang sama. Apabila total input ke neuron tersebut melebihi threshold, maka neuron akan meneruskan sinyal



model neuron McP

- Neuron Y menerima sinyal dari  $(n+m)$  buah neuron  $x^1, x^2, \dots, x^n, x^{n+1}, \dots, x^{n+m}$
- $n$  buah penghubung dengan dari  $x^1, x^2, \dots, x^n$  ke Y merupakan garis yang memperkuat sinyal (bobot positif), sedangkan  $m$  buah penghubung dari  $x^{n+1}, \dots, x^{n+m}$  ke Y merupakan garis yang memperlemah sinyal (bobot negatif).

1

## Neuron McCulloch-Pitts (McP)

- Semua penghubung dari  $x^1, x^2, \dots, x^n$  ke  $Y$  memiliki bobot yang sama. Hal yang sama dengan penghubung dari  $x^{n+1}, \dots, x^{n+m}$  ke  $Y$  memiliki bobot yang sama. Namun jika ada neuron lain katakan  $Y_2$ , maka bobot  $x^1$  ke  $Y_1$  boleh berbeda dengan bobot dari  $x^2$  ke  $Y_2$ .
- Fungsi aktivasi neuron  $Y$  adalah
$$f(\text{net}) = \begin{cases} 1 & \text{jika } \text{net} \geq a \\ 0 & \text{jika } \text{net} < a \end{cases}$$
- Bobot tiap garis tidak ditentukan dengan proses pelatihan, tetapi dengan metode analitik. Beberapa contoh berikut memaparkan bagaimana neuron McP digunakan untuk memodelkan fungsi logika sederhana.

## 1

# Neuron McCulloch-Pitts (McP)

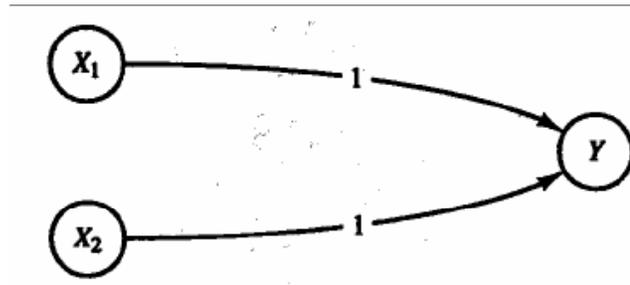
- Contoh: Fungsi logika "AND" dengan 2 masukan  $x_1$  dan  $x_2$  akan memiliki keluaran  $Y = 1$  jika dan hanya jika kedua masukan bernilai 1.

Tabel kebenaran:

$x_1$	$x_2$	$Y$
1	1	1
1	0	0
0	1	0
0	0	0

- Buatlah model neuron McP untuk menyatakan fungsi logika AND
- Penyelesaian

Model neuron fungsi AND tampak pada gambar di bawah ini. Bobot tiap garis adalah = 1 dan fungsi aktivasi memiliki nilai threshold = 2



## 1

# Neuron McCulloch-Pitts (McP)

- Untuk semua kemungkinan masukan, nilai aktivasi tampak pada tabel berikut:

$x_1$	$x_2$	$net = \sum_{i=1}^2 x_i w_i$	$f(net) = \begin{cases} 1 & \text{jika } net \geq 2 \\ 0 & \text{jika } net < 2 \end{cases}$
1	1	$1*1 + 1*1=2$	1
1	0	$1*1 + 0*1=1$	0
0	1	$0*1 + 1*0=1$	0
0	0	$0*1 + 0*1=0$	0

- Tampak bahwa keluaran jaringan tepat sama dengan tabel logika AND. Berarti jaringan dapat dengan tepat merepresentasikan fungsi AND
- Besarnya nilai threshold dapat diganti menjadi suatu bias dengan nilai yang sama. Dengan menggunakan nilai bias, batas garis pemisah ditentukan dari persamaan

$$net = b + \sum_i x_i w_i = 0$$

$$b + x_1 w_1 + x_2 w_2 = 0 \text{ atau } x_2 = -w_1 x_1 / w_2 - b / w_2$$

Apabila garis pemisahnya diambil dengan persamaan  $x_1 + x_2 = 2$ , maka berarti

$$-w_1 / w_2 = -1 \text{ dan } -b / w_2 = 2.$$

Ada banyak  $w_1$ ,  $w_2$  dan  $b$  yang memenuhi persamaan tersebut, salah satunya

adalah  $w_1 = w_2 = 1$  dan  $b = -2$ , seperti penyelesaian contoh di atas.

# LATIHAN

1. **Buatlah model neuron McP untuk menyatakan fungsi logika OR**
2. **Buatlah model neuron McP untuk menyatakan fungsi logika XOR**
3. **Buatlah model neuron McP untuk menyatakan fungsi logika  $x_1$  AND NOT  $x_2$**