

BAB II

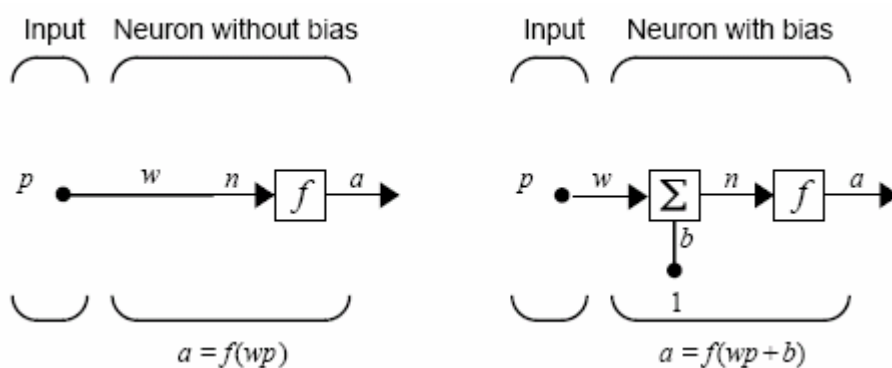
MODEL NEURON DAN ARSITEKTUR JARINGAN

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

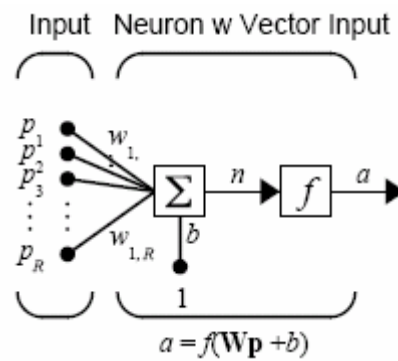
- ▶ Himpunan unit2 yang dihubungkan dengan jalur koneksi. Jalur tersebut memiliki bobot yang berbeda-beda. Bobot yang bernilai positif akan memperkuat sinyal dan yang bernilai negatif akan memperlemah sinyal yang dibawahnya. Jumlah, struktur dan pola hubungan antar unit2 tersebut akan menentukan "ARSISTEKTUR JARINGAN" (dan juga model jaringan yang terbentuk
- ▶ Suatu unit penjumlah yang akan menjumlahkan input2 sinyal yang sudah dikalikan dengan bobot. Misalkan x_1, x_2, \dots, x_m adalah unit-2 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 $u_j = x_1w_{j1} + x_2w_{j2} + \dots + x_mw_{jm}$
- ▶ Fungsi aktivasi yang akan menentukan apakah sinyal dari input neuron akan diteruskan ke neuron lain atautkah tidak.

Contoh:

Single-Input Neuron



Multiple-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

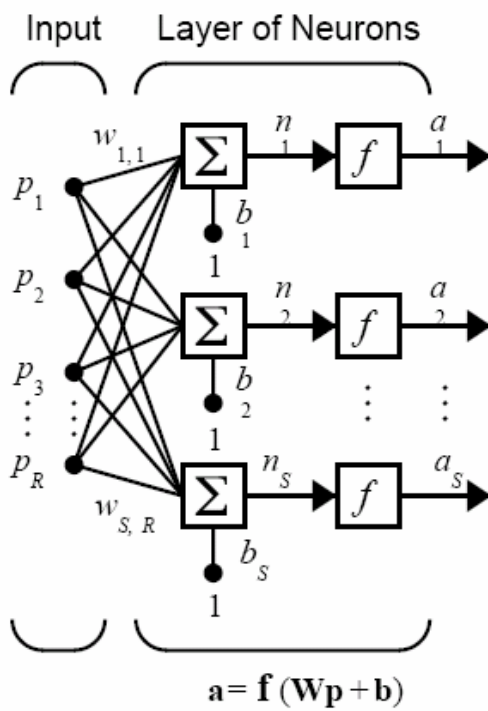
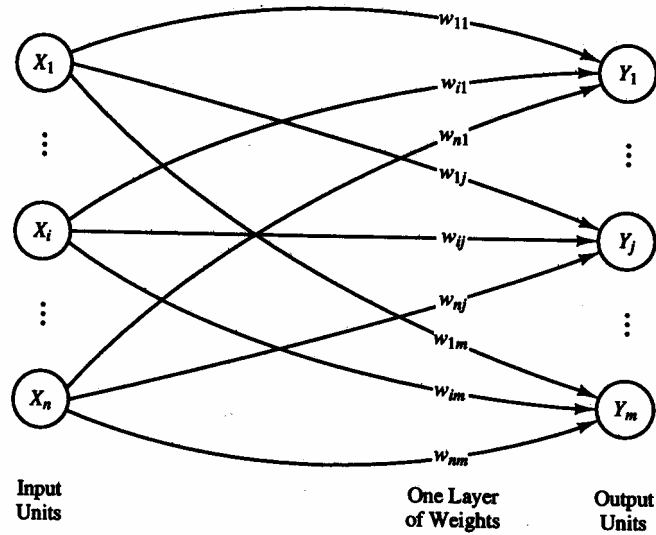
$$n = \mathbf{Wp} + b$$

Output neuron (a) :

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

II.1 Arsitektur Jaringan

II.1.1 Neuron selapis



Dengan

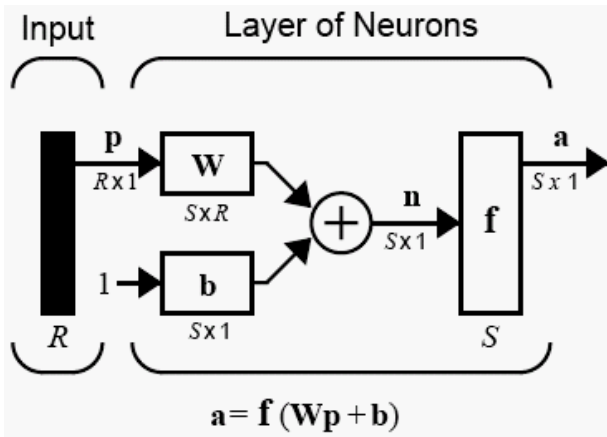
R sebagai jumlah elemen vektor input dan

S sebagai jumlah neuron dalam lapisan

Elemen vektor input jaringan dipresentasikan dalam bentuk matriks :

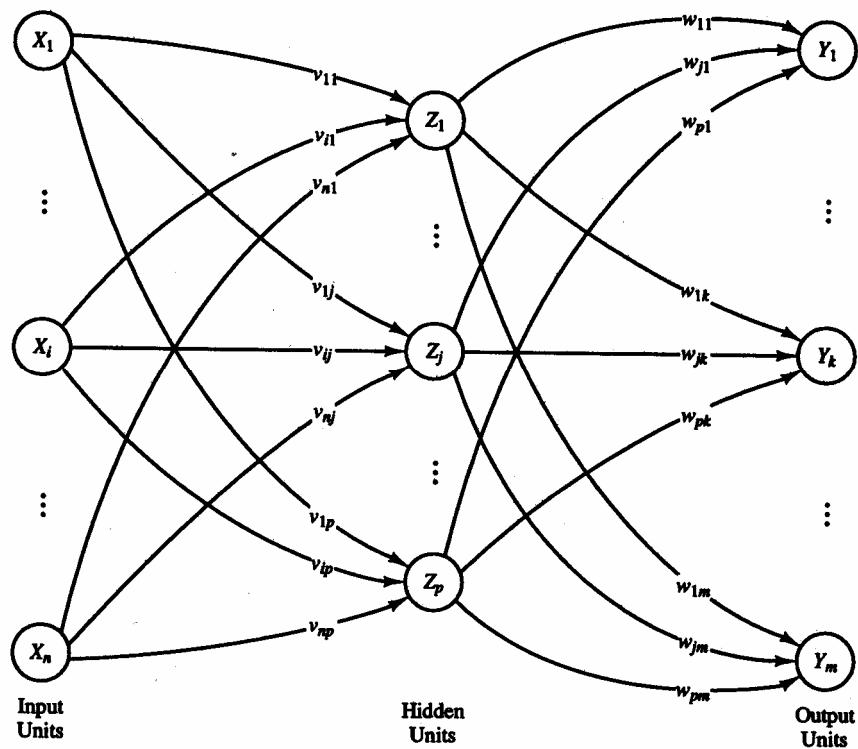
$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \vdots & \vdots & \ddots & \vdots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix}$$

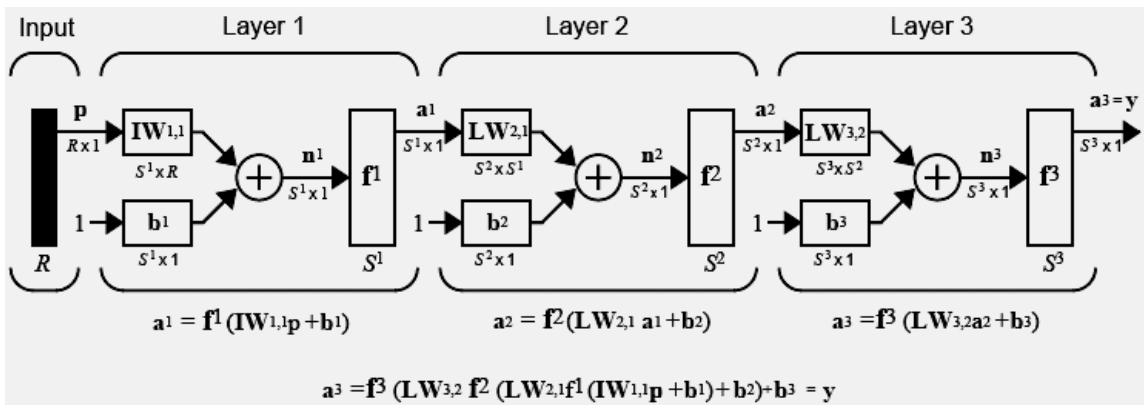
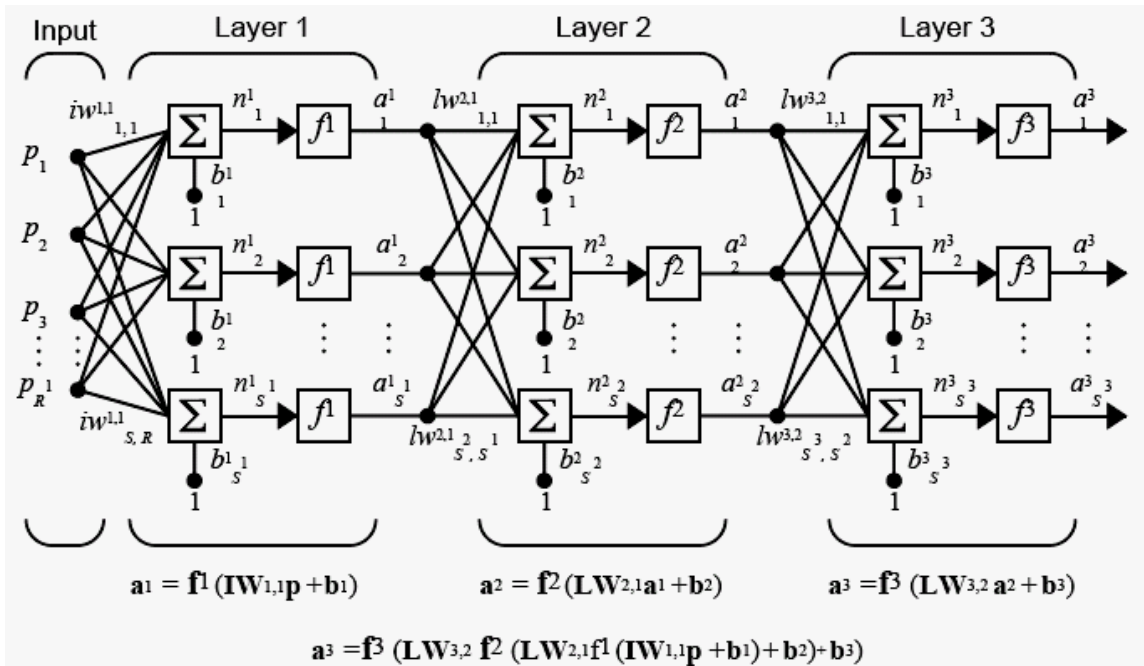
Selapis neuron dapat digambarkan dengan notasi sbb. :



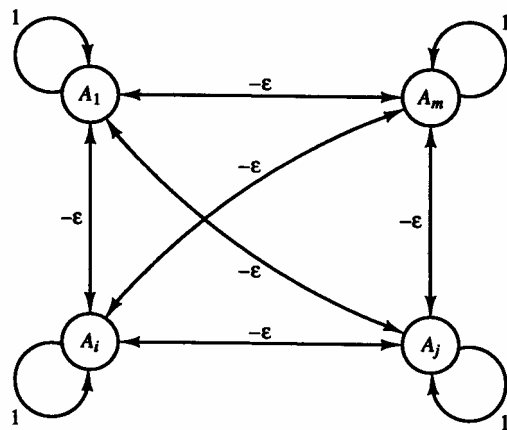
Dengan
 R sebagai jumlah elemen vektor input dan
 S sebagai jumlah neuron dalam lapisan

II.1.2 Neuron beberapa lapis

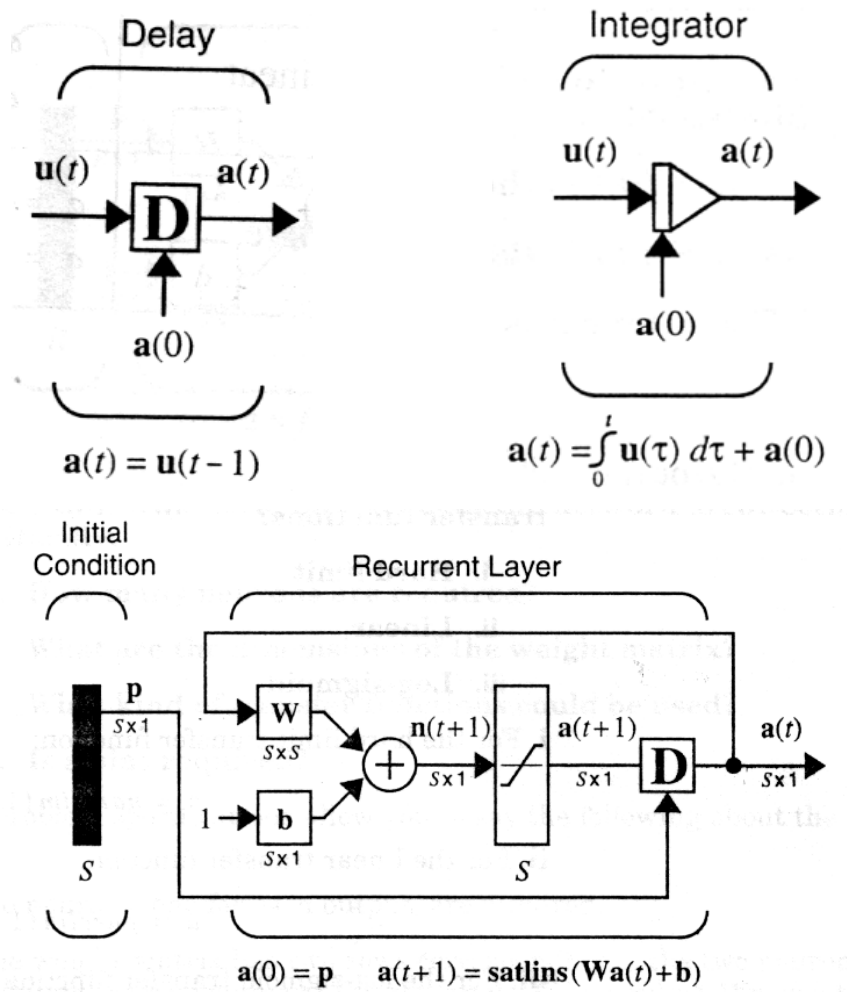




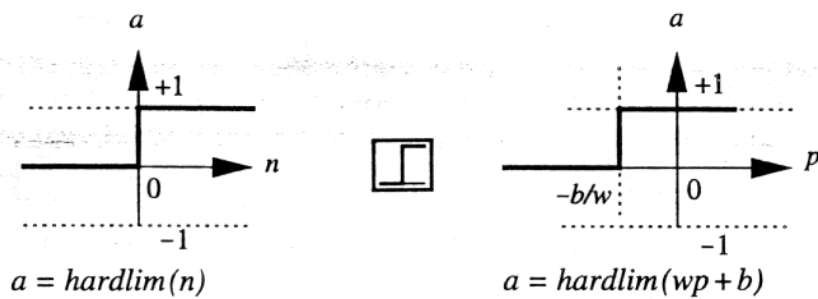
II.1.3 Jaringan Kompetitif



II.1.4 Jaringan *recurrent*



II.2 Fungsi Aktivasi

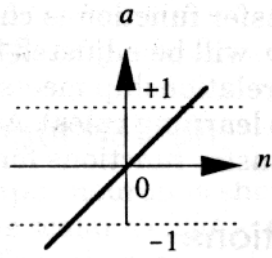


Hard Limit Transfer Function

Single-Input *hardlim* Neuron

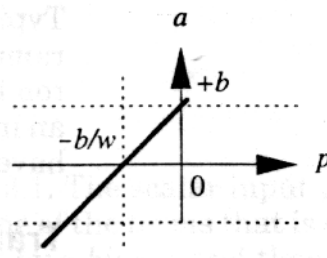
Hardlimit disebut juga fungsi threshold.

Symmetric Hardlimit disebut juga fungsi threshold bipolar



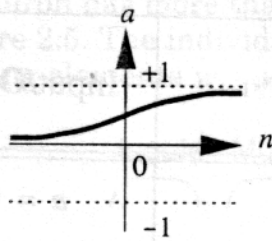
$$a = \text{purelin}(n)$$

Linear Transfer Function



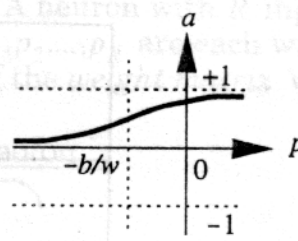
$$a = \text{purelin}(wp + b)$$

Single-Input *purelin* Neuron



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

Log-Sigmoid Transfer Function



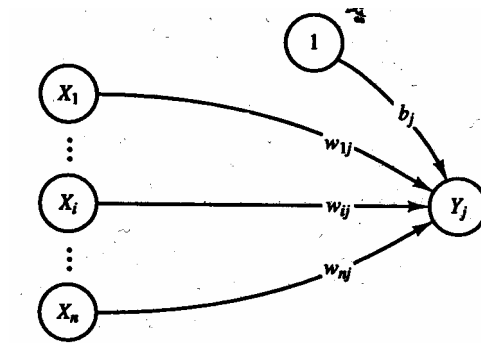
$$a = \text{logsig}(wp + b)$$

Single-Input *logsig* Neuron

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	$a = 1 \quad \text{neuron with max } n$ $a = 0 \quad \text{all other neurons}$		compet

II.3 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

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

Fungsi aktivasi threshold menjadi:

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

Contoh:

Suatu jaringan layar 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:

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

Karena $\text{net} > 0$ maka keluaran dari jaringan $y = f(\text{net}) = 1$

II.4 Klasifikasi JST berdasarkan pelatihan umum

Berdasarkan cara memodifikasi/encoding/decoding JST diklasifikasikan sbb:

		Decoding	
		Feedforward	Feedback
Encoding	Supervised	I	IV
	Unsupervised	II	III

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

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 samapi diperoleh keseimbangan tertentu.

II.5 Taksonomi JST

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

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

Petunjuk untuk memilih arsitektur jaringan

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

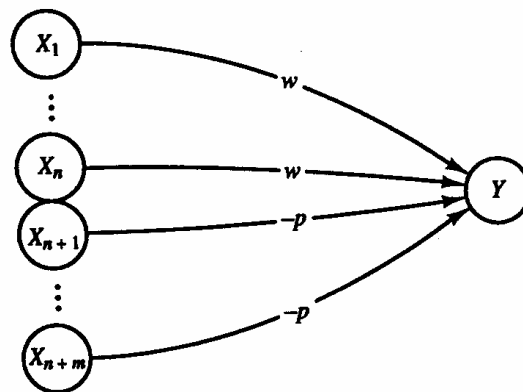
Aplikasi yang sudah ditemukan

- a. Klasifikasi. Model yang digunakan: ADALINE, LVQ, Backpropagation
- b. Pengenalan 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

II.6 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 bata 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).

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 (dan coba-2). Beberapa contoh berikut memaparkan bagaimana neuron McP digunakan untuk memodelkan fungsi logika sederhana.

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.

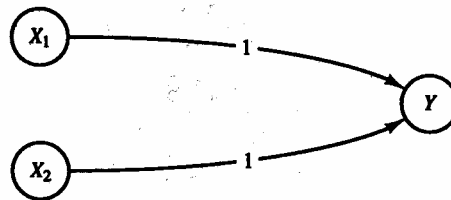
Tabel kebenaran:

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



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

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

$$\text{net} = b + \sum_i x_i w_i = 0$$

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

Apabila garis pemisahnya diambil dengan persamaan $x_1 + x_2 = 2$, maka berarti $-w_1/w_2 = -1$ 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