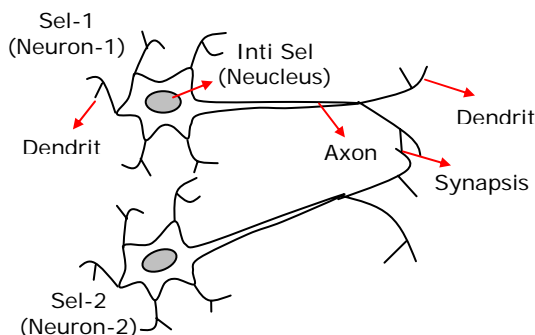


BAB VIII

JARINGAN SYARAF TIRUAN

A. OTAK MANUSIA

Otak manusia berisi berjuta-juta sel syaraf yang bertugas untuk memproses informasi. Tiap-tiap sel bekerja seperti suatu prosesor sederhana. Masing-masing sel tersebut saling berinteraksi sehingga mendukung kemampuan kerja otak manusia.



Gambar 8.1 Susunan syaraf manusia.

Gambar 8.1 menunjukkan susunan syaraf pada manusia. Setiap sel syaraf (neuron) akan memiliki satu inti sel, inti sel ini nanti yang akan bertukan untuk melakukan pemrosesan informasi. Informasi yang datang akan diterima oleh dendrit. Selain menerima informasi, dendrit juga menyertai axon sebagai keluaran dari suatu pemrosesan informasi. Informasi hasil olahan ini akan menjadi masukan bagi neuron lain yang mana antar dendrit kedua sel tersebut dipertemukan dengan synapsis. Informasi yang dikirimkan antar neuron ini berupa rangsangan yang dilewatkan melalui dendrit. Informasi yang datang dan diterima oleh dendrit akan dijumlahkan dan dikirim melalui axon ke dendrit akhir yang bersentuhan dengan dendrit dari neuron yang lain. Informasi ini akan diterima oleh neuron lain jika memenuhi batasan tertentu, yang sering dikenal dengan nama nilai ambang (*threshold*). Pada kasus ini, neuron tersebut dikatakan teraktivasi. Hubungan antar neuron terjadi secara adaptif, artinya struktur hubungan tersebut terjadi secara dinamis. Otak manusia selalu memiliki kemampuan untuk belajar dengan melakukan adaptasi.

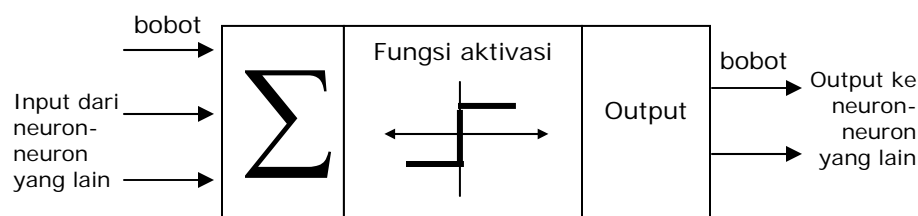
B. SEJARAH

Mulai dari ditemukannya, jaringan syaraf tiruan telah mengalami tahap-tahap perkembangan, antara lain:

- ✧ Pada tahun 1943, McCulloch dan Pitts merancang model formal yang pertama kali sebagai perhitungan dasar neuron.
- ✧ Pada tahun 1949, Hebb menyatakan bahwa informasi dapat disimpan dalam koneksi-koneksi dan mengusulkan adanya skema pembelajaran untuk memperbaiki koneksi-koneksi antar neuron tersebut.
- ✧ Pada tahun 1958, Rosenblatt mengembangkan konsep dasar tentang perceptron untuk klasifikasi pola.
- ✧ Pada tahun 1982, Kohonen mengembangkan metode pembelajaran jaringan syaraf yang tidak terawasi (*unsupervised learning*) untuk pemetaan.
- ✧ Pada tahun 1982, Grossberg mengembangkan teori jaringan yang diinspirasi oleh perkembangan psikologi. Bersama Carpenter, mereka mengenalkan sejumlah arsitektur jaringan, antara lain: Adaptive Resonance Theory (ART), ART2, dan ART3.
- ✧ Pada tahun 1982, Hopfield mengembangkan jaringan syaraf *recurrent* yang dapat digunakan untuk menyimpan informasi dan optimasi.

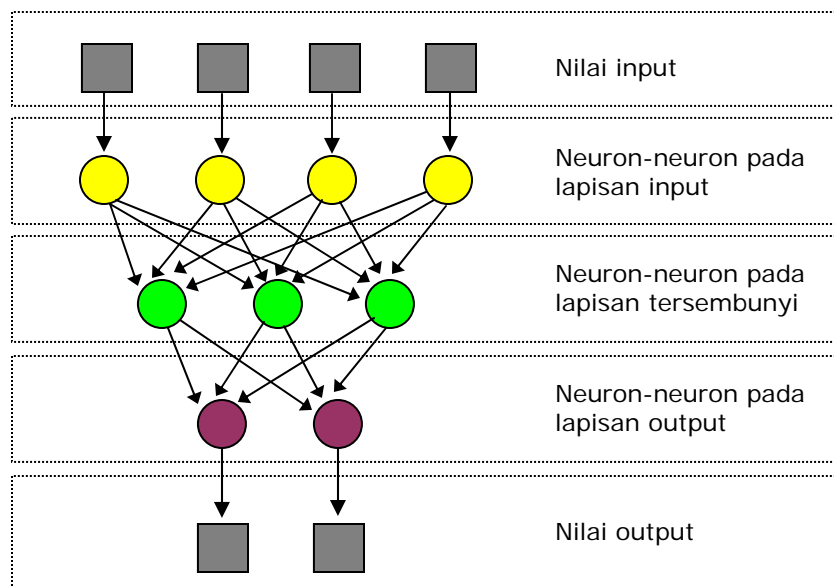
C. KOMPONEN JARINGAN SYARAF

Ada beberapa tipe jaringan syaraf, namun demikian, hampir semuanya memiliki komponen-komponen yang sama. Seperti halnya otak manusia, jaringan syaraf juga terdiri-dari beberapa **neuron**, dan ada hubungan antara neuron-neuron tersebut. Neuron-neuron tersebut akan mentransformasikan informasi yang diterima melalui sambungan keluarnya menuju ke neuron-neuron yang lain. Pada jaringan syaraf, hubungan ini dikenal dengan nama **bobot**. Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tersebut. Gambar 8.2 menunjukkan struktur neuron pada jaringan syaraf.



Jika kita lihat, neuron buatan ini sebenarnya mirip dengan sel neuron biologis. Neuron-neuron buatan tersebut bekerja dengan cara yang sama pula dengan neuron-neuron biologis. Informasi (disebut dengan: **input**) akan dikirim ke neuron dengan bobot kedatangan tertentu. Input ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan ini kemudian akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui **fungsi aktivasi** setiap neuron. Apabila input tersebut melewati suatu nilai ambang tertentu, maka neuron tersebut akan diaktifkan, tapi kalau tidak, maka neuron tersebut tidak akan diaktifkan. Apabila neuron tersebut diaktifkan, maka neuron tersebut akan mengirimkan **output** melalui bobot-bobot outputnya ke semua neuron yang berhubungan dengannya. Demikian seterusnya.

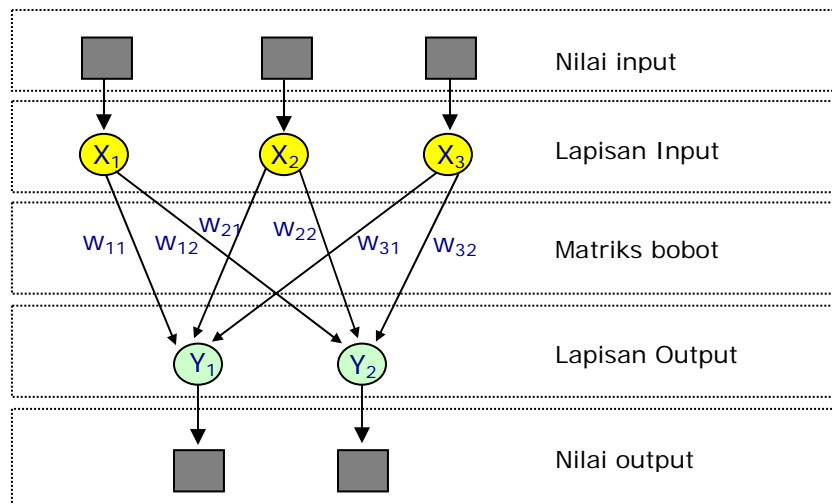
Pada jaringan syaraf, neuron-neuron akan dikumpulkan dalam lapisan-lapisan (*layer*) yang disebut dengan lapisan neuron (*neuron layers*). Biasanya neuron-neuron pada satu lapisan akan dihubungkan dengan lapisan-lapisan sebelum dan sesudahnya (kecuali lapisan input dan lapisan output). Informasi yang diberikan pada jaringan syaraf akan dirambatkan lapisan ke lapisan, mulai dari lapisan input sampai ke lapisan output melalui lapisan yang lainnya, yang sering dikenal dengan nama lapisan tersembunyi (*hidden layer*). Tergantung pada **algoritma pembelajarannya**, bisa jadi informasi tersebut akan dirambatkan secara mundur pada jaringan. Gambar 8.3 menunjukkan jaringan syaraf dengan 3 lapisan.



D. ARSITEKTUR JARINGAN

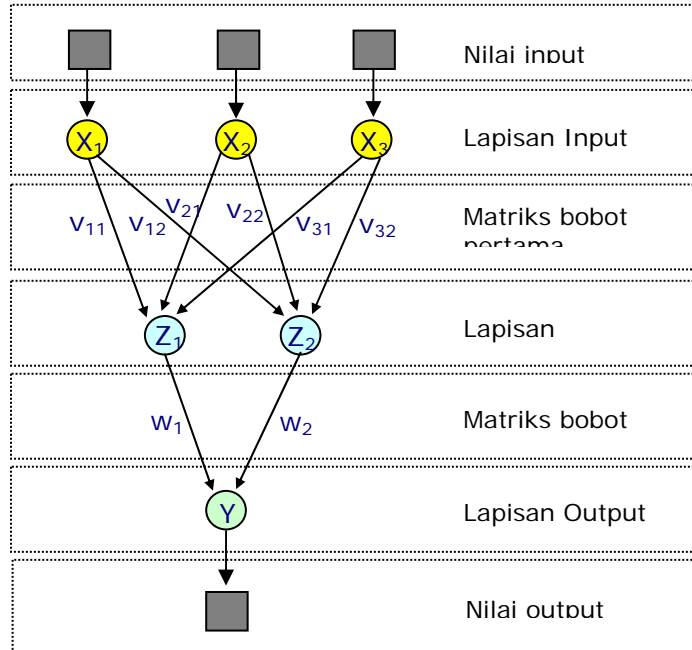
a. Jaringan dengan lapisan tunggal (*single layer net*)

Jaringan dengan lapisan tunggal hanya memiliki satu lapisan dengan bobot-bobot terhubung. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi output tanpa harus melalui lapisan tersembunyi (Gambar 8.4). Pada Gambar 8.4 tersebut, lapisan input memiliki 3 neuron, yaitu X_1 , X_2 dan X_3 . Sedangkan pada lapisan output memiliki 2 neuron yaitu Y_1 dan Y_2 . Neuron-neuron pada kedua lapisan saling berhubungan. Seberapa besar hubungan antara 2 neuron ditentukan oleh bobot yang bersesuaian. Semua unit input akan dihubungkan dengan setiap unit output.



b. Jaringan dengan banyak lapisan (*multilayer net*)

Jaringan dengan banyak lapisan memiliki 1 atau lebih lapisan yang terletak diantara lapisan input dan lapisan output (memiliki 1 atau lebih lapisan tersembunyi), seperti terlihat pada Gambar 8.5. Umumnya, ada lapisan bobot-bobot yang terletak antara 2 lapisan yang bersebelahan. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit daripada lapisan dengan lapisan tunggal, tentu saja dengan pembelajaran yang lebih rumit. Namun demikian, pada banyak kasus, pembelajaran pada jaringan dengan banyak lapisan ini lebih sukses dalam menyelesaikan masalah.



E. FUNGSI AKTIVASI

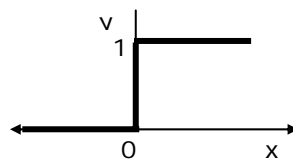
Ada beberapa fungsi aktivasi yang sering digunakan dalam jaringan syaraf tiruan, antara lain:

a. Fungsi Undak Biner (*Hard Limit*)

Jaringan dengan lapisan tunggal sering menggunakan fungsi undak (*step function*) untuk mengkonversikan input dari suatu variabel yang bernilai kontinu ke suatu output biner (0 atau 1) (Gambar 8.7).

Fungsi undak biner (*hard limit*) dirumuskan sebagai:

$$y = \begin{cases} 0, & \text{jika } x \leq 0 \\ 1, & \text{jika } x > 0 \end{cases}$$

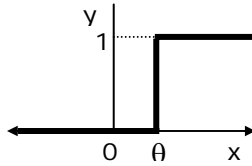


b. Fungsi Undak Biner (*Threshold*)

Fungsi undak biner dengan menggunakan nilai ambang sering juga disebut dengan nama fungsi nilai ambang (*threshold*) atau fungsi Heaviside (Gambar 8.8).

Fungsi undak biner (dengan nilai ambang θ) dirumuskan sebagai:

$$y = \begin{cases} 0, & \text{jika } x < \theta \\ 1, & \text{jika } x \geq \theta \end{cases}$$

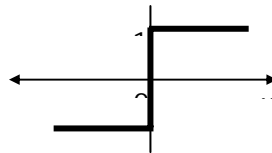


c. Fungsi Bipolar (*Symmetric Hard Limit*)

Fungsi bipolar sebenarnya hampir sama dengan fungsi undak biner, hanya saja output yang dihasilkan berupa 1, 0 atau -1 (Gambar 8.9).

Fungsi *Symmetric Hard Limit* dirumuskan sebagai:

$$y = \begin{cases} 1, & \text{jika } x > 0 \\ 0, & \text{jika } x = 0 \\ -1, & \text{jika } x < 0 \end{cases}$$

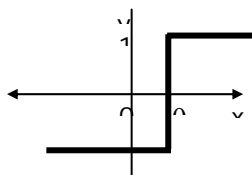


d. Fungsi Bipolar (dengan *threshold*)

Fungsi bipolar sebenarnya hampir sama dengan fungsi undak biner dengan *threshold*, hanya saja output yang dihasilkan berupa 1, 0 atau -1 (Gambar 8.10).

Fungsi bipolar (dengan nilai ambang θ) dirumuskan sebagai:

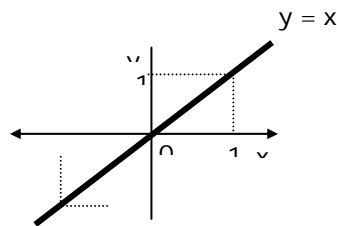
$$y = \begin{cases} 1, & \text{jika } x \geq \theta \\ -1, & \text{jika } x < \theta \end{cases}$$



e. Fungsi Linear (identitas)

Fungsi linear memiliki nilai output yang sama dengan nilai inputnya (Gambar 8.11).

Fungsi linear dirumuskan sebagai:

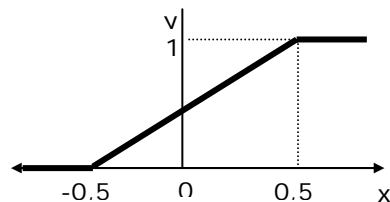


f. Fungsi Saturating Linear

Fungsi ini akan bernilai 0 jika inputnya kurang dari $-\frac{1}{2}$, dan akan bernilai 1 jika inputnya lebih dari $\frac{1}{2}$. Sedangkan jika nilai input terletak antara $-\frac{1}{2}$ dan $\frac{1}{2}$, maka outputnya akan bernilai sama dengan nilai input ditambah $\frac{1}{2}$ (Gambar 8.12).

Fungsi *saturating linear* dirumuskan sebagai:

$$y = \begin{cases} 1; & \text{jika } x \geq 0,5 \\ x + 0,5; & \text{jika } -0,5 \leq x \leq 0,5 \\ 0; & \text{jika } x \leq -0,5 \end{cases}$$

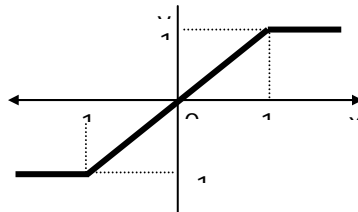


g. Fungsi Symetric Saturating Linear

Fungsi ini akan bernilai -1 jika inputnya kurang dari -1 , dan akan bernilai 1 jika inputnya lebih dari 1. Sedangkan jika nilai input terletak antara -1 dan 1, maka outputnya akan bernilai sama dengan nilai inputnya (Gambar 8.13).

Fungsi *symetric saturating linear* dirumuskan sebagai:

$$y = \begin{cases} 1; & \text{jika } x \geq 1 \\ x; & \text{jika } -1 \leq x \leq 1 \\ -1; & \text{jika } x \leq -1 \end{cases}$$



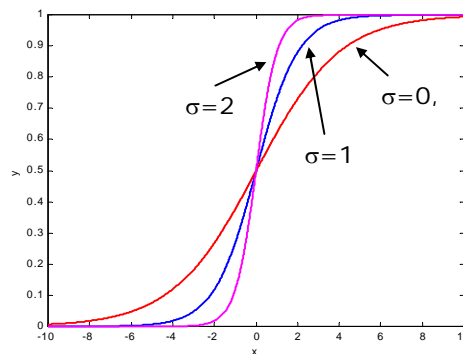
h. Fungsi Sigmoid Biner.

Fungsi ini digunakan untuk jaringan syaraf yang dilatih dengan menggunakan metode *backpropagation*. Fungsi sigmoid biner memiliki nilai pada range 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan untuk jaringan syaraf yang membutuhkan nilai output yang terletak pada interval 0 sampai 1. Namun, fungsi ini bisa juga digunakan oleh jaringan syaraf yang nilai outputnya 0 atau 1 (Gambar 8.14).

Fungsi sigmoid biner dirumuskan sebagai:

$$y = f(x) = \frac{1}{1 + e^{-\sigma x}}$$

dengan: $f'(x) = \sigma f(x)[1 - f(x)]$



i. Fungsi Sigmoid Bipolar

Fungsi sigmoid bipolar hampir sama dengan fungsi sigmoid biner, hanya saja output dari fungsi ini memiliki range antara 1 sampai -1 (Gambar 8.15).

Fungsi sigmoid bipolar dirumuskan sebagai:

$$y = f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

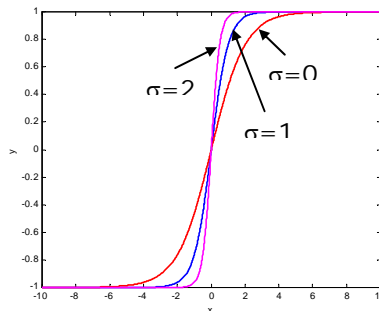
$$\text{dengan: } f'(x) = \frac{\sigma}{2} [1 + f(x)][1 - f(x)]$$

Fungsi ini sangat dekat dengan fungsi *hyperbolic tangent*. Keduanya memiliki range antara -1 sampai 1. Untuk fungsi *hyperbolic tangent*, dirumuskan sebagai:

$$y = f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$\text{atau } y = f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

$$\text{dengan: } f'(x) = [1 + f(x)][1 - f(x)]$$



F. PROSES PEMBELAJARAN

Pada otak manusia, informasi yang dilewatkan dari satu neuron ke neuron yang lainnya berbentuk rangsangan listrik melalui dendrit. Jika rangsangan tersebut diterima oleh suatu neuron, maka neuron tersebut akan membangkitkan output ke semua neuron yang berhubungan dengannya sampai informasi tersebut sampai ke tujuannya yaitu terjadinya suatu reaksi. Jika rangsangan yang diterima terlalu halus, maka output yang dibangkitkan oleh neuron tersebut tidak akan direspon. Tentu saja sangatlah sulit untuk memahami bagaimana otak manusia bisa belajar. Selama proses pembelajaran, terjadi perubahan yang cukup berarti pada bobot-bobot yang menghubungkan antar neuron. Apabila ada rangsangan

yang sama dengan rangsangan yang telah diterima oleh neuron, maka neuron akan memberikan reaksi dengan cepat. Namun apabila kelak ada rangsangan yang berbeda dengan apa yang telah diterima oleh neuron, maka neuron akan segera beradaptasi untuk memberikan reaksi yang sesuai.

Jaringan syaraf akan mencoba untuk mensimulasikan kemampuan otak manusia untuk belajar. Jaringan syaraf tiruan juga tersusun atas neuron-neuron dan dendrit. Tidak seperti model biologis, jaringan syaraf memiliki struktur yang tidak dapat diubah, dibangun oleh sejumlah neuron, dan memiliki nilai tertentu yang menunjukkan seberapa besar koneksi antara neuron (yang dikenal dengan nama bobot). Perubahan yang terjadi selama proses pembelajaran adalah perubahan nilai bobot. Nilai bobot akan bertambah, jika informasi yang diberikan oleh neuron yang bersangkutan tersampaikan, sebaliknya jika informasi tidak disampaikan oleh suatu neuron ke neuron yang lain, maka nilai bobot yang menghubungkan keduanya akan dikurangi. Pada saat pembelajaran dilakukan pada input yang berbeda, maka nilai bobot akan diubah secara dinamis hingga mencapai suatu nilai yang cukup seimbang. Apabila nilai ini telah tercapai mengindikasikan bahwa tiap-tiap input telah berhubungan dengan output yang diharapkan.

a. Pembelajaran terawasi (*supervised learning*)

Metode pembelajaran pada jaringan syaraf disebut terawasi jika output yang diharapkan telah diketahui sebelumnya.

Contoh: andaikan kita memiliki jaringan syaraf yang akan digunakan untuk mengenali pasangan pola, misalkan pada operasi AND:

Input	target
0 0	0
0 1	0
1 0	0
1 1	1

Pada proses pembelajaran, satu pola input akan diberikan ke satu neuron pada lapisan input. Pola ini akan dirambatkan di sepanjang jaringan syaraf hingga sampai ke neuron pada lapisan output. Lapisan output ini akan membangkitkan pola output yang nantinya akan dicocokkan dengan pola output targetnya. Apabila terjadi perbedaan antara pola output hasil pembelajaran dengan pola target, maka disini akan muncul error. Apabila nilai error ini masih cukup besar, mengindikasikan bahwa masih perlu dilakukan lebih banyak pembelajaran lagi.

b. Pembelajaran tak terawasi (*unsupervised learning*)

Pada metode pembelajaran yang tak terawasi ini tidak memerlukan target output. Pada metode ini, tidak dapat ditentukan hasil yang seperti apakah yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu range tertentu tergantung pada nilai input yang diberikan. Tujuan pembelajaran ini adalah mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu. Pembelajaran ini biasanya sangat cocok untuk pengelompokan (klasifikasi) pola.

G. PEMBELAJARAN TERAWASI (SUPERVISED LEARNING)

1.. Hebb Rule

Hebb rule adalah metode pembelajaran yang paling sederhana. Pada metode ini pembelajaran dilakukan dengan cara memperbaiki nilai bobot sedemikian rupa sehingga jika ada 2 neuron yang terhubung, dan keduanya pada kondisi 'hidup' (on) pada saat yang sama, maka bobot antara keduanya dinaikkan. Apabila data direpresentasikan secara bipolar, maka perbaikan bobotnya adalah:

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

dengan:

w_i : bobot data input ke-i;

x_i : input data ke-i.

y : output data.

Misalkan kita gunakan pasangan vektor input s dan vektor output sebagai pasangan vektor yang akan dilatih. Sedangkan vektor yang hendak digunakan untuk testing adalah vektor x .

Algoritma

0. Inisialisasi semua bobot:

$$w_{ij} = 0; \text{ dengan } i=1,2,\dots,n; \text{ dan } j=1,2,\dots,m.$$

1. Untuk setiap pasangan input-output (s - t), lakukan langkah-langkah sebagai berikut:

a. Set input dengan nilai sama dengan vektor input:

$$x_i = s_i; \quad (i=1,2,\dots,n)$$

b. Set output dengan nilai sama dengan vektor output:

$$y_j = t_j; \quad (j=1,2,\dots,m)$$

c. Perbaiki bobot:

$$w_{ij}(\text{baru}) = w_{ij}(\text{lama}) + x_i * y_j;$$

$$(i=1,2,\dots,n; \text{ dan } j=1,2,\dots,m)$$

dengan catatan bahwa nilai bias selalu 1.

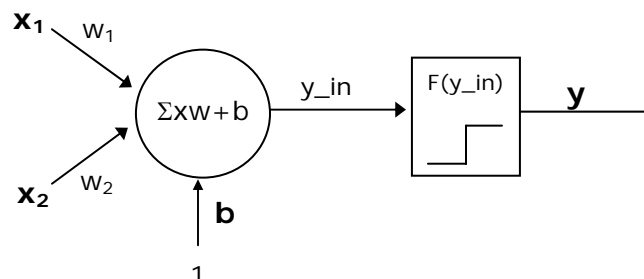
Contoh 8.1:

Misalkan kita ingin membuat jaringan syaraf untuk melakukan pembelajaran terhadap fungsi OR dengan input dan target bipolar sebagai berikut:

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

Bobot awal dan bobot bias kita set=0.

Arsitektur jaringan untuk contoh 8.1.



$$X = \begin{matrix} -1 & -1 \\ -1 & 1 \\ 1 & -1 \\ 1 & 1 \end{matrix}$$

$$T = \begin{matrix} -1 \\ 1 \\ 1 \\ 1 \end{matrix}$$

Bobot awal =

$$w = \begin{matrix} 0 \\ 0 \end{matrix}$$

$$b = 0$$

Perubahan bobot:

Data ke-1

$$\begin{aligned} w_1 &= 0 + 1 = 1 \\ w_2 &= 0 + 1 = 1 \\ b &= 0 - 1 = -1 \end{aligned}$$

Data ke-2

$$\begin{aligned} w_1 &= 1 - 1 = 0 \\ w_2 &= 1 + 1 = 2 \\ b &= -1 + 1 = 0 \end{aligned}$$

Data ke-3

$$\begin{aligned} w_1 &= 0 + 1 = 1 \\ w_2 &= 2 - 1 = 1 \\ b &= 0 + 1 = 1 \end{aligned}$$

Data ke-4

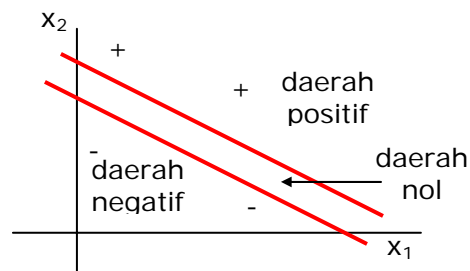
$$\begin{aligned} w_1 &= 1 + 1 = 2 \\ w_2 &= 1 + 1 = 2 \\ b &= 1 + 1 = 2 \end{aligned}$$

Kita melakukan pengetesan terhadap salah satu data yang ada, misal kita ambil $x = [-1 \ -1]$.

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

2. Perceptron

Perceptron juga termasuk salah satu bentuk jaringan syaraf yang sederhana. Perceptron biasanya digunakan untuk mengklasifikasikan suatu tipe pola tertentu yang sering dikenal dengan pemisahan secara linear. Pada dasarnya, perceptron pada jaringan syaraf dengan satu lapisan memiliki bobot yang bisa diatur dan suatu nilai ambang (*threshold*). Algoritma yang digunakan oleh aturan perceptron ini akan mengatur parameter-parameter bebasnya melalui proses pembelajaran. Nilai *threshold* (θ) pada fungsi aktivasi adalah non negatif. Fungsi aktivasi ini dibuat sedemikian rupa sehingga terjadi pembatasan antara daerah positif dan daerah negatif (Gambar 8.17).



Garis pemisah antara daerah positif dan daerah nol memiliki pertidaksamaan:

$$w_1x_1 + w_2x_2 + b > \theta$$

Sedangkan garis pemisah antara daerah negatif dengan daerah nol memiliki pertidaksamaan:

$$w_1x_1 + w_2x_2 + b < -\theta$$

Misalkan kita gunakan pasangan vektor input s dan vektor output sebagai pasangan vektor yang akan dilatih.

Algoritma:

0. Inisialisasi semua bobot dan bias:

(untuk sederhananya set semua bobot dan bobot bias sama dengan nol).

Set *learning rate*: α ($0 < \alpha \leq 1$).

(untuk sederhananya set sama dengan 1).

1. Selama kondisi berhenti bernilai *false*, lakukan langkah-langkah sebagai berikut:

(i). Untuk setiap pasangan pembelajaran s-t, kerjakan:

a. Set input dengan nilai sama dengan vektor input:

$$x_i = s_i;$$

b. Hitung respon untuk unit output:

$$y_{in} = b + \sum_i x_i w_i$$

$$y = \begin{cases} 1, & \text{jika } y_{in} > \theta \\ 0, & \text{jika } -\theta \leq y_{in} \leq \theta \\ -1, & \text{jika } y_{in} < -\theta \end{cases}$$

c. Perbaiki bobot dan bias jika terjadi *error*:

Jika $y \neq t$ maka:

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

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

jika tidak, maka:

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

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

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

Algoritma di atas bisa digunakan baik untuk input biner maupun bipolar, dengan θ tertentu, dan bias yang dapat diatur. Pada algoritma tersebut bobot-bobot yang diperbaiki hanyalah bobot-bobot yang berhubungan dengan input yang aktif ($x_i \neq 0$) dan bobot-bobot yang tidak menghasilkan nilai y yang benar.

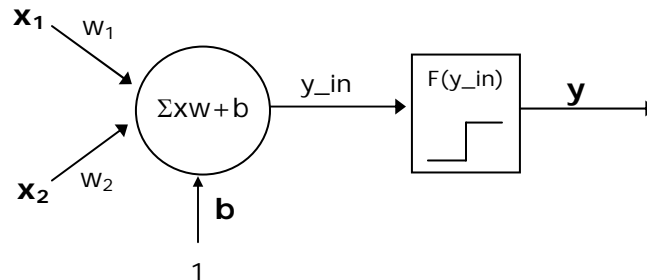
Contoh 8.2:

Misalkan kita ingin membuat jaringan syaraf untuk melakukan pembelajaran terhadap fungsi AND dengan input biner dan target bipolar sebagai berikut:

Input Bias Target

1	1	1	1
1	0	1	-1
0	1	1	-1
0	0	1	-1

Arsitektur jaringan untuk contoh 8.2 terlihat pada Gambar 8.18.



Bobot awal : $w = [0,0,0,0]$

Bobot bias awal : $b = [0,0]$

Learning rate (alfa): 0,8

Threshold (tetha) : 0,5

Epoh ke-1

Data ke-1

$y_{in} = 0,0 + 0,0 + 0,0 = 0,0$

Hasil aktivasi = 0 ($-0,5 < y_{in} < 0,5$)

Target = 1

Bobot baru:

$w1 = 0,0 + 0,8 * 1,0 * 1,0 = 0,8$

$w2 = 0,0 + 0,8 * 1,0 * 1,0 = 0,8$

Bobot bias baru:

$b = 0,0 + 0,8 * 1,0 = 0,8$

Data ke-2

$y_{in} = 0,8 + 0,8 + 0,0 = 1,6$

Hasil aktivasi = 1 ($y_{in} > 0,5$)

Target = -1

Bobot baru:

$w1 = 0,8 + 0,8 * -1,0 * 1,0 = 0,0$

$w2 = 0,8 + 0,8 * -1,0 * 0,0 = 0,8$

Bobot bias baru:

$b = 0,8 + 0,8 * -1,0 = 0,0$

Data ke-3

$$y_{in} = 0,0 + 0,0 + 0,8 = 0,8$$

$$\text{Hasil aktivasi} = 1 (y_{in} > 0,5)$$

$$\text{Target} = -1$$

Bobot baru:

$$w1 = 0,0 + 0,8 * -1,0 * 0,0 = 0,0$$

$$w2 = 0,8 + 0,8 * -1,0 * 1,0 = 0,0$$

Bobot bias baru:

$$b = 0,0 + 0,8 * -1,0 = -0,8$$

Data ke-4

$$y_{in} = -0,8 + 0,0 + 0,0 = -0,8$$

$$\text{Hasil aktivasi} = -1 (y_{in} < -0,5)$$

$$\text{Target} = -1$$

Epoche-2

Data ke-1

$$y_{in} = -0,8 + 0,0 + 0,0 = -0,8$$

$$\text{Hasil aktivasi} = -1 (y_{in} < -0,5)$$

$$\text{Target} = 1$$

Bobot baru:

$$w1 = 0,0 + 0,8 * 1,0 * 1,0 = 0,8$$

$$w2 = 0,0 + 0,8 * 1,0 * 1,0 = 0,8$$

Bobot bias baru:

$$b = -0,8 + 0,8 * 1,0 = 0,0$$

Data ke-2

$$y_{in} = 0,0 + 0,8 + 0,0 = 0,8$$

$$\text{Hasil aktivasi} = 1 (y_{in} > 0,5)$$

$$\text{Target} = -1$$

Bobot baru:

$$w1 = 0,8 + 0,8 * -1,0 * 1,0 = 0,0$$

$$w2 = 0,8 + 0,8 * -1,0 * 0,0 = 0,8$$

Bobot bias baru:

$$b = 0,0 + 0,8 * -1,0 = -0,8$$

Data ke-3

$$y_{in} = -0,8 + 0,0 + 0,8 = 0,0$$

$$\text{Hasil aktivasi} = 0 (-0,5 < y_{in} < 0,5)$$

$$\text{Target} = -1$$

Bobot baru:

$$w1 = 0,0 + 0,8 * -1,0 * 0,0 = 0,0$$

$w_2 = 0,8 + 0,8 * -1,0 * 1,0 = 0,0$
Bobot bias baru:
 $b = -0,8 + 0,8 * -1,0 = -1,6$

Data ke-4
 $y_{in} = -1,6 + 0,0 + 0,0 = -1,6$
Hasil aktivasi = -1 ($y_{in} < -0,5$)
Target = -1

Epoche ke-3

Data ke-1
 $y_{in} = -1,6 + 0,0 + 0,0 = -1,6$
Hasil aktivasi = -1 ($y_{in} < -0,5$)
Target = 1
Bobot baru:
 $w_1 = 0,0 + 0,8 * 1,0 * 1,0 = 0,8$
 $w_2 = 0,0 + 0,8 * 1,0 * 1,0 = 0,8$
Bobot bias baru:
 $b = -1,6 + 0,8 * 1,0 = -0,8$

Data ke-2
 $y_{in} = -0,8 + 0,8 + 0,0 = 0,0$
Hasil aktivasi = 0 ($-0,5 < y_{in} < 0,5$)
Target = -1
Bobot baru:
 $w_1 = 0,8 + 0,8 * -1,0 * 1,0 = 0,0$
 $w_2 = 0,8 + 0,8 * -1,0 * 0,0 = 0,8$
Bobot bias baru:
 $b = -0,8 + 0,8 * -1,0 = -1,6$

Data ke-3
 $y_{in} = -1,6 + 0,0 + 0,8 = -0,8$
Hasil aktivasi = -1 ($y_{in} < -0,5$)
Target = -1

Data ke-4
 $y_{in} = -1,6 + 0,0 + 0,0 = -1,6$
Hasil aktivasi = -1 ($y_{in} < -0,5$)
Target = -1

Epoche ke-4

Data ke-1
 $y_{in} = -1,6 + 0,0 + 0,8 = -0,8$
Hasil aktivasi = -1 ($y_{in} < -0,5$)

Target = 1

Bobot baru:

$$w1 = 0,0 + 0,8 * 1,0 * 1,0 = 0,8$$

$$w2 = 0,8 + 0,8 * 1,0 * 1,0 = 1,6$$

Bobot bias baru:

$$b = -1,6 + 0,8 * 1,0 = -0,8$$

Data ke-2

$$y_in = -0,8 + 0,8 + 0,0 = 0,0$$

$$\text{Hasil aktivasi} = 0 \text{ } (-0,5 < y_in < 0,5)$$

Target = -1

Bobot baru:

$$w1 = 0,8 + 0,8 * -1,0 * 1,0 = 0,0$$

$$w2 = 1,6 + 0,8 * -1,0 * 0,0 = 1,6$$

Bobot bias baru:

$$b = -0,8 + 0,8 * -1,0 = -1,6$$

Data ke-3

$$y_in = -1,6 + 0,0 + 1,6 = 0,0$$

$$\text{Hasil aktivasi} = 0 \text{ } (-0,5 < y_in < 0,5)$$

Target = -1

Bobot baru:

$$w1 = 0,0 + 0,8 * -1,0 * 0,0 = 0,0$$

$$w2 = 1,6 + 0,8 * -1,0 * 1,0 = 0,8$$

Bobot bias baru:

$$b = -1,6 + 0,8 * -1,0 = -2,4$$

Data ke-4

$$y_in = -2,4 + 0,0 + 0,0 = -2,4$$

$$\text{Hasil aktivasi} = -1 \text{ } (y_in < -0,5)$$

Target = -1

Epoh ke-5

Data ke-1

$$y_in = -2,4 + 0,0 + 0,8 = -1,6$$

$$\text{Hasil aktivasi} = -1 \text{ } (y_in < -0,5)$$

Target = 1

Bobot baru:

$$w1 = 0,0 + 0,8 * 1,0 * 1,0 = 0,8$$

$$w2 = 0,8 + 0,8 * 1,0 * 1,0 = 1,6$$

Bobot bias baru:

$$b = -2,4 + 0,8 * 1,0 = -1,6$$

Data ke-2

$y_{in} = -1,6 + 0,8 + 0,0 = -0,8$
Hasil aktivasi = -1 ($y_{in} < -0,5$)
Target = -1

Data ke-3

$y_{in} = -1,6 + 0,0 + 1,6 = -0,0$
Hasil aktivasi = 0 ($-0,5 < y_{in} < 0,5$)
Target = -1

Bobot baru:

$$w1 = 0,8 + 0,8 * -1,0 * 0,0 = 0,8$$

$$w2 = 1,6 + 0,8 * -1,0 * 1,0 = 0,8$$

Bobot bias baru:

$$b = -1,6 + 0,8 * -1,0 = -2,4$$

Data ke-4

$y_{in} = -2,4 + 0,0 + 0,0 = -2,4$
Hasil aktivasi = -1 ($y_{in} < -0,5$)
Target = -1

Epok ke-6

Data ke-1

$y_{in} = -2,4 + 0,8 + 0,8 = -0,8$
Hasil aktivasi = -1 ($y_{in} < -0,5$)
Target = 1

Bobot baru:

$$w1 = 0,8 + 0,8 * 1,0 * 1,0 = 1,6$$

$$w2 = 0,8 + 0,8 * 1,0 * 1,0 = 1,6$$

Bobot bias baru:

$$b = -2,4 + 0,8 * 1,0 = -1,6$$

Data ke-2

$y_{in} = -1,6 + 1,6 + 0,0 = -0,0$
Hasil aktivasi = 0 ($-0,5 < y_{in} < 0,5$)
Target = -1

Bobot baru:

$$w1 = 1,6 + 0,8 * -1,0 * 1,0 = 0,8$$

$$w2 = 1,6 + 0,8 * -1,0 * 0,0 = 1,6$$

Bobot bias baru:

$$b = -1,6 + 0,8 * -1,0 = -2,4$$

Data ke-3

$y_{in} = -2,4 + 0,0 + 1,6 = -0,8$
Hasil aktivasi = -1 ($y_{in} < -0,5$)
Target = -1

Data ke-4

$$y_{in} = -2,4 + 0,0 + 0,0 = -2,4$$

$$\text{Hasil aktivasi} = -1 (y_{in} < -0,5)$$

$$\text{Target} = -1$$

Epoh ke-7

Data ke-1

$$y_{in} = -2,4 + 0,8 + 1,6 = -0,0$$

$$\text{Hasil aktivasi} = 0 (-0,5 < y_{in} < 0,5)$$

$$\text{Target} = 1$$

Bobot baru:

$$w1 = 0,8 + 0,8 * 1,0 * 1,0 = 1,6$$

$$w2 = 1,6 + 0,8 * 1,0 * 1,0 = 2,4$$

Bobot bias baru:

$$b = -2,4 + 0,8 * 1,0 = -1,6$$

Data ke-2

$$y_{in} = -1,6 + 1,6 + 0,0 = -0,0$$

$$\text{Hasil aktivasi} = 0 (-0,5 < y_{in} < 0,5)$$

$$\text{Target} = -1$$

Bobot baru:

$$w1 = 1,6 + 0,8 * -1,0 * 1,0 = 0,8$$

$$w2 = 2,4 + 0,8 * -1,0 * 0,0 = 2,4$$

Bobot bias baru:

$$b = -1,6 + 0,8 * -1,0 = -2,4$$

Data ke-3

$$y_{in} = -2,4 + 0,0 + 2,4 = 0,0$$

$$\text{Hasil aktivasi} = 0 (-0,5 < y_{in} < 0,5)$$

$$\text{Target} = -1$$

Bobot baru:

$$w1 = 0,8 + 0,8 * -1,0 * 0,0 = 0,8$$

$$w2 = 2,4 + 0,8 * -1,0 * 1,0 = 1,6$$

Bobot bias baru:

$$b = -2,4 + 0,8 * -1,0 = -3,2$$

Data ke-4

$$y_{in} = -3,2 + 0,0 + 0,0 = -3,2$$

$$\text{Hasil aktivasi} = -1 (y_{in} < -0,5)$$

$$\text{Target} = -1$$

Epoh ke-8

Data ke-1

$$y_{in} = -3,2 + 0,8 + 1,6 = -0,8$$

Hasil aktivasi = -1 ($y_{in} < -0,5$)

Target = 1

Bobot baru:

$$w1 = 0,8 + 0,8 * 1,0 * 1,0 = 1,6$$

$$w2 = 1,6 + 0,8 * 1,0 * 1,0 = 2,4$$

Bobot bias baru:

$$b = -3,2 + 0,8 * 1,0 = -2,4$$

Data ke-2

$$y_{in} = -2,4 + 1,6 + 0,0 = -0,8$$

Hasil aktivasi = -1 ($y_{in} < -0,5$)

Target = -1

Data ke-3

$$y_{in} = -2,4 + 0,0 + 2,4 = 0,0$$

Hasil aktivasi = 0 ($-0,5 < y_{in} < 0,5$)

Target = -1

Bobot baru:

$$w1 = 1,6 + 0,8 * -1,0 * 0,0 = 1,6$$

$$w2 = 2,4 + 0,8 * -1,0 * 1,0 = 1,6$$

Bobot bias baru:

$$b = -2,4 + 0,8 * -1,0 = -3,2$$

Data ke-4

$$y_{in} = -3,2 + 0,0 + 0,0 = -3,2$$

Hasil aktivasi = -1 ($y_{in} < -0,5$)

Target = -1

Epok ke-9

Data ke-1

$$y_{in} = -3,2 + 1,6 + 1,6 = 0,0$$

Hasil aktivasi = 0 ($-0,5 < y_{in} < 0,5$)

Target = 1

Bobot baru:

$$w1 = 1,6 + 0,8 * 1,0 * 1,0 = 2,4$$

$$w2 = 1,6 + 0,8 * 1,0 * 1,0 = 2,4$$

Bobot bias baru:

$$b = -3,2 + 0,8 * 1,0 = -2,4$$

Data ke-2

$$y_{in} = -2,4 + 2,4 + 0,0 = 0,0$$

Hasil aktivasi = 0 ($-0,5 < y_{in} < 0,5$)

Target = -1

Bobot baru:

$$w1 = 2,4 + 0,8 * -1,0 * 1,0 = 1,6$$

$$w_2 = 2,4 + 0,8 * -1,0 * 0,0 = 2,4$$

Bobot bias baru:

$$b = -2,4 + 0,8 * -1,0 = -3,2$$

Data ke-3

$$y_{in} = -3,2 + 0,0 + 2,4 = -0,8$$

Hasil aktivasi = -1 ($y_{in} < -0,5$)

Target = -1

Data ke-4

$$y_{in} = -3,2 + 0,0 + 0,0 = -3,2$$

Hasil aktivasi = -1 ($y_{in} < -0,5$)

Target = -1

Epo ke-10

Data ke-1

$$y_{in} = -3,2 + 1,6 + 2,4 = 0,8$$

Hasil aktivasi = 1 ($y_{in} > 0,5$)

Target = 1

Data ke-2

$$y_{in} = -3,2 + 1,6 + 0,0 = -1,6$$

Hasil aktivasi = -1 ($y_{in} < -0,5$)

Target = -1

Data ke-3

$$y_{in} = -3,2 + 0,0 + 2,4 = -0,8$$

Hasil aktivasi = -1 ($y_{in} < -0,5$)

Target = -1

Data ke-4

$$y_{in} = -3,2 + 0,0 + 0,0 = -3,2$$

Hasil aktivasi = -1 ($y_{in} < -0,5$)

Target = -1

Pada epok ke-10 ini sudah tidak terjadi perubahan bobot, sehingga proses pembelajaran dihentikan. Hasil akhir diperoleh:

Nilai bobot, $w_1 = 1,6$; dan $w_2 = 2,4$.

Bobot bias, $b = -3,2$.

Dengan demikian garis yang membatasi daerah positif dengan daerah nol memenuhi pertidaksamaan:

$$1,6 x_1 + 2,4 x_2 - 3,2 > 0,5$$

Sedangkan garis yang membatasi daerah negatif dengan daerah nol memenuhi pertidaksamaan:

$$1,6 x_1 + 2,4 x_2 - 3,2 < -0,5$$

F. Learning Vector Quantization (LVQ)

Learning Vector Quantization (LVQ) adalah suatu metode untuk melakukan pembelajaran pada lapisan kompetitif yang terawasi. Suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vektor-vektor input. Kelas-kelas yang didapatkan sebagai hasil dari lapisan kompetitif ini hanya tergantung pada jarak antara vektor-vektor input. Jika 2 vektor input mendekati sama, maka lapisan kompetitif akan meletakkan kedua vektor input tersebut ke dalam kelas yang sama.

Algoritma

1. Tetapkan: W, MaxIter, Eps, α .
2. Masukkan:
 - o input $x(m,n)$;
 - o m = jumlah input,
 - o n = jumlah data
 - o target(1,n).
3. Tetapkan kondisi awal:
 - o epoch=0;
 - o err=1.
4. Kerjakan jika: (epoch<MaxIter) atau ($\alpha > \text{eps}$)
 - a. epoch = epoch+1;
 - b. Kerjakan untuk i=1 sampai n

i. Tentukan J sedemikian hingga $\|x-w_j\|$ minimum

ii. Perbaiki w_j dengan ketentuan:

o Jika $T = C_j$ maka:

$$w_j(\text{baru}) = w_j(\text{lama}) + \alpha (x - w_j(\text{lama}))$$

o Jika $T \neq C_j$ maka:

$$w_j(\text{baru}) = w_j(\text{lama}) - \alpha (x - w_j(\text{lama}))$$

c. Kurangi nilai α .

Contoh 8.8:

Misalkan diketahui 10 input vektor dalam 2 kelas sebagai berikut:

1. (1, 0, 0, 0, 1, 0)	1
2. (0, 1, 1, 1, 1, 0)	2
3. (0, 0, 1, 0, 0, 1)	1
4. (0, 0, 1, 0, 1, 0)	1
5. (0, 1, 0, 0, 0, 1)	1
6. (1, 0, 1, 0, 1, 1)	1
7. (0, 0, 1, 1, 0, 0)	2
8. (0, 1, 0, 1, 0, 0)	2
9. (1, 0, 0, 1, 0, 1)	2
10. (0, 1, 1, 1, 1, 1)	2

Dua input pertama akan dijadikan sebagai inisialisasi bobot:

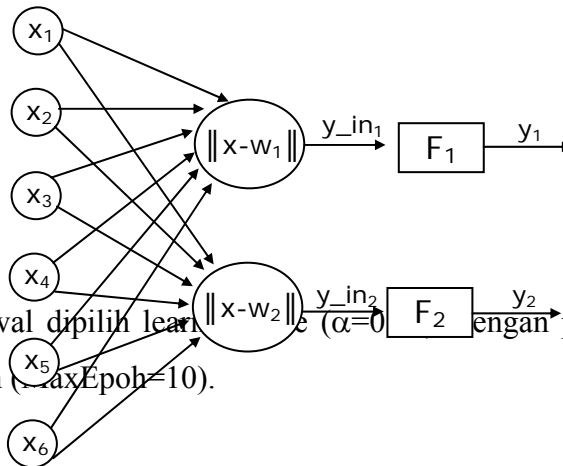
<u>Vektor (w)</u>	<u>Kelas</u>
1. (1, 0, 0, 0, 1, 0)	1
2. (0, 1, 1, 1, 1, 0)	2

Sedangkan 4 input sisanya:

Vektor (x)	Kelas
------------	-------

1. (0, 0, 1, 0, 0, 1) 1
2. (0, 0, 1, 0, 1, 0) 1
3. (0, 1, 0, 0, 0, 1) 1
4. (1, 0, 1, 0, 1, 1) 1
5. (0, 0, 1, 1, 0, 0) 2
6. (0, 1, 0, 1, 0, 0) 2
7. (1, 0, 0, 1, 0, 1) 2
8. (0, 1, 1, 1, 1, 1) 2

akan digunakan sebagai data yang akan dilatih. Arsitektur jaringan syaraf pada contoh 8.8 ini seperti terlihat pada Gambar 8.29.



Sebagai nilai awal dipilih learning rate $\alpha = 0$ dengan pengurangan sebesar $0,1 * \alpha$; dan maksimum epoch ($\text{MaxEpoch} = 10$).

Epok ke-1:

⊕ **Data ke-1: (0, 0, 1, 0, 0, 1)**

Jarak pada:

* bobot ke-1 =

$$= \sqrt{(0-1)^2 + (0-0)^2 + (1-0)^2 + (0-0)^2 + (0-1)^2 + (1-0)^2} = 2,0.$$

* bobot ke-2 =

$$= \sqrt{(0-1)^2 + (0-1)^2 + (1-1)^2 + (0-1)^2 + (0-1)^2 + (1-0)^2} = 2,0.$$

Jarak terkecil pada bobot ke-1

Target data ke-1= 1

Bobot ke- 1 baru:

$$\begin{aligned} w_{11} &= w_{11} + \alpha * (x_{11} - w_{11}) = 1 + 0,05 * (0-1) = 0,9500; \\ w_{12} &= w_{12} + \alpha * (x_{12} - w_{12}) = 0 + 0,05 * (0-0) = 0,0000; \\ w_{13} &= w_{13} + \alpha * (x_{13} - w_{13}) = 0 + 0,05 * (1-0) = 0,0500; \\ w_{14} &= w_{14} + \alpha * (x_{14} - w_{14}) = 0 + 0,05 * (0-0) = 0,0000; \\ w_{15} &= w_{15} + \alpha * (x_{15} - w_{15}) = 1 + 0,05 * (0-1) = 0,9500; \\ w_{16} &= w_{16} + \alpha * (x_{16} - w_{16}) = 0 + 0,05 * (1-0) = 0,0500; \end{aligned}$$

Jadi

$$w_1 = (0,9500 \ 0,0000 \ 0,0500 \ 0,0000 \ 0,9500 \ 0,0500)$$

⊕ **Data ke-2: (0, 0, 1, 0, 1, 0)**

Jarak pada:

* bobot ke-1 =

$$\begin{aligned} &= \sqrt{(0 - 0,95)^2 + (0 - 0)^2 + (1 - 0,05)^2 + (0 - 0)^2 + (1 - 0,95)^2 + (0 - 0,05)^2} \\ &= 1,3454. \end{aligned}$$

* bobot ke-2 =

$$\begin{aligned} &= \sqrt{(0 - 0)^2 + (0 - 1)^2 + (1 - 1)^2 + (0 - 1)^2 + (1 - 1)^2 + (0 - 0)^2} \\ &= 1,4142. \end{aligned}$$

Jarak terkecil pada bobot ke-1

Target data ke-2= 1

Bobot ke- 1 baru:

$$\begin{aligned} w_{11} &= w_{11} + \alpha*(x_{21} - w_{11}) = 0,95 + 0,05*(0-0,95) = 0,9025; \\ w_{12} &= w_{12} + \alpha*(x_{22} - w_{12}) = 0,00 + 0,05*(0-0,00) = 0,0000; \\ w_{13} &= w_{13} + \alpha*(x_{23} - w_{13}) = 0,05 + 0,05*(1-0,05) = 0,0975; \\ w_{14} &= w_{14} + \alpha*(x_{24} - w_{14}) = 0,00 + 0,05*(0-0,00) = 0,0000; \\ w_{15} &= w_{15} + \alpha*(x_{25} - w_{15}) = 0,95 + 0,05*(1-0,95) = 0,9525; \\ w_{16} &= w_{16} + \alpha*(x_{26} - w_{16}) = 0,05 + 0,05*(0-0,05) = 0,0475; \end{aligned}$$

Jadi

$$w_1 = (0,9025 \ 0,0000 \ 0,0975 \ 0,0000 \ 0,9525 \ 0,0475)$$

⊕ **Data ke-3: (0, 1, 0, 0, 0, 1)**

Jarak pada:

* bobot ke-1 =

$$\begin{aligned} &= \sqrt{(0 - 0,9025)^2 + (1 - 0)^2 + (0 - 0,0975)^2 + (0 - 0)^2 + (0 - 0,9525)^2 + (1 - 0,0475)^2} \\ &= 1,9075 \end{aligned}$$

* bobot ke-2 =

$$\begin{aligned} &= \sqrt{(0 - 0)^2 + (1 - 1)^2 + (0 - 1)^2 + (0 - 1)^2 + (0 - 1)^2 + (1 - 0)^2} \\ &= 2,000. \end{aligned}$$

Jarak terkecil pada bobot ke-1

Target data ke-3= 1

Bobot ke- 1 baru:

$$\begin{aligned}w_{11} &= w_{11} + \alpha*(x_{31} - w_{11}) = 0,9025 + 0,05*(0-0,9025) = 0,8574; \\w_{12} &= w_{12} + \alpha*(x_{32} - w_{12}) = 0,0000 + 0,05*(0-0,0000) = 0,0500; \\w_{13} &= w_{13} + \alpha*(x_{33} - w_{13}) = 0,0975 + 0,05*(1-0,0975) = 0,0926; \\w_{14} &= w_{14} + \alpha*(x_{34} - w_{14}) = 0,0000 + 0,05*(0-0,0000) = 0,0000; \\w_{15} &= w_{15} + \alpha*(x_{35} - w_{15}) = 0,9525 + 0,05*(1-0,9525) = 0,9049; \\w_{16} &= w_{16} + \alpha*(x_{36} - w_{16}) = 0,0475 + 0,05*(0-0,0475) = 0,0951;\end{aligned}$$

Jadi:

$$w_1 = (0,8574 \ 0,0500 \ 0,0926 \ 0,0000 \ 0,9049 \ 0,0951)$$

⊕ **Data ke-4: (1, 0, 1, 0, 1, 1)**

Jarak pada:

* bobot ke-1 =

$$\begin{aligned}&= \sqrt{(1 - 0,8574)^2 + (0 - 0,0500)^2 + (1 - 0,0926)^2 + (0 - 0)^2 + (1 - 0,9049)^2 + (1 - 0,0951)^2} \\&= 1,2938\end{aligned}$$

* bobot ke-2 =

$$\begin{aligned}&= \sqrt{(1 - 0)^2 + (0 - 1)^2 + (1 - 1)^2 + (0 - 1)^2 + (1 - 1)^2 + (1 - 0)^2} \\&= 2,000.\end{aligned}$$

Jarak terkecil pada bobot ke-1

Target data ke-4= 1

Bobot ke- 1 baru:

$$\begin{aligned}w_{11} &= w_{11} + \alpha*(x_{41} - w_{11}) = 0,8574 + 0,05*(1-0,8574) = 0,8645; \\w_{12} &= w_{12} + \alpha*(x_{42} - w_{12}) = 0,0500 + 0,05*(0-0,0500) = 0,0475; \\w_{13} &= w_{13} + \alpha*(x_{43} - w_{13}) = 0,0926 + 0,05*(1-0,0926) = 0,1380; \\w_{14} &= w_{14} + \alpha*(x_{44} - w_{14}) = 0,0000 + 0,05*(0-0,0000) = 0,0000; \\w_{15} &= w_{15} + \alpha*(x_{45} - w_{15}) = 0,9049 + 0,05*(1-0,9049) = 0,9096; \\w_{16} &= w_{16} + \alpha*(x_{46} - w_{16}) = 0,0951 + 0,05*(1-0,0951) = 0,1404;\end{aligned}$$

Jadi:

$$w_1 = (0,8645 \ 0,0475 \ 0,1380 \ 0,0000 \ 0,9096 \ 0,1404)$$

⊕ **Data ke-5: (0, 0, 1, 1, 0, 0)**

Jarak pada:

* bobot ke-1 =

$$= \sqrt{(0 - 0,8645)^2 + (0 - 0,0475)^2 + (1 - 0,1380)^2 + (1 - 0)^2 + (0 - 0,9096)^2 + (0 - 0,1404)^2}$$

$$= 1,8275$$

* bobot ke-2 =

$$= \sqrt{(0 - 0)^2 + (0 - 1)^2 + (1 - 1)^2 + (1 - 1)^2 + (0 - 1)^2 + (0 - 0)^2}$$

$$= 1,4142.$$

Jarak terkecil pada bobot ke-2

Target data ke-5= 2

Bobot ke- 2 baru:

$$w_{21} = w_{21} + \alpha*(x_{51} - w_{21}) = 0 + 0,05*(1-0) = 0,0000;$$

$$w_{22} = w_{22} + \alpha*(x_{52} - w_{22}) = 1 + 0,05*(0-1) = 0,9500;$$

$$w_{23} = w_{23} + \alpha*(x_{53} - w_{23}) = 1 + 0,05*(1-1) = 1,0000;$$

$$w_{24} = w_{24} + \alpha*(x_{54} - w_{24}) = 1 + 0,05*(0-1) = 1,0000;$$

$$w_{25} = w_{25} + \alpha*(x_{55} - w_{25}) = 1 + 0,05*(1-1) = 0,9500;$$

$$w_{26} = w_{26} + \alpha*(x_{56} - w_{26}) = 0 + 0,05*(1-0) = 0,0000;$$

Jadi:

$$w_2 = (0,0000 \ 0,9500 \ 1,0000 \ 1,0000 \ 0,9500 \ 0,0000)$$

☹ **Data ke-6: (0, 1, 0, 1, 0, 0)**

Jarak pada:

* bobot ke-1 =

$$= \sqrt{(0 - 0,8645)^2 + (1 - 0,0475)^2 + (0 - 0,1380)^2 + (1 - 0)^2 + (0 - 0,9096)^2 + (0 - 0,1404)^2}$$

$$= 1,8764$$

* bobot ke-2 =

$$= \sqrt{(0 - 0)^2 + (1 - 0,9500)^2 + (0 - 1)^2 + (1 - 1)^2 + (0 - 0,9500)^2 + (0 - 0)^2}$$

$$= 1,3802.$$

Jarak terkecil pada bobot ke-2

Target data ke-6= 2

Bobot ke- 2 baru:

$$w_{21} = w_{21} + \alpha*(x_{61} - w_{21}) = 0,0000 + 0,05*(0-0,0000) = 0,0000;$$

$$w_{22} = w_{22} + \alpha*(x_{62} - w_{22}) = 0,9500 + 0,05*(1-0,9500) = 0,9525;$$

$$w_{23} = w_{23} + \alpha*(x_{63} - w_{23}) = 1,0000 + 0,05*(0-1,0000) = 0,9500;$$

$$w_{24} = w_{24} + \alpha*(x_{64} - w_{24}) = 1,0000 + 0,05*(1-1,0000) = 1,0000;$$

$$w_{25} = w_{25} + \alpha*(x_{65} - w_{25}) = 0,9500 + 0,05*(0-0,9500) = 0,9025;$$

$$w_{26} = w_{26} + \alpha*(x_{66} - w_{26}) = 0,0000 + 0,05*(0-0,0000) = 0,0000;$$

Jadi:

$$w_2 = (0,0000 \ 0,9525 \ 0,9500 \ 1,0000 \ 0,9025 \ 0,0000)$$

⊗ **Data ke-7: (1, 0, 0, 1, 0, 1)**

Jarak pada:

* bobot ke-1 =

$$= \sqrt{(1 - 0,8645)^2 + (0 - 0,0475)^2 + (0 - 0,1380)^2 + (1 - 0)^2 + (0 - 0,9096)^2 + (1 - 0,1404)^2}$$

$$= 1,6143$$

* bobot ke-2 =

$$= \sqrt{(1 - 0)^2 + (0 - 0,9525)^2 + (0 - 0,9500)^2 + (1 - 1)^2 + (0 - 0,9025)^2 + (1 - 0)^2}$$

$$= 2,1504$$

Jarak terkecil pada bobot ke-1

Target data ke-7= 2

Bobot ke- 1 baru:

$$w_{11} = w_{11} - \alpha*(x_{71} - w_{11}) = 0,8645 - 0,0266*(1-0,8645) = 0,8577;$$

$$w_{12} = w_{12} - \alpha*(x_{72} - w_{12}) = 0,0475 - 0,0266*(0-0,0475) = 0,0499;$$

$$w_{13} = w_{13} - \alpha*(x_{73} - w_{13}) = 0,1380 - 0,0266*(0-0,1380) = 0,1449;$$

$$w_{14} = w_{14} - \alpha*(x_{74} - w_{14}) = 0,0000 - 0,0266*(1-0,0000) = -0,0500;$$

$$w_{15} = w_{15} - \alpha*(x_{75} - w_{15}) = 0,9096 - 0,0266*(0-0,9096) = 0,9551;$$

$$w_{16} = w_{16} - \alpha*(x_{76} - w_{16}) = 0,1404 - 0,0266*(1-0,1404) = 0,0974;$$

Jadi:

$$w_1 = (0,8577 \ 0,0499 \ 0,1449 \ -0,0500 \ 0,9511 \ 0,0974)$$

⊗ **Data ke-8: (0, 1, 1, 1, 1, 1)**

Jarak pada:

* bobot ke-1 =

$$= \sqrt{(0 - 0,8577)^2 + (1 - 0,0499)^2 + (1 - 0,1449)^2 + (1 + 0,0500)^2 + (1 - 0,9551)^2 + (1 - 0,0974)^2}$$

$$= 2,0710$$

* bobot ke-2 =

$$= \sqrt{(0 - 0)^2 + (1 - 0,9525)^2 + (1 - 0,9500)^2 + (1 - 1)^2 + (1 - 0,9025)^2 + (1 - 0)^2}$$

$$= 1,0028$$

Jarak terkecil pada bobot ke-2

Target data ke-8= 2

Bobot ke- 2 baru:

$$w_{21} = w_{21} + \alpha*(x_{81} - w_{21}) = 0,0000 + 0,05*(0-0,0000) = 0,0000;$$

$$w_{22} = w_{22} + \alpha*(x_{82} - w_{22}) = 0,9525 + 0,05*(1-0,9525) = 0,9549;$$

$$w_{23} = w_{23} + \alpha*(x_{83} - w_{23}) = 0,9500 + 0,05*(0-0,9500) = 0,9525;$$

$$w_{24} = w_{24} + \alpha*(x_{84} - w_{24}) = 1,0000 + 0,05*(1-1,0000) = 1,0000;$$

$$w_{25} = w_{25} + \alpha*(x_{85} - w_{25}) = 0,9025 + 0,05*(0-0,9025) = 0,9074;$$

$$w_{26} = w_{26} + \alpha*(x_{86} - w_{26}) = 0,0000 + 0,05*(0-0,0000) = 0,0500;$$

Jadi:

$$w_2 = (0,0000 \ 0,9549 \ 0,9525 \ 1,0000 \ 0,9074 \ 0,0500)$$

$$\alpha = \alpha - 0,1*\alpha = 0,05 - 0,1*0,05 = 0,045$$

Proses tersebut diteruskan untuk epoch ke-2 sampai ke-10, untuk setiap data dengan menggunakan cara yang sama. Setelah mencapai epoch yang ke-10 diperoleh bobot akhir:

$$w_1 = (0,3727 \ 0,2161 \ 0,6347 \ -0,2164 \ 0,7981 \ 0,4254)$$

$$w_2 = (0,0000 \ 0,7969 \ 0,7900 \ 1,0000 \ 0,5869 \ 0,2171)$$

Apabila kita ingin mensimulasikan input: (0, 1, 0, 1, 1, 0), maka kita cari terlebih dahulu jarak input tersebut dengan kedua bobot. Nomor dari bobot dengan jarak yang terpendek akan menjadi kelasnya.

Jarak pada:

* bobot ke-1 =

$$= \sqrt{(0 - 0,3727)^2 + (1 - 0,2161)^2 + (0 - 0,6347)^2 + (1 + 0,2164)^2 + (1 - 0,7981)^2 + (0 - 0,4254)^2}$$

$$= 1,6904$$

* bobot ke-2 =

$$= \sqrt{(0 - 0)^2 + (1 - 0,7969)^2 + (0 - 0,7900)^2 + (1 - 1)^2 + (1 - 0,5869)^2 + (0 - 0,2171)^2}$$

$$= 0,9398$$

Jarak terkecil pada bobot ke-2

Sehingga input tersebut termasuk dalam kelas 2.