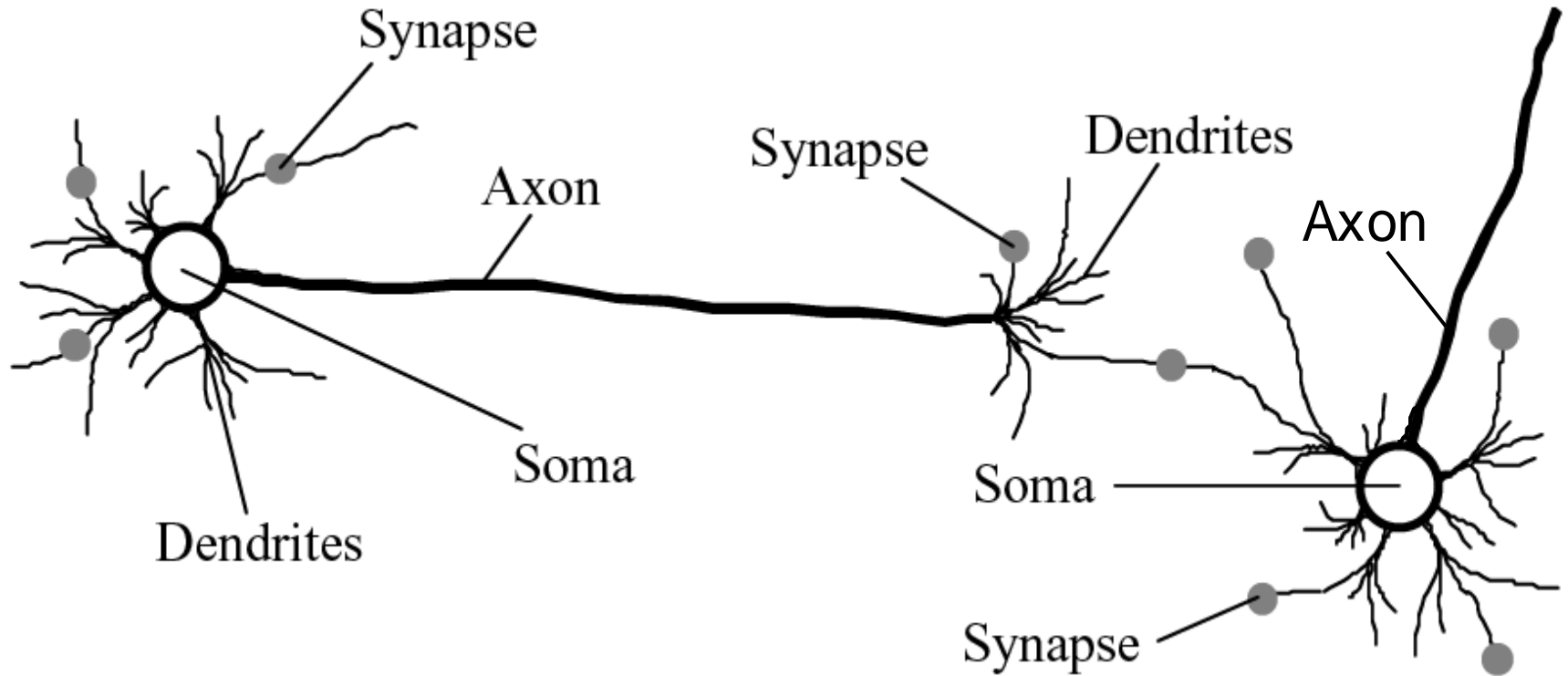




JARINGAN SYARAF TIRUAN

Dr. Eng. Anik Nur Handayani, S.T.M.T

JARINGAN SYARAF BIOLOGI



JARINGAN SYARAF BIOLOGI



- Otak manusia terdiri dari neuron-neuron dan penghubung yang disebut sinapsis.
- Neuron bekerja berdasarkan impuls/sinyal yang diberikan pada neuron.
- Neuron meneruskan pada neuron lain.
Manusia mempunyai memiliki 10^{12} neuron dan $6 \cdot 10^8$ sinapsis.

JARINGAN SYARAF BIOLOGI



Neuron memiliki 3 komponen penting:

- Dendrit

Dendrit menerima sinyal dari neuron lain -> impuls elektrik -> dikirim -> celah sinaptik melalui proses kimiawi.

- Soma

Menjumlahkan semua sinyal-sinyal yang masuk. Jika jumlahan tersebut cukup kuat dan melebihi batas ambang/*threshold* sinyal tersebut akan diteruskan ke sel lain melalui axon.

- Axon

JARINGAN SYARAF TIRUAN



JST adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi.

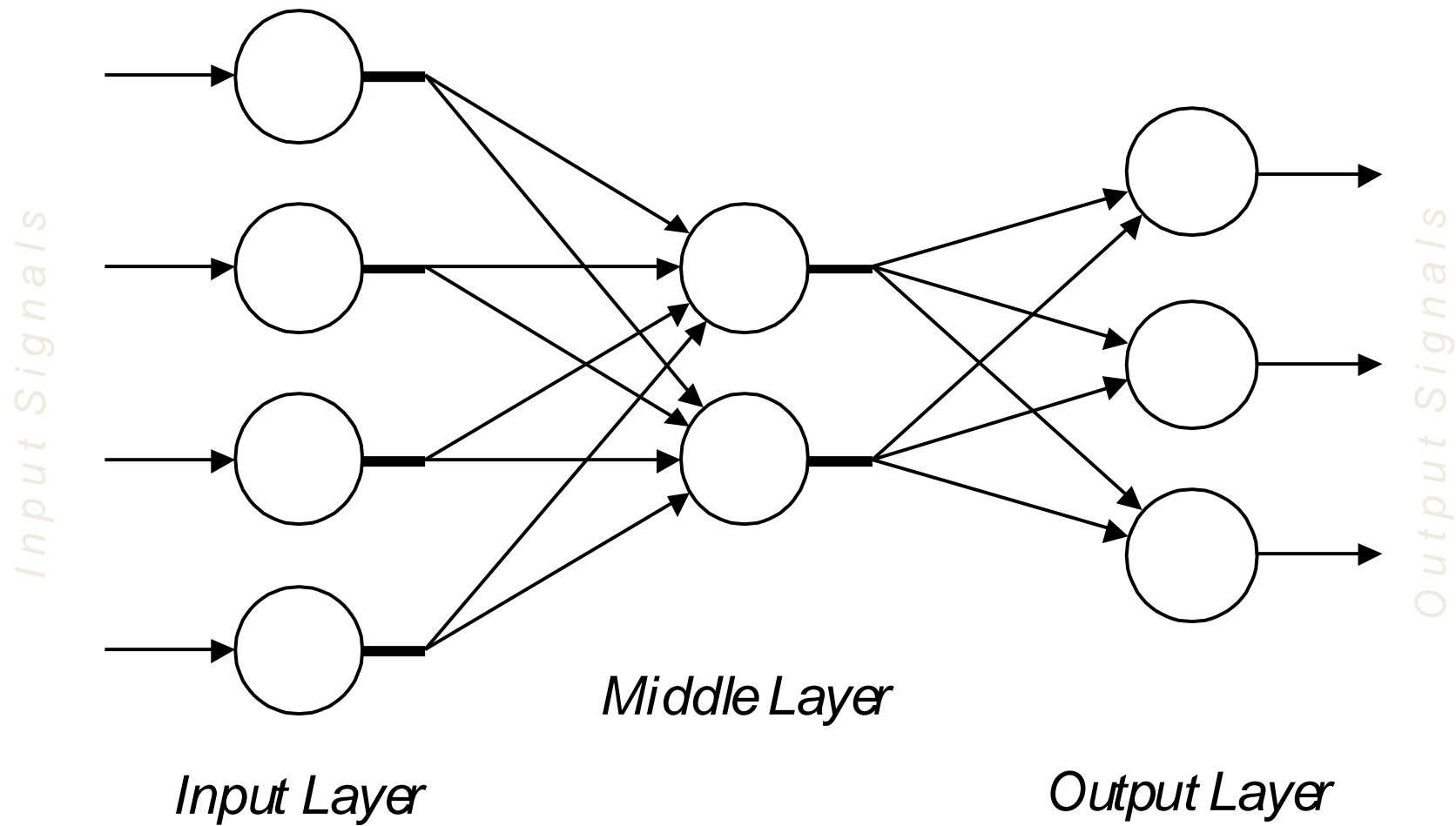
JST dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi, dengan asumsi :

- Pemrosesan informasi terjadi pada banyak elemen sederhana (neuron)
- Sinyal dikirimkan diantara neuron-neuron melalui penghubung-penghubung

JARINGAN SYARAF TIRUAN



- Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal.
- Untuk menentukan output setiap neuron menggunakan fungsi aktivasi / fungsi yang bukan linier, yang dikenakan pada penjumlahan input yang diterima. Biasanya output ini selanjutnya dibandingkan dengan suatu batas ambang.

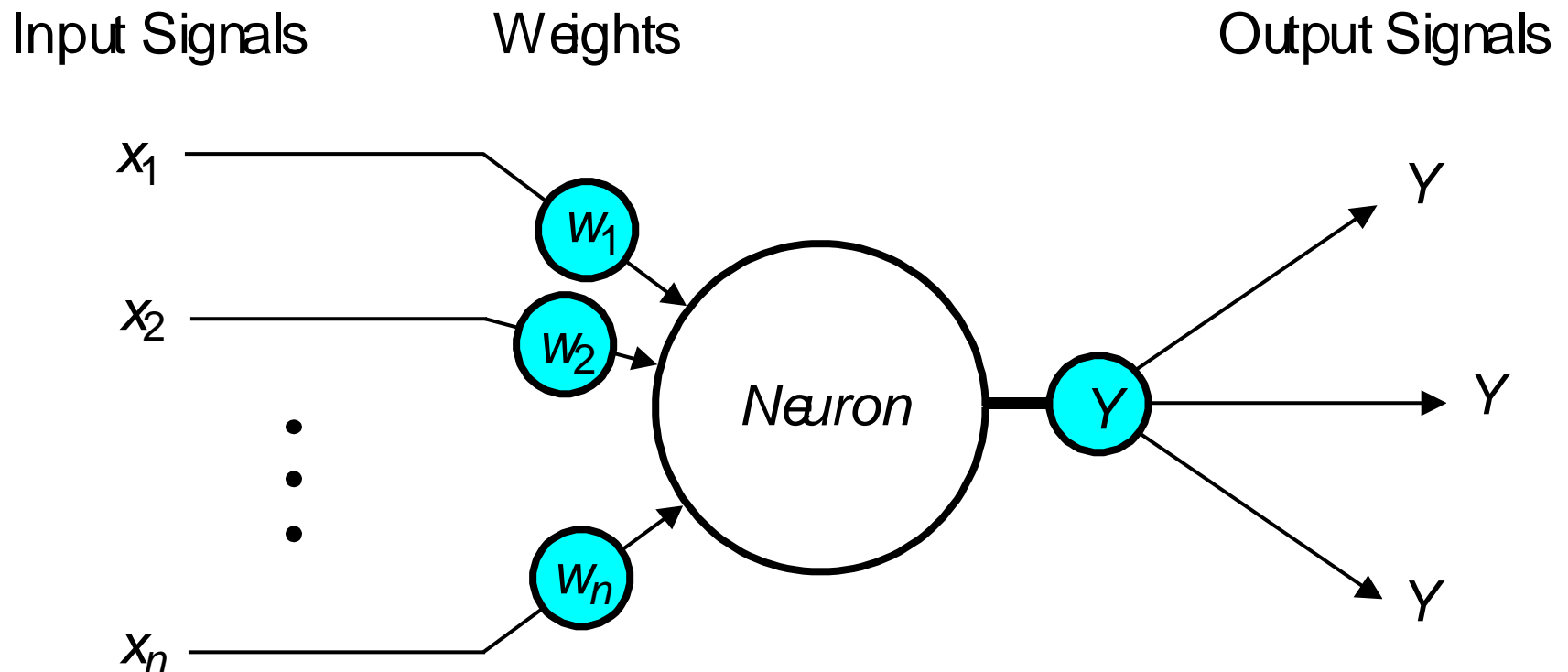


Analogy between biological and artificial neural networks

<i>Biological Neural Network</i>	<i>Artificial Neural Network</i>
Soma	Neuron
Dendrite	Input
Axon	Output
Synapse	Weight

The neuron as a simple computing element

Diagram of a neuron





JARINGAN SYARAF TIRUAN

JST, ditentukan oleh 3 hal :

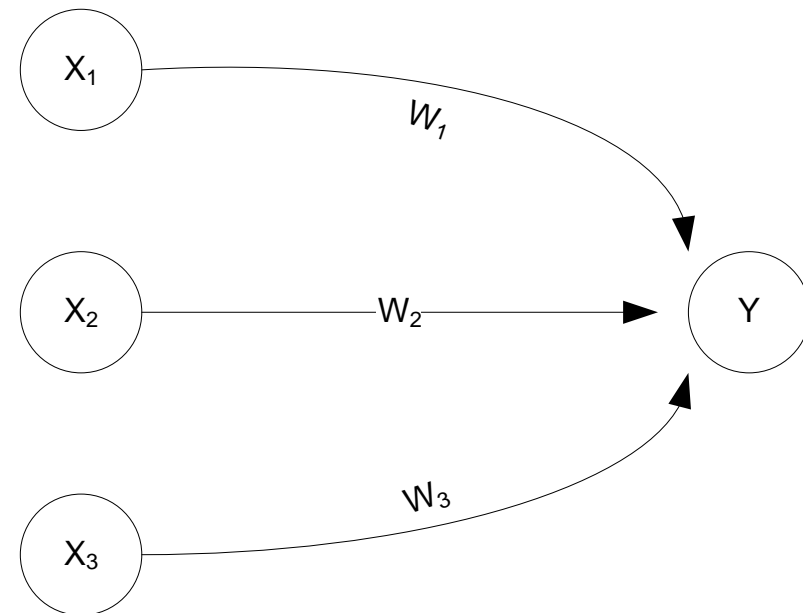
1. Pola hubungan antar neuron (disebut arsitektur jaringan)
2. Metode untk menentukan bobot penghubung/metode training atau leraning algoritma
3. Fungsi aktivasi

JARINGAN SYARAF TIRUAN

Contoh :

Y menerima input dari neuron x_1 , x_2 , dan x_3 dengan bobot hubungan masing-masing w_1 , w_2 , dan w_3 . Ketiga impuls neuron yang ada dijumlahkan:

$$\text{net} = x_1w_1 + x_2w_2 + x_3w_3$$





DASAR-DASAR MATEMATIKA

VEKTOR

Vektor adalah tupel
bilangan-bilangan riil,
dimana mempunyai
besar dan arah.

Notasinya adalah huruf-
huruf kecil seperti x , y , z

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$$

$$x^t = (x_1 \quad x_2 \quad \dots \quad x_n)$$

Apabila $x_1=y_1, x_2=y_2, \dots, x_n = y_n$, maka :

$$\begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix}$$



Operasi-operasi vektor

- Perkalian vektor dengan skalar

$$kx = \begin{pmatrix} kx_1 \\ kx_2 \\ \dots \\ kx_n \end{pmatrix}$$

Operasi-operasi vektor

- Penjumlahan dan pengurangan 2 buah vektor

1. $x + y$

$$\begin{pmatrix} x1 \\ x2 \\ \dots \\ xn \end{pmatrix} + \begin{pmatrix} y1 \\ y2 \\ \dots \\ yn \end{pmatrix} = \begin{pmatrix} x1 + y1 \\ x2 + y2 \\ \dots \\ xn + yn \end{pmatrix}$$

2. $x - y$

$$\begin{pmatrix} x1 \\ x2 \\ \dots \\ xn \end{pmatrix} - \begin{pmatrix} y1 \\ y2 \\ \dots \\ yn \end{pmatrix} = \begin{pmatrix} x1 - y1 \\ x2 - y2 \\ \dots \\ xn - yn \end{pmatrix}$$

Operasi-operasi vektor

- Hasil kali titik 2 vektor

$$\text{Jika } x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} \text{ dan } y = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix}$$

Maka $x \cdot y = x_1 y_1 + x_2 y_2 + \dots + x_n y_n$



Operasi-operasi vektor

- Operasi penjumlahan vektor dan perkalian dengan skalar
 1. Komutatif : $x+y = y+x$
 2. Asosiatif : $x+(y+z) = (x+y)+z$
 $c_1(c_2x) = (c_1c_2)x$
 3. Distributif : $c_1(x+y) = c_1x + c_1y$
 $(c_1+c_2)x = c_1x + c_2x$
- Operasi perkalian titik
 1. Komutatif : $x.y = y.x$
 2. Asosiatif : $(c_1x).y = c_1.(xy) = x.(c_1y)$
 3. Distributif : $x(y+z) = x.y + x.z$
 4. $x.x = 0$, jika $x = 0$

MATRIKS

- Matriks adalah kumpulan bilangan-bilangan yang disusun dalam larik baris dan kolom. Matriks hanya mempunyai besar saja. Umumnya matriks diberi notasi huruf kapital A, B,
- Jika matriks A terdiri dari m baris dan n kolom (sering disebut ordo $m \times n$), maka:

$$A = a_{ij} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

MATRIKS

Jenis-jenis matriks:

- Matriks nol adalah matriks yang semua elemennya = 0
- Matriks bujur sangkar adalah matriks yang jumlah baris = kolom
- Matriks diagonal D adalah matriks bujur sangkar dimana semua elemen di luar diagonal utama = 0, dan tidak semua elemen pada diagonal utama = 0.

$$D = \begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{nn} \end{pmatrix}$$



MATRIKS

Jika semua elemen diagonal utama pada matriks diagonal = 1 maka disebut dengan matriks identitas.

$$D = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

Operasi Pada Matriks

- Perkalian Matriks dengan Skalar

Jika A adalah sembarang matriks dan c adalah skalar, maka cA adalah matriks yang elemennya merupakan perkalian elemen matriks A dengan skalar c .

$$cA = \begin{pmatrix} ca_{11} & ca_{12} & \dots & ca_{1n} \\ ca_{21} & ca_{22} & \dots & ca_{2n} \\ & \dots & & \\ ca_{m1} & ca_{m2} & \dots & ca_{mn} \end{pmatrix}$$



Operasi Pada Matriks

- Penjumlahan/Pengurangan Matriks

Dua buah matriks dapat dijumlahkan/dikurangkan jika ordenya sama. Hasil penjumlahan/pengurangan 2 buah matriks sama dengan penjumlahan/pengurangan elemen-elemen matriks yang seletak.

Operasi Pada Matriks

$$\text{Jika } A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

$$\text{dan } B = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{pmatrix}$$

$$\text{maka } A + B = \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \dots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} & \dots & a_{2n} + b_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} + b_{m1} & a_{m2} + b_{m2} & \dots & a_{mn} + b_{mn} \end{pmatrix}$$

Operasi Pada Matriks

- Perkalian Matriks

Matriks A dapat dikalikan dengan matriks B apabila jumlah kolom matriks A = jumlah baris matriks B. Apabila A berorde $m \times n$ dan B berorde $n \times p$, maka $C = A \cdot B$ berorde $m \times p$ dengan elemen C :

$$c_{ik} = \sum_{j=1}^n a_{ij} b_{jk} = a_{i1} b_{1k} + a_{i2} b_{2k} + \dots + a_{in} b_{nk}$$

Operasi Pada Matriks

- Transpose Matriks

Transpose matriks A (simbol A^t) diperoleh dari matriks A dengan cara menukar baris dan kolomnya. Jika A berorde $m \times n$, maka A^t berorde $n \times m$

$$\text{jika } A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

$$\text{maka } A^t = \begin{pmatrix} a_{11} & a_{21} & \dots & a_{n1} \\ a_{12} & a_{22} & \dots & a_{n2} \\ \dots & \dots & \dots & \dots \\ a_{1m} & a_{2m} & \dots & a_{nm} \end{pmatrix}$$

CONTOH

- Diketahui matriks A, B, dan C sebagai berikut:

$$A = \begin{pmatrix} 2 & 1 & -2 \\ 0 & 1 & 1 \\ -4 & 1 & 4 \end{pmatrix}; B = \begin{pmatrix} 3 & 1 \\ -1 & 2 \\ 1 & 0 \end{pmatrix}; C = \begin{pmatrix} 4 & 0 & 1 \\ 2 & -2 & 1 \end{pmatrix}$$

Hitunglah (jika mungkin) :

- a. $2A$
- b. $A+B$
- c. AB
- d. BA
- e. $(CB)^t$
- f. AI (dengan I adalah matriks identitas)

Jawab:

- a.
$$2A = 2 \begin{pmatrix} 2 & 1 & -2 \\ 0 & 1 & 1 \\ -4 & 1 & 4 \end{pmatrix} = \begin{pmatrix} 4 & 2 & -4 \\ 0 & 2 & 2 \\ -8 & 2 & 8 \end{pmatrix}$$
- b. A+B tidak mungkin dilakukan karena ordenya tidak sama (orde A=3x3 sedangkan orde B=3x2)
- c.
$$AB = \begin{pmatrix} 2 & 1 & -2 \\ 0 & 1 & 1 \\ -4 & 1 & 4 \end{pmatrix} \begin{pmatrix} 3 & 1 \\ -1 & 2 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 3 & 4 \\ 0 & 2 \\ -9 & -2 \end{pmatrix}$$



Jawab:

- d. BA tidak mungkin dilakukan karena jumlah kolom B ($=2$) tidak sama dengan jumlah baris A ($=3$). Kenyataan ini menunjukkan bahwa perkalian matriks tidak komutatif.

Jawab:

e.

$$CB = \begin{pmatrix} 4 & 0 & 1 \\ 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} 3 & 1 \\ -1 & 2 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 13 & 4 \\ 9 & -2 \end{pmatrix}$$

$$\text{maka } (CB)^t = \begin{pmatrix} 13 & 9 \\ 4 & -2 \end{pmatrix}$$

Jawab:

- f.

$$AI = \begin{pmatrix} 2 & 1 & -2 \\ 0 & 1 & 1 \\ -4 & 1 & 4 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 1 & -2 \\ 0 & 1 & 1 \\ -4 & 1 & 4 \end{pmatrix}$$

Secara umum, untuk sembarang matriks bujur sangkar A, $AI = IA = A$

MODEL NEURON

Neuron adalah unit pemroses informasi yang menjadi dasar dalam pengoperasian jaringan syaraf tiruan.

Neuron terdiri dari 3 elemen:

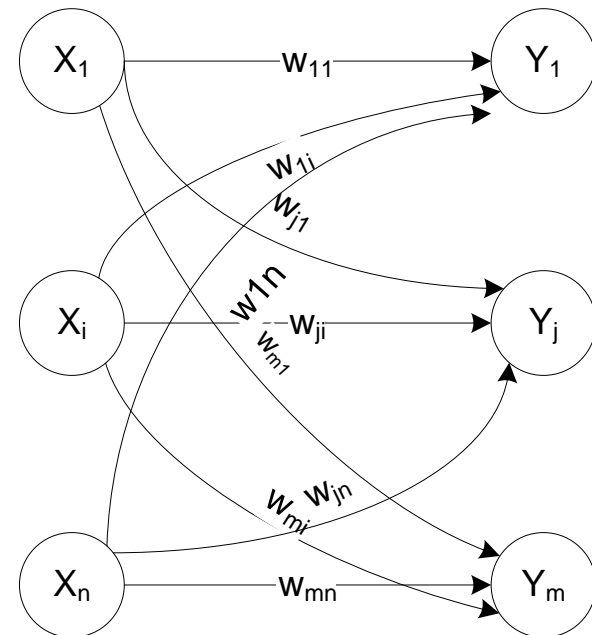
1. Himpunan unit-unit yang dihubungkan dengan jalur koneksi, yang memiliki bobot/kekuatan yang berbeda-beda.
2. Suatu unit penjumlah yang akan menjumlahkan input-input sinyal yang sudah dikalikan dengan bobotnya.
3. Fungsi aktivasi yang akan menentukan apakah sinyal dari input neuron akan diteruskan ke neuron lain atau tidak.

ARSITEKTUR JARINGAN MODEL NEURON

Beberapa arsitektur jaringan yang sering dipakai :

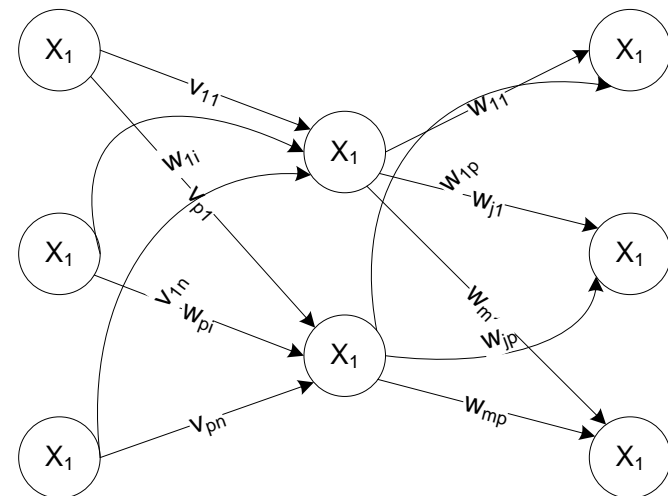
- Jaringan Layar Tunggal (Single Layer Network)

- >> Dalam jaringan ini, sekumpulan input neuron dihubungkan langsung dengan sekumpulan outputnya.
- >> “*Semua unit input dihubungkan dengan semua unit output, meskipun dengan bobot yang berbeda-beda*”
- >> Besaran W_{ji} menyatakan bobot hubungan antara unit ke- i dalam input dengan unit ke- j dalam output.



ARSITEKTUR JARINGAN MODEL NEURON

- Jaringan layer jamak (Multi Layer Network)
 - >> Jaringan layer jamak merupakan perluasan dari layer tunggal.
 - >> Diantara unit input dan output, terdapat layer tersembunyi (*Hidden Layer*)
 - >> Jaringan layer jamak dapat menyelesaikan masalah yang lebih kompleks dibandingkan dengan layer tunggal, meskipun kadang dalam proses pelatihan lebih kompleks.
- Jaringan Reccurent



FUNGSI AKTIVASI

- >> Dalam jaringan syaraf tiruan, fungsi aktivasi dipakai untuk menentukan keluaran suatu neuron.
- >> Argumen fungsi aktivasi adalah net masukan (kombinasi linier masukan dan bobotnya).

$$net = \sum X_i W_i, \text{ maka fungsi aktivasiya } f(net) = f\left(\sum X_i W_i\right)$$



FUNGSI AKTIVASI

- Fungsi Threshold (Batas Ambang)

Untuk beberapa kasus, fungsi threshold yang dibuat tidak berharga 0 atau 1, tapi berharga -1 atau 1 (threshold bipolar)

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

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

FUNGSI AKTIVASI

- Fungsi sigmoid

Fungsi sigmoid sering dipakai karena nilai fungsinya yang terletak antara 0 dan 1.

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

$$f'(x) = f(x)(1 - f(x))$$



FUNGSI AKTIVASI

- Fungsi Identitas

$$f(x) = x$$

Fungsi identitas sering dipakai apabila kita menginginkan keluaran jaringan berupa sembarang bilangan riil, bukan hanya pada range $[0,1]$ atau $[-1,1]$



BIAS DAN THERSHOLD

- >> Dalam arsitektur JST, terkadang ditambahkan sebuah unit masukan yang nilainya selalu=1. Atau dapat juga didefinisikan sebagai sebuah input yang nilainya=1.
- >> Bias berfungsi untuk mengubah nilai threshold menjadi = 0 (bukan = a).

BIAS DAN THERSHOLD

- Jika melibatkan bias, maka keluaran unit penjumlah :

$$net = b + \sum_i X_i W_i$$

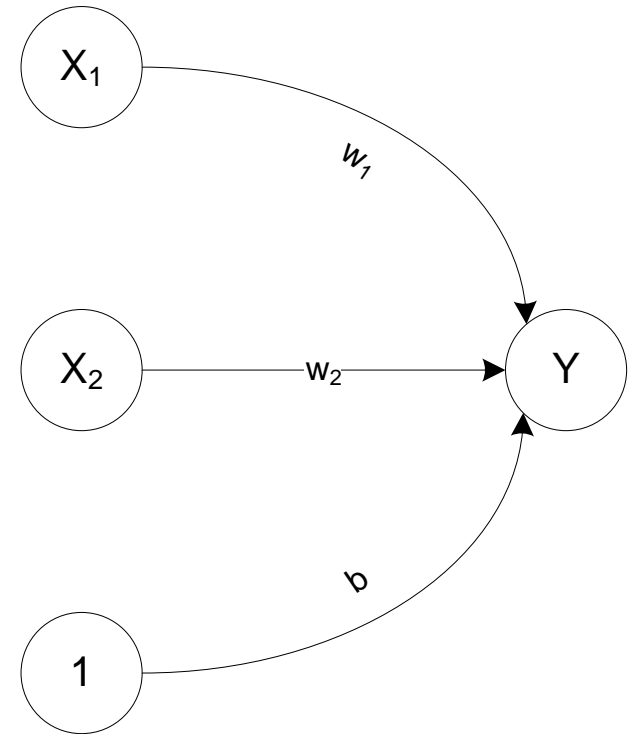
- Fungsi aktivasi threshold menjadi:

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

BIAS DAN THERSHOLD

Contoh Soal:

Suatu jaringan layar tunggal seperti di samping ini, terdiri dari 2 input $x_1=0.7$ dan $x_2=2.1$ dan memiliki bias. Bobot garis $w_1=0.5$, $w_2=-0.3$, bobot bias $b=1.2$. Tentukan keluaran neuron Y jika fungsi aktivasinya adalah fungsi threshold bipolar.





BIAS DAN THERSHOLD

Jawab:

$$\begin{aligned} net &= b + \sum_i X_i W_i \\ &= 1.2 + (0.7 * 0.5) + (2.1 * -0.3) \\ &= 0.92 \end{aligned}$$

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

PELATIHAN DENGAN DAN TANPA SUPERVISI



Berdasarkan cara memodifikasi bobotnya, ada 2 macam pelatihan yaitu dengan

- supervisi (*supervised*)
- Tanpa supervisi (*unsupervised*).



PELATIHAN DENGAN DAN TANPA SUPERVISI

- *Supervised*

Dalam pelatihan ini, terdapat sejumlah pasangan data (masukan-keluaran target) yang dipakai untuk melatih jaringan, hingga diperoleh bobot yang diinginkan. Pasangan data tersebut berfungsi sebagai “Guru” untuk melatih jaringan hingga diperoleh bentuk yang terbaik.



- *Unsupervised*

Dalam pelatihan ini tidak ada “Guru” yang akan mengarahkan proses pelatihan. Dalam pelatihannya, perubahan bobot jaringan dilakukan berdasarkan parameter tertentu dan jaringan dimodifikasi menurut ukuran parameter tersebut.

TAKSONOMI JARINGAN SYARAF TIRUAN



Pembagian jaringan syaraf tiruan berdasarkan cara pembelajaran, aplikasi dan jenis arsitektur.

Berdasarkan strategi, model jaringan dibagi menjadi:

- Pelatihan dengan supervisi

Model Hebbian, Perceptron, ADALINE, Boltzman, Hopfield, Backpropagation.

- Pelatihan tanpa supervisi

Model competitive, Hebbian, Kohonen, LVQ (*Learning Vector Quantification*), Neocognitron

TAKSONOMI JARINGAN SYARAF TIRUAN



Berdasarkan arsitekturnya:

- Jaringan layar tunggal
ADALINE, Hopfield, Perceptron, LVQ
- Jaringan layar jamak
MADALINE, Backpropagation, Neocognitron
- Reccurent
BAM (*Bidirectional Associative Memory*), Boltzman Machine, Hopfield

TAKSONOMI JARINGAN SYARAF TIRUAN



Berdasarkan aplikasinya:

- Klasifikasi
ADALINE, LVQ, Bacpropagation
- Pengenalan Pola
ART (Adaptive Resonance Theory), LVQ,
Backpropagation, Neocognitron
- Peramalan
ADALINE, MADALINE, Backpropagation
- Optimasi
ADALINE, Hopfield, Boltzman, Backpropagation

NEURON McCulloch - Pitts

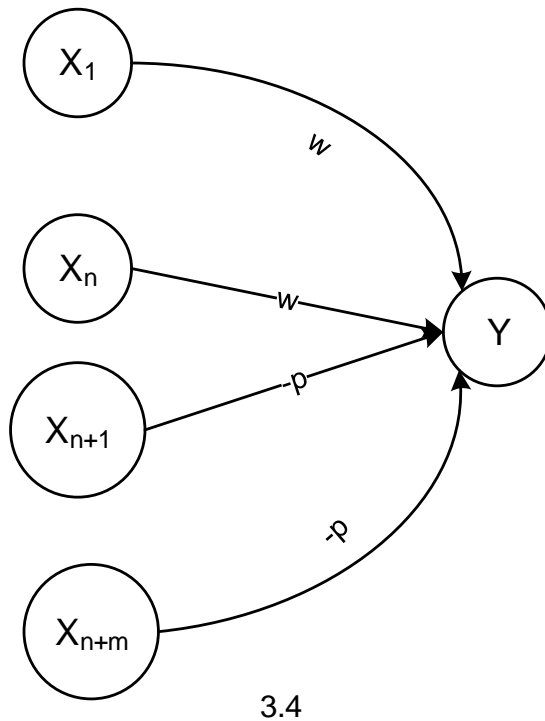


Karakteristiknya:

- Fungsi aktivasinya biner
- Semua garis yang memperkuat sinyal (bobot positif) memiliki kekuatan yang sama (besar bobot). Hal yang sama berlaku untuk garis yang memperlemah sinyal
- Setiap neuron memiliki batas ambang (*threshold*) yang sama.

Apabila total input ke neuron tersebut melebihi *threshold* maka neuron akan meneruskan sinyal.

NEURON McCulloch - Pitts



Model Neuron McCulloch-Pitts.

- Neuron Y menerima sinyal dari X_1, \dots, X_{n+m} buah neuron.
- n buah merupakan memperkuat sinyal.
- m buah memperlemah sinyal.
- Fungsi aktivasi neuron Y :

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

NEURON McCulloch - Pitts

Contoh Soal 1:

Fungsi logika “dan” dengan 2 masukan x_1 dan x_2 akan memiliki keluaran $y = 1$ bila dan hanya bila kedua masukan bernilai 1.

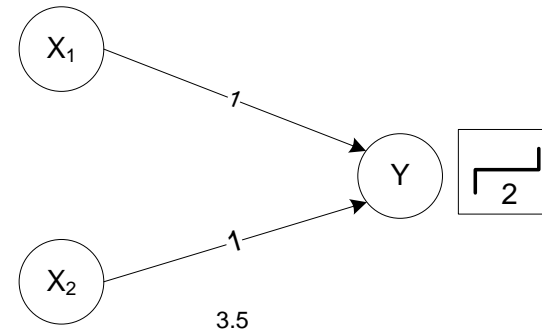
Buatlah model neuron McCulloch-Pitts untuk menyatakan logika “dan” !

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

NEURON McCulloch - Pitts

Jawab:

Model neuron fungsi “dan” tampak pada gambar di samping. Bobot tiap garis adalah = 1 dan fungsi aktivasi memiliki nilai threshold = 2.



x1	x2	$net = \sum_{i=1}^2 x_i w_i$	$y = f(net) = \begin{cases} 1 & \text{jika } net \geq 2 \\ 0 & \text{jika } net < 2 \end{cases}$
0	0	$0.1 + 0.1 = 0$	0
1	0	$1.1 + 0.1 = 1$	0
0	1	$0.1 + 1.1 = 1$	0
1	1	$1.1 + 1.1 = 2$	1

NEURON McCulloch - Pitts

Contoh Soal 2:

Fungsi logika “atau” dengan 2 masukan x_1 dan x_2 akan memiliki keluaran $Y=1$ bila dan hanya bila salah satu masukannya bernilai 1.

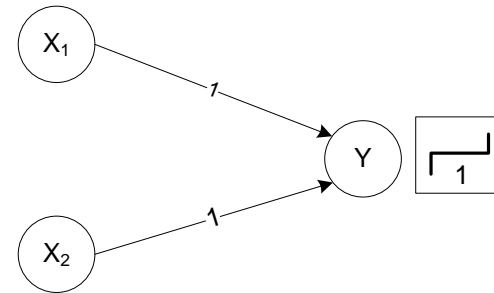
Buatlah model neuron McCulloch-Pitts untuk menyatakan logika “atau” !

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

NEURON McCulloch - Pitts

Jawab:

Model neuron fungsi “dan” tampak pada gambar di samping. Bobot tiap garis adalah = 1 dan fungsi aktivasi memiliki nilai threshold = 2.



x1	x2	$net = \sum_{i=1}^2 x_i w_i$	$y = f(net) = \begin{cases} 1 & \text{jika } net \geq 1 \\ 0 & \text{jika } net < 1 \end{cases}$
0	0	$0.1 + 0.1 = 0$	0
1	0	$1.1 + 0.1 = 1$	1
0	1	$0.1 + 1.1 = 1$	1
1	1	$1.1 + 1.1 = 2$	1



JARINGAN HEBB

- Kelemahan Model McCulloch-Pitts, mengharuskan kita untuk menentukan bobot garis dan bias secara analitik
- Model Hebb memperkenalkan cara menghitung bobot dan bias secara iteratif
- Model Hebb menggunakan aturan supervisi



JARINGAN HEBB

- Model Hebb, dalam setiap iterasi bobot sinapsis dan bias diubah berdasarkan perkalian neuron-neuron dikedua sisinya
- Untuk fungsi aktivasi *threshold* representasi yang sering dipakai adalah bipolar (nilai 1 atau -1)

JARINGAN HEBB

- Untuk jaringan layar tunggal dengan 1 unit keluaran, dimana semua unit masukan x_i terhubung langsung dengan unit keluaran y , maka perubahan nilai bobot dilakukan menggunakan persamaan:

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



JARINGAN HEBB

Algoritma pelatihan Hebb dengan vektor input s dan unit target t adalah sebagai berikut:

1. Inisialisasi semua bobot = $w_i = 0$; ($i = 1, \dots, n$)
 2. Untuk semua vektor input s dan unit target t , lakukan:
 - * Set aktivasi unit masukan $x_i = s_i$; ($i = 1, \dots, n$)
 - ** Set aktivasi unit keluaran : $y = t$
 - *** Perbaiki bobot menurut persamaan
$$w_i (\text{baru}) = w_i (\text{lama}) + \Delta w ; (i = 1, \dots, n), \text{ dengan}$$
$$\Delta w = x_i y$$
 - **** Perbaiki bias menurut persamaan $b (\text{baru}) = b(\text{lama}) + y$
- (NOTE: Perbaiki bias diperlakukan seperti perbaikan bobot)



JARINGAN HEBB

Contoh 1:

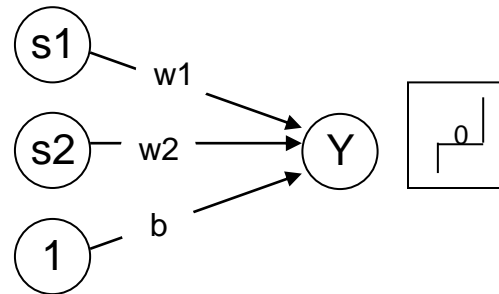
Buatlah jaringan Hebb untuk menyatakan fungsi logika “dan” jika representasi masukan/keluaran yang dipakai adalah:

- a. Biner
- b. Masukan biner dan keluaran bipolar
- c. Masukan dan keluaran bipolar

JARINGAN HEBB

JAWAB :

- a. Gambar arsitektur jaringan Hebb untuk menyatakan fungsi “dan”. Tabel dibawah ini adalah tabel masukan dan target dalam biner.



Masukan			Target
x1	x2	1	t
1	1	1	1
1	0	1	0
0	1	1	0
0	0	1	0

JARINGAN HEBB

- Mula-mula semua bobot dan bias diberi nilai = 0. Untuk setiap data masukan dan target, perubahan bobot dihitung dari perkalian data masukan targetnya.

$$\Delta w_1 = x_1 t$$

$$\Delta w_2 = x_2 t$$

$$\Delta b = 1 \cdot t = t$$

Bobot w_i baru = bobot w_i lama + Δw_i ; ($i=1,2$)

JARINGAN HEBB

- Hasil Iterasi :

Masukan			Target	Perubahan Bobot			Bobot Baru		
x1	x2	1	t	$\Delta w1$	$\Delta w2$	Δb	x1	x2	bias
Inisialisasi							0	0	0
1	1	1	1	1	1	1	1	1	1
1	0	1	0	0	0	0	1	1	1
0	1	1	0	0	0	0	1	1	1
0	0	1	0	0	0	0	1	1	1

JARINGAN HEBB

- Tampak bahwa bobot hanya berubah akibat pasangan data pertama saja. Pada data ke-2 hingga ke-4, tidak ada perubahan bobot karena target = 0 sehingga perubahan bobot (hasil kali masukan dan target = 0).
- Jadi bila dilihat dari tabel, bobot jaringan akhir adalah $w_1=1$, $w_2=1$, dan $b=1$.

$$\text{Net} = \sum_{i=1}^2 w_i x_i + b = 1.x1 + 1.x2 + 1 = x1+x2+1$$

JARINGAN HEBB

x1	x2	$net = \sum_{i=1}^2 x_i w_i + b$	$y = f(net) = \begin{cases} 1 & \text{jika } net \geq 0 \\ 0 & \text{jika } net < 0 \end{cases}$
1	1	$1.1+1.1+1=3$	1
1	0	$1.1+0.1+1=2$	1
0	1	$0.1+1.1+1=2$	1
0	0	$0.1+0.1+1=1$	1

Tampak bahwa nilai $f(net)$ tidak sama dengan target yang dimaksudkan dalam fungsi “dan”. Berarti jaringan tidak dapat mengerti pola yang dimaksudkan.

JARINGAN HEBB

b. Jika target berupa data bipolar, maka tabel masukan dan target :

Masukan			Target
x1	x2	1	t
1	1	1	1
1	0	1	-1
0	1	1	-1
0	0	1	-1

JARINGAN HEBB

- Dengan cara yang sama didapat :

Masukan			Target	Perubahan Bobot			Bobot Baru		
x1	x2	1	t	$\Delta w1$	$\Delta w2$	Δb	x1	x2	bias
Inisialisasi							0	0	0
1	1	1	1	1	1	1	1	1	1
1	0	1	-1	-1	0	-1	0	1	0
0	1	1	-1	0	-1	-1	0	0	-1
0	0	1	-1	0	0	-1	0	0	-2

Note: Diperoleh $w1 = 0$, $w2 = 0$ dan $b = -2$

JARINGAN HEBB

- Jika diujicobakan pada data masukan, maka diperoleh:

x1	x2	$net = \sum_{i=1}^2 x_i w_i + b$	$y = f(net) = \begin{cases} 1 & \text{jika } net \geq 0 \\ -1 & \text{jika } net < 0 \end{cases}$
1	1	$1.0 + 1.0 + (-2) = -2$	-1
1	0	$1.1 + 0.0 + (-2) = -2$	-1
0	1	$0.0 + 1.0 + (-2) = -2$	-1
0	0	$0.0 + 0.0 + (-2) = -2$	-1

Tampak bahwa nilai $f(net)$ tidak sama dengan target yang dimaksudkan dalam fungsi “dan”. **Berarti jaringan tidak dapat mengerti** pola yang dimaksudkan

JARINGAN HEBB

c. Tabel masukan dan target bipolar tampak pada tabel berikut:

Masukan			Target
x1	x2	1	t
1	1	1	1
1	-1	1	-1
-1	1	1	-1
-1	-1	1	-1

JARINGAN HEBB

- Menggunakan cara yang sama seperti jawaban a, diperoleh:

Masukan			Targe t	Perubahan Bobot			Bobot Baru		
x1	x2	1	t	$\Delta w1$	$\Delta w2$	Δb	x1	x2	bias
Inisialisasi							0	0	0
1	1	1	1	1	1	1	1	1	1
1	-1	1	-1	-1	1	-1	0	2	0
-1	1	1	-1	1	-1	-1	1	1	-1
-1	-1	1	-1	1	1	-1	2	2	-2

Note: Diperoleh $w1 = 2$, $w2 = 2$ dan $b = -2$

JARINGAN HEBB

- $\text{Net} = \sum w_i x_i + b = 2.x1 + 2.x2 - 2 = 2.x1 + 2.x2 - 2$

hasil yang diperoleh:

x1	x2	$net = \sum_{i=1}^2 x_i w_i + b$	$y = f(net) = \begin{cases} 1 & \text{jika } net \geq 0 \\ -1 & \text{jika } net < 0 \end{cases}$
1	1	$1.2 + 1.2 + (-2) = 2$	1
1	-1	$1.2 + (-1).2 + (-2) = -2$	-1
-1	1	$(-1).2 + 1.2 + (-2) = -2$	-1
-1	-1	$(-1).2 + (-1).2 + (-2) = -6$	-1

Tampak bahwa nilai $f(net)$ sama dengan target yang dimaksudkan dalam fungsi “dan”. Berarti jaringan mampu mengenali pola yang dimaksudkan

JARINGAN HEBB

TUGAS 1:

Buatlah jaringan hebb dengan 3 masukan dan sebuah target keluaran untuk mengenali pola pada tabel di samping :

Masukan			Target
x1	x2	1	t
1	1	1	1
1	0	1	0
0	1	1	0
0	0	1	0



JARINGAN HEBB UNTUK PENGENALAN POLA

. . . #
. # . # .
. . # . .
. # . # .
. . .

POLA 1

. # # # .
. . . #
. . . #
. . . #
. # # # .

POLA 2



JARINGAN HEBB UNTUK PENGENALAN POLA

PENYELESAIAN:

- Untuk mempresentasikan kasus tersebut dalam Jaringan Hebb, tiap karakter pola dianggap sebagai sebuah unit masukan.
- Misalkan karakter “ # ” $\rightarrow 1$, karakter “ . ” $\rightarrow -1$
- Karena setiap pola terdiri dari 5 baris dan 5 kolom, berarti jaringan Hebb terdiri dari 25 unit masukan ($x_1 \dots x_{25}$)
 - $x_1 \dots x_5 \rightarrow$ karakter baris 1
 - $x_6 \dots x_{10} \rightarrow$ karakter baris 2
 - $x_{21} \dots x_{25} \rightarrow$ karakter baris 5
- Bias bernilai 1
- Target diambil, bernilai 1 jika masukan berupa pola 1. Target bernilai -1 jika masukan berupa pola 2.



JARINGAN HEBB UNTUK PENGENALAN POLA

Masukan															Target
x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	t
1	-1	-1	-1	1	-1	1	-1	1	-1	-1	-1	1	-1	-1	1
x16	x17	x18	x19	x20	x21	x22	x23	x24	x25						
-1	1	-1	1	-1	1	-1	-1	-1	1						
Masukan															Target
x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	t
-1	1	1	1	-1	1	-1	-1	-1	1	1	-1	-1	-1	1	-1
x16	x17	x18	x19	x20	x21	x22	x23	x24	x25						
1	-1	-1	-1	1	-1	1	1	1	-1						

JARINGAN HEBB UNTUK PENGENALAN POLA

- Jika pola pertama dimasukkan, perubahan pola yang terjadi merupakan hasil kali antara target dengan masukan pertama.
- Karena target = 1, maka hasil kali akan sama dengan pola pertama.

x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15
1	-1	-1	-1	1	-1	1	-1	1	-1	-1	-1	1	-1	-1
x16	x17	x18	x19	x20	x21	x22	x23	x24	x25					
-1	1	-1	1	-1	1	-1	-1	-1	1					

Bobot bias adalah = 1

JARINGAN HEBB UNTUK PENGENALAN POLA

- Perkalian masukan kedua dengan targetnya menghasilkan Δw_i ($i = 1..25$)

x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15
1	-1	-1	-1	1	-1	1	-1	1	-1	-1	-1	1	-1	-1
x16	x17	x18	x19	x20	x21	x22	x23	x24	x25					
-1	1	-1	1	-1	1	-1	-1	-1	1					

dan perubahan bobot bias = $\Delta b = (-1).1 = -1$

JARINGAN HEBB UNTUK PENGENALAN POLA



- Jika Δw_i ditambahkan ke bobot jaringan hasil pola pertama maka diperoleh bobot final, $w =$
2 -2 -2 -2 2; -2 2 0 2 -2; -2 0 2 0 -2;
-2 2 0 2 -2; 2 -2 -2 -2 2
Bobot bias = $1 + (-1) = 0$

JARINGAN HEBB UNTUK PENGENALAN POLA



- Untuk pola 1

$$\begin{aligned} \text{Net} = & \{1(2)+(-1)(-2)+(-1)(-2)+(-1)(-2) + (1)(2)\} + \{(-1)(-2)+1(2)+(-1)0+1(2)+(-1)(-2)\} + \{(-1)(-2) + (-1)0+1(2)+(-1)0+(-1)(-2)\} + \{(-1)(-2)+1(2)+(-1)0+1(2)+(-1)(-2)\} + \\ & \{1(2)+(-1)(-2)+(-1)(-2)+(-1)(-2)+1(2)\} = 42. \text{ Maka } f(\text{net}) \\ & = 1 \end{aligned}$$

JARINGAN HEBB UNTUK PENGENALAN POLA



- Untuk pola 2

$$\text{Net} = \{(-1)(2)+1(-2)+1(-2)+1(-2)+1(-2)\} + \{1(-2)+(-1)2+(-1)0+(-1)2+1(-2)\} + \{1(-2)+(-1)0+(-1)2+(-1)0+1(-2)\} + \{1(-2)+(-1)2+(-1)0+(-1)2+1(-2)\} + \{(-1)2+1(-2)+1(-2)+1(-2)+(-1)2\} = -42.$$

$$\text{Maka } f(\text{net}) = -1$$

“Tampak bahwa untuk kedua pola, keluaran jaringan sama dengan target yang diinginkan. Berarti jaringan telah mengenali pola.”

JARINGAN HEBB



- **TUGAS 2:**

Apakah jaringan Hebb dapat membedakan 2 macam pola seperti berikut ini:

# . . . #	. . # . .
. # . # .	. . # . .
. . # . .	# # # # #
. # . # .	. . # . .
# . . . #	. . # . .

JARINGAN PERCEPTRON



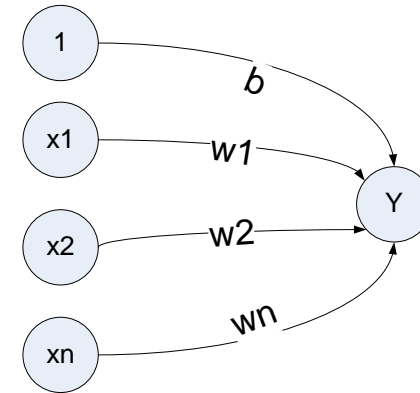
Model jaringan perceptron ditemukan oleh Rosenbalt (1962) dan Minsky – Papert (1969). Model tersebut merupakan model yang memiliki aplikasi dan pelatihan yang paling baik pada era tersebut

JARINGAN PERCEPTRON

Arsitektur Jaringan

- Jaringan terdiri dari beberapa unit masukan (ditambah satu bias), dan memiliki sebuah unit keluaran.
- Fungsi aktivasi mempunyai kemungkinan nilai -1,0,1
- Harga threshold :

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



JARINGAN PERCEPTRON

Algoritma pelatihan perceptron lebih baik dibandingkan model Hebb, karena:

1. Setiap kali sebuah pola dimasukkan, hasil keluaran jaringan dibandingkan dengan target yang sesungguhnya.
2. Modifikasi bobot tidak hanya ditentukan oleh perkalian antara target dengan masukan, tapi juga melibatkan suatu laju pemahaman (*learning rate*) yang besarnya bisa diatur.
3. 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).

JARINGAN PERCEPTRON



Pelatihan Perceptron

Misalkan:

1. S adalah vektor masukan dan t adalah target keluaran
2. α adalah learning rate yang ditentukan (laju pemahaman)
3. Θ adalah threshold yang ditentukan

JARINGAN PERCEPTRON



Algoritma pelatihan perceptron adalah:

1. Inisialisasi semua bobot dan bias (umumnya $w_i = b = 0$)
Tentukan laju pemahaman ($=\alpha$). Untuk penyederhanaan biasanya diberi nilai 1.
2. Selama ada elemen vektor masukan yang respon unit keluarannya tidak sama dengan target, lakukan:

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

>> Hitung respon unit keluaran :

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

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



JARINGAN PERCEPTRON

>> Perbaiki bobot pola yang mengandung kesalahan ($y \neq t$) menurut persamaan:

** $w_i (\text{baru}) = w_i (\text{lama}) + \Delta w$; ($i=1, \dots, n$) dengan

$$\Delta w = \alpha t x_i$$

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



JARINGAN PERCEPTRON

Ada beberapa hal yang perlu diperhatikan dalam algoritma perceptron:

1. Iterasi dilakukan terus hingga semua pola memiliki keluaran jaringan yang sama dengan targetnya (jaringan sudah memahami pola).
2. Iterasi tidak berhenti setelah semua pola dimasukkan seperti yang terjadi pada model Hebb

JARINGAN PERCEPTRON



- Perubahan bobot hanya dilakukan pada pola yang mengandung kesalahan (keluaran jaringan \neq target). Perubahan tersebut merupakan hasil kali unit masukan dengan target laju pemahaman.
- Kecepatan iterasi ditentukan pula oleh laju pemahaman (α), dimana $0 \leq \alpha \leq 1$

JARINGAN PERCEPTRON



Contoh 1:

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

JARINGAN PERCEPTRON

Tabel masukan dan target fungsi logika “dan” keluaran bipolar :

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

Masukan			Target
x1	x2	1	t
1	1	1	1
1	-1	1	-1
-1	1	1	-1
-1	-1	1	-1

JARINGAN PERCEPTRON



Iterasi untuk seluruh pola yang ada disebut epoch 1,
adalah sebagai berikut:

Masukan			Target	net	Y = f (net)	Perubahan Bobot			Bobot Baru		
x1	x2	1	t			Δw1	Δw2	Δb	w1	w2	bias
Inisialisasi Baru									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

Iterasi untuk seluruh pola yang ada disebut epoch 2, adalah sebagai berikut:

Masukan			Target	net	Y = f (net)	Perubahan Bobot			Bobot Baru		
x1	x2	1	t			$\Delta w1$	$\Delta w2$	Δb	w1	w2	bias
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

JARINGAN PERCEPTRON



Pada input pertama $(x_1 \ x_2 \ 1) = (1 \ 1 \ 1)$. Harga net dihitung berdasarkan bobot yang sudah ada sebelumnya yaitu $(w_1 \ w_2 \ b) = (0 \ 0 \ 0)$. Maka net $= \sum x_i w_i + b = 1(0) + 1(0) + 0 = 0 \rightarrow f(\text{net}) = f(0) = 0$.

Keluaran jaringan ($=f(\text{net}) = 0$) tidak sama dengan target yang diinginkan (dalam iterasi ini adalah 1), maka bobot diubah menggunakan rumusan $\Delta w = \alpha t x_i = t x_i$, (karena $\alpha=1$). Bobot baru = bobot lama + Δw .

JARINGAN PERCEPTRON

Input pola kedua dan seterusnya dihitung secara analog. Pada pola terakhir $(x_1 \ x_2 \ 1) = (-1 \ -1 \ 1)$, harga $f(\text{net}) = -1$ yang sama dengan targetnya. Maka bobot tidak diubah. Hal ini dinyatakan dengan kondisi $\Delta w = 0$.

JARINGAN PERCEPTRON

TUGAS 1:

Diketahui perceptron dengan 3 masukan biner x_1 , x_2 , x_3 sebuah bias dan sebuah keluaran bipolar.

Carilah bobot yang akan mengenali pola sebagai berikut :

- a. target keluaran bernilai 1 apabila semua masukan bernilai 1 dan
- b. target bernilai -1 apabila tepat salah satu dari masukan bernilai 0 (tidak diketahui bagaimana target apabila ada 2 atau lebih masukan yang bernilai 0).

Gunakan bobot awal dan bias = 0 dengan laju pemahaman $\alpha=1$; threshold $(\Theta)=0.1$.

JARINGAN PERCEPTRON

PENGENALAN POLA KARAKTER

Perceptron dapat pula dipakai untuk mengenali pola karakter. Dengan berbagai pola masukan yang menyerupai huruf-huruf alfabet, perceptron dapat dilatih untuk mengenalinya

JARINGAN PERCEPTRON

Algoritma untuk mengenali apakah pola masukan yang diberikan menyerupai sebuah karakter tertentu (misal 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 inisialisai bobot, laju pemahaman dan threshold
4. Lakukan proses pelatihan perceptron

JARINGAN PERCEPTRON

.	.	#	#	.	.	.
.	.	.	#	.	.	.
.	.	.	#	.	.	.
.	.	#	.	#	.	.
.	.	#	.	#	.	.
.	#	#	#	#	#	.
.	#	.	.	.	#	.
.	#	.	.	.	#	.
#	#	#	.	#	#	#

POLA 1

.	.	.	#	.	.	.
.	.	.	#	.	.	.
.	.	.	#	.	.	.
.	.	#	.	#	.	.
.	.	#	.	#	.	.
.	#	#	#	#	#	.
.	#	.	.	.	#	.
.	#	.	.	.	#	.
.	#	.	.	.	#	.

POLA 4

#	#	#	#	#	#	.
.	#	#
.	#	#
.	#	#
.	#	#	#	#	#	.
.	#	#
.	#	#
.	#	#
#	#	#	#	#	#	.

POLA 2

#	#	#	#	#	#	.
#	#
#	#
#	#
#	#	#	#	#	#	.
#	#
#	#
#	#
#	#	#	#	#	#	.

POLA 5

.	.	#	#	#	#	#
.	#	#
#
#
#
#
#
.	#	#
.	.	#	#	#	#	#

POLA 3

.	.	#	#	#	.	.
.	#	.	.	.	#	.
#	#
#
#
#
#	#
.	#	.	.	.	#	.
.	.	#	#	#	.	.

POLA 6

JARINGAN PERCEPTRON

Misal, kita akan membuat model perceptron untuk mengenali pola “A” !

vektor masukan pola 1

	k1	k2	k3	k4	k5	k6	k7
b1	-1	-1	1	1	-1	-1	-1
b2	-1	-1	-1	1	-1	-1	-1
b3	-1	-1	-1	1	-1	-1	-1
b4	-1	-1	1	-1	1	-1	-1
b5	-1	-1	1	-1	1	-1	-1
b6	-1	1	1	1	1	1	-1
b7	-1	1	-1	-1	-1	1	-1
b8	-1	1	-1	-1	-1	1	-1
b9	1	1	1	-1	1	1	1



JARINGAN PERCEPTRON

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

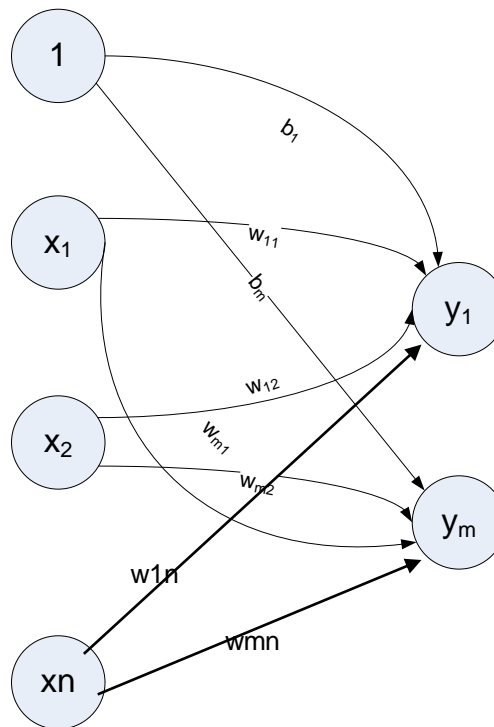
- Maka perceptron yang dipakai untuk mengenali pola huruf “A” (atau bukan “A”) memiliki 63 unit masukan, sebuah bias dan sebuah unit keluaran.
- Misalkan bobot awal diambil = 0 untuk semua bobot maupun bias, laju pemahaman diambil $\alpha = 1$ dan threshold = 0.5
- Pelatihan dilakukan dengan cara memasukkan 63 unit masukan (atau sebuah pola huruf). Dihitung . Berikutnya, fungsi aktivasi dihitung menggunakan persamaan :

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

- Apabila $f(net) \neq target$, 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 (missal “A” atau bukan “A”, “B” atau bukan “B”, dan seterusnya) dilakukan dengan cara menggabungkan beberapa model perceptron.



Setiap unit masukan dihubungkan dengan setiap unit target. Bobot penghubung dari unit x_i ke y_i 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 pelatihan perceptron untuk pengenalan beberapa pola sekaligus adalah sebagai berikut:

- Nyatakan tiap pola masukan sebagai vektor bipolar yang elemennya adalah tiap titik dalam pola tersebut.
- Berikan nilai target $t_j = +1$ jika pola masukan menyerupai huruf yang diinginkan. Jika sebaliknya, berikan nilai target $= t_j = -1$ ($j = 1, 2, \dots, m$)
- Berikan inisialisasi bobot, bias, laju pemahaman dan threshold
- lakukan proses pelatihan perceptron :



JARINGAN PERCEPTRON

- a. Hitung respon unit keluaran ke-j :

$$net_j = \sum x_i w_{ji} + b_j$$

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

- 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

$$t_j = y_j; j = 1, \dots, m$$

JARINGAN PERCEPTRON



- Contoh soal :

Diketahui 6 buah pola masukan seperti gambar sebelumnya. Buatlah model perceptron untuk mengenali pola “A”, “B”, dan “C”.

JARINGAN PERCEPTRON



Penyelesaian:

Mula-mula dibuat 6 buah vektor masukan seperti sebelumnya. Ada 3 buah vektor keluaran yang masing-masing menyatakan bahwa pola menyerupai huruf “A”, “B”, dan “C”. Vektor target kemudian ditentukan seperti berikut :



JARINGAN PERCEPTRON

Pola Masukan	t1	t2	t3
Pola 1	1	-1	-1
Pola2	-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, dengan beberapa perubahan sebagai berikut:

- Vektor masukan terdiri dari 63 elemen x_1, x_2, \dots, x_{63} dan sebuah bias b
- Ada 3 target t_1, t_2, t_3
- Ada 3 kolom net yaitu net_1, net_2 dan net_3 , masing-masing merupakan hasil kali bobot dengan vektor masukan
- Ada 3 buah y yang merupakan fungsi aktivasi ketiga kolom net.
 $y_1=f(net_1), y_2=f(net_2), y_3=f(net_3)$
- Ada 3×63 buah kolom perubahan bobot yaitu
- $\Delta w_{11}, \Delta w_{12}, \dots, \Delta w_{1.63}, \quad \Delta w_{21}, \Delta w_{22}, \dots, \Delta w_{2.63}, \quad \dots$
- $\Delta w_{31}, \Delta w_{32}, \dots, \Delta w_{3.63}$
- Ada 3×63 buah bobot, masing-masing:
- $w_{11}, w_{12}, \dots, w_{1.63}, w_{21}, w_{22}, \dots, w_{2.63}, \dots, w_{31}, w_{32}, \dots, w_{3.63}$



PELATIHAN ADALINE

Model ADALINE (Adaptive Linear Neuron) ditemukan oleh Widrow dan Hoff (1960). Arsitekturnya mirip dengan perceptron.

Perbedaan dengan Perceptron adalah dalam hal cara memodifikasi bobotnya. Bobot dimodifikasi dengan aturan DELTA (*Least Mean Square*).

- Fungsi aktivasi yang dipakai :

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

$$y = f(\text{net}) = \sum_i x_i w_i + b$$

- Kuadrat selisih antara target (t) dan keluaran jaringan ($f(\text{net})$) merupakan error yang terjadi. Dalam aturan delta, bobot dimodifikasi sedemikian hingga errornya minimum.

$$E = (t - f(\text{net}))^2 = [t - (\sum x_i w_i + b)]^2$$



- E merupakan fungsi bobot w_i . Penurunan E tercepat terjadi pada arah:

$$\frac{\partial E}{\partial w_i} = -2[t - (\sum x_i w_i + b)] x_i = -2(t - y)x_i$$

Maka perubahan bobot adalah :

$$\Delta w_i = \alpha(t - y)x_i$$

dengan α merupakan bilangan positif kecil (umumnya diambil = 0,1)

Algoritma pelatihan ADALINE adalah sebagai berikut:

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

Tentukan laju pemahaman ($=\alpha$). Untuk penyederhanaan biasanya α diberi nilai kecil ($=0,1$). Tentukan toleransi kesalahan yang diijinkan

2. Selama $\max \Delta w_i >$ batas toleransi lakukan:

- a. Set aktivasi unit masukan $x_i = s_i$ ($i=1,..,n$)
- b. Hitung respon unit keluaran :
$$\text{net} = \sum_i x_i w_i + b$$
$$y = f(\text{net}) = \text{net}$$
- c. Perbaiki bobot pola yang mengandung kesalahan ($y \neq t$), menurut persamaan:
** $w_i (\text{baru}) = w_i (\text{lama}) + \alpha(t-y)x_i$
** $b (\text{baru}) = b(\text{lama}) + \alpha(t-y)$

Setelah proses pelatihan selesai, ADALINE dapat dipakai untuk pengenalan pola. Untuk itu, umumnya dipakai fungsi threshold bipolar. Caranya adalah sebagai berikut:

1. Inisialisasi semua bobot dan bias dengan bobot dan bias hasil pelatihan
2. Untuk setiap input masukan bipolar x , lakukan :
 - a. Set aktivasi unit masukan $x_i = s_i$
($i=1, \dots, n$)

b. Hitung net vektor keluaran :

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

c. Kenakan fungsi aktivasi :

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

CONTOH :

Gunakan model ADALINE untuk mengenali pola fungsi logika “dan” dengan masukan dan target bipolar:

(Gunakan batas toleransi = 0.05 dan $\alpha=0.1$)

Masukan		Target
x1	x2	t
1	1	1
1	-1	-1
-1	1	-1
-1	-1	-1

PENYELESAIAN:

Dengan $\alpha = 0.1$, maka perubahan bobotnya:

$$\begin{aligned}\Delta w_i &= 0.1(t - f(\text{net}))x_i \\ &= 0.1(t - y) x_i\end{aligned}$$

Iterasi untuk epoch-1 tampak pada tabel berikut:

Masukan			t	net	Y = f(net)	t-y	Perubahan Bobot			Bobot Baru		
x1	x2	1					Δw1	Δw2	Δb	w1	w2	b
Inisialisasi										0	0	0
1	1	1	1	0	1	1	0,1	0,1	0,1	0,1	0,1	0,1
1	-1	1	-1	0,1	1	-1,1	-0,11	0,11	-0,11	-0,01	0,21	-0,01
-1	1	1	-1	0,21	1	-1,21	0,12	-0,12	-0,12	0,11	0,09	-0,13
-1	-1	1	-1	-0,33	-1	-0,67	0,07	0,07	-0,07	0,18	0,16	-0,2

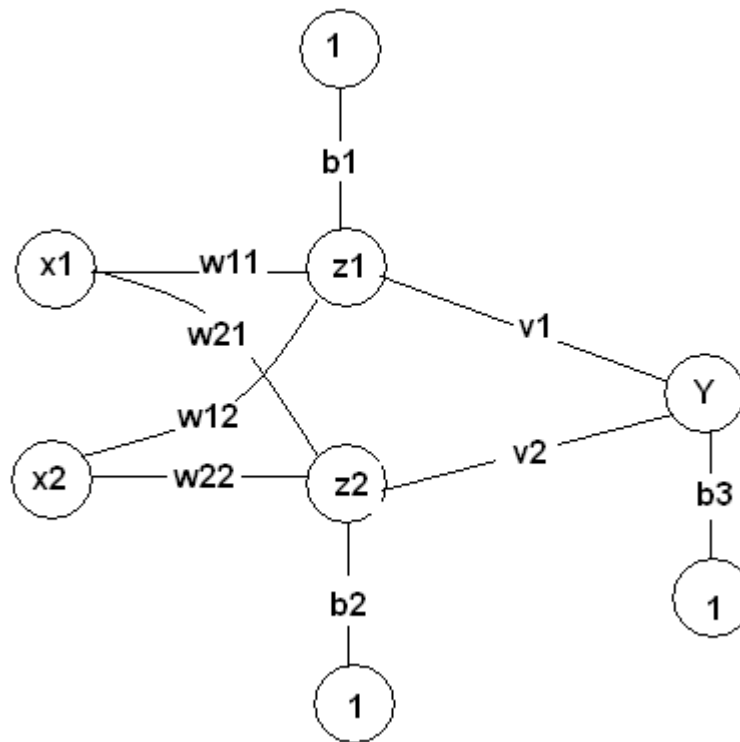
Maksimum $\Delta w_i = 0,07 > \text{toleransi}$, maka lanjut ke epoch 2

Masukan			t	net	Y = f(net)	t-y	Perubahan Bobot			Bobot Baru		
x1	x2	1					$\Delta w1$	$\Delta w2$	Δb	w1	w2	b
Inisialisasi										0,18	0,16	-0,2
1	1	1	1	0,14	0,14	0,86	0,09	0,09	0,09	0,26	0,24	-0,11
1	-1	1	-1	-0,09	-0,09	-0,91	-0,09	0,09	-0,09	0,17	0,33	-0,2
-1	1	1	-1	-0,04	-0,04	-0,96	0,1	-0,1	-0,1	0,27	0,24	-0,3
-1	-1	1	-1	-0,8	-0,8	-0,2	0,02	0,02	-0,02	0,29	0,26	-0,32

Masukan		net	f(net)
x1	x2		
1	1	0,23	1
1	-1	-0,29	-1
-1	1	-0,35	-1
-1	-1	-0,87	-1

PELATIHAN MADALINE

Beberapa ADALINE dapat digabungkan untuk membentuk suatu jaringan baru → MADALINE (many ADALINE).



Algoritma Pelatihan MADALINE:

1. Inisialisasi bobot dan bias dengan bilangan acak kecil. Inisialisasi laju pemahaman (α) dengan bilangan kecil.
2. Selama perubahan bobot lebih besar dari toleransi atau jumlah epoch sebelum melebihi batas yang ditentukan, lakukan tahap 2.1 sampai dengan 2.5:
 - 2.1 Set aktivasi unit masukan : $x_i = s_i$ untuk semua i .
 - 2.2 Hitung net input untuk setiap unit tersembunyi

ADALINE (z_1, z_2, \dots, z_n)

$$zin_j = b_j + \sum_i x_i w_{ji}$$

2.3 Hitung keluaran setiap unit tersembunyi dengan menggunakan fungsi aktivasi bipolar :

$$z_{in_j} = b_j + \sum_i x_i w_{ji}$$

2.4 Tentukan keluaran jaringan:

$$Y_{in} = b_k + \sum_j x_j w_{ji}$$

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

2.5 Hitung error yang ditentukan perubahan bobot

Jika $y = \text{target}$, maka tidak dilakukan perubahan bobot

Jika $y \neq \text{target}$:

- Untuk $t = 1$, ubah bobot ke unit z_j yang z_{in} nya terdekat dengan 0 (misal ke unit z_p) sebagai berikut:

$$b_p \text{ baru} = b_p \text{ lama} + \alpha (1 - Z_{in-j})$$

$$w_{ji} \text{ baru} = w_{pi} \text{ lama} + \alpha (1 - z_{in-p}) x_i$$

- Untuk $t = -1$, ubah semua bobot ke unit z_k yang z_{in} nya positif sebagai berikut:

$$b_k \text{ baru} = b_k \text{ lama} + \alpha (-1 - z_{in-k})$$

$$w_{ki} \text{ baru} = w_{ki} \text{ lama} + \alpha (-1 - z_{in-k}) x_i$$

- Jika $y \neq t$ dan $t = 1$ (karena y bipolar berarti $y = -1$), maka $f(\text{net})$ di $z_1 = z_2 = -1$. Untuk mengenali pola, paling sedikit salah satu dari z tersebut harus dijadikan $= 1$. Bobot yang dimodifikasi adalah bobot ke unit z yang netna paling dekat dengan 0. Misalkan unit z yg keluarannya terdekat dengan 0 adalah z_p . Maka bobot dan bias akan dimodifikasi menurut persamaan:

$$b_p \text{ baru} = b_p \text{ lama} + \alpha(1 - z_{in-p})$$

$$w_{pi} \text{ baru} = w_{pi} \text{ lama} + \alpha(1 - z_{in-p})x_i$$

- Jika $y \neq t$ dan $t = -1$ (dengan kata lain $y = 1$), berarti minimal salah satu z memiliki $f(\text{net}) = 1$ (atau dala z yang netnya positip). Padahal semua z harus memiliki $f(\text{net}) = -1$. Maka bobot yang dimodifikasi adalah bobot yang netnya positip (misal unit z_k) menurut persamaan:

$$b_k \text{ baru} = b_k \text{ lama} + \alpha(-1 - z_{in-k})$$

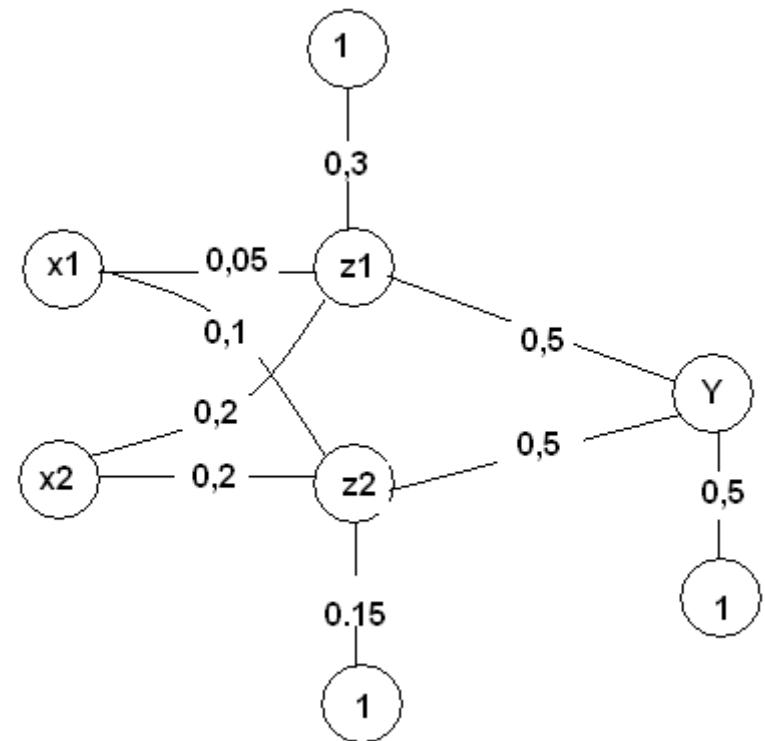
$$w_{ki} \text{ baru} = w_{ki} \text{ lama} + \alpha(-1 - z_{in-k})x_i$$

Contoh :

Fungsi logika XOR memiliki pasangan masukan target seperti yang tampak pada tabel berikut:

Masukan		Target
X1	X2	t
1	1	-1
1	-1	1
-1	1	1
-1	-1	-1

Ke unit tersembunyi		
Dari unit	Z1	Z2
x1	$w_{11} = 0,05$	$w_{21}=0,1$
x2	$w_{12}=0,2$	$w_{22}=0,2$
bias	$b_1=0,3$	$b_2=0,15$



Pola 1:

Masukan $x_1=1$, $x_2=1$, $t = -1$

Algoritma :

2.2 Hitung net untuk unit tersembunyi z_1 dan z_2 :

$$\begin{aligned} \text{zin_1} &= b_1 + x_1w_{11} + x_2w_{12} \\ &= 0,3 + 1 (0,05) + 1(0,2) = 0,55 \end{aligned}$$

$$\begin{aligned} \text{zin_2} &= b_2 + x_1w_{21} + x_2w_{22} \\ &= 0,15 + 1 (0,1) + 1 (0,2) = 0,45 \end{aligned}$$

2.3 Hitung keluaran unit tersembunyi z1 dan z2 menggunakan fungsi aktivasi bipolar. Didapat:

$$z1 = f(zin_1) = 1$$

$$z2 = f(zin_2) = 1$$

2.4 Tentuan keluaran jaringan y :

$$\begin{aligned} y_in &= b3 + z1v1 + z2v2 \\ &= 0,5 + 1 (0,5) + 1 (0,5) \\ &= 1,5 \end{aligned}$$

$$\text{Maka } y = f(y_in) = 1$$

$$2.5 \quad t-y = -1 - 1 = -2, \neq 0 \text{ dan } t = -1$$

Maka semua bobot yang menghasilkan z_{in} yang positif dimodifikasi. Karena $z_{in_1} > 0$ dan $z_{in_2} > 0$, maka semua bobot dimodifikasi :

$$b1 \text{ baru} = b1 \text{ lama} + \alpha (-1 - z_{in_1}) = 0,3 + 0,5 (-1 - 0,55) = -0,475$$

$$\begin{aligned} w11 \text{ baru} &= w11 \text{ lama} + \alpha (-1 - z_{in_1}) x1 = 0,05 + 0,5 (-1 - 0,55) \\ &= -0,725 \end{aligned}$$

$$\begin{aligned} w12 \text{ baru} &= w12 \text{ lama} + \alpha (-1 - z_{in_1}) x2 = 0,2 + 0,5 (-1 - 0,55) \\ &= -0,575 \end{aligned}$$

Perubahan bobot ke unit tersembunyi:

$$b2 \text{ baru} = b2 \text{ lama} + \alpha (-1 - z_{in_2}) = 0,15 + 0,5 (-1 - 0,45) = -0,575$$

$$\begin{aligned} w21 \text{ baru} &= w21 \text{ lama} + \alpha (-1 - z_{in_2}) x1 = 0,1 + 0,5 (-1 - 0,45) \\ &= -0,625 \end{aligned}$$

$$\begin{aligned} w22 \text{ baru} &= w22 \text{ lama} + \alpha (-1 - z_{in_2}) x2 = 0,2 + 0,5 (-1 - 0,45) \\ &= -0,525 \end{aligned}$$

Bobot mula-mula		Bobot setelah iterasi	$ \Delta w $
b1	0,3	-0,475	0,775
w11	0,05	-0,725	0,73
w12	0,2	-0,575	0,775
b2	0,15	-0,575	0,725
w21	0,1	-0,625	0,725
w22	0,2	-0,525	0,725

Karena masih ada (bahkan semua) perubahan bobot $>$ toleransi yang ditetapkan, maka iterasi dilanjutkan untuk pola2.

Iterasi dilakukan untuk semua pola. Apabila ada perubahan bobot yang masih lebih besar dari batas toleransi, maka iterasi dilanjutkan untuk epoch 2 dan seterusnya.

BACK PROPAGATION

- JST dengan layar tunggal memiliki keterbatasan dalam pengenalan pola. Kelemahan ini bisa ditanggulangi dengan menambahkan satu/beberapa layar tersembunyi diantara layar masukan dan keluaran, tetapi memerlukan waktu yang lama.

- Arsitektur Jaringan:
- Fungsi Aktivasi :
Fungsi sigmoid biner memiliki range (0,1)
- Pelatihan Standar Backpropagation
 - Fase 1. Propagasi maju
 - Fase 2. Propagasi mundur
 - Fase 3. Perubahan bobot

Algoritma pelatihan untuk jaringan dengan satu layer tersembunyi

Langkah 0 : Inisialisasi semua bobot dengan bilangan acak kecil

Langkah 1 : Jika kondisi penghentian belum terpenuhi, lakukan langkah 2 – 9

Fase I : Propagasi Maju

Langkah 3: Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya

Langkah 4 : Hitung semua keluarandi unit tersembunyi z_j ($j = 1, 2, \dots, p$)

$$z_{net_j} = v_{jo} + \sum_{i=1}^n x_i w_{ij}$$

$$z_j = f(z_{net_j}) = \frac{1}{1 + e^{-z_{net_j}}}$$

Langkah 5 : Hitung semua keluaran jaringan di unit y_k ($k = 1, 2, \dots, m$)

$$y_{net_k} = w_{ko} + \sum_{j=1}^k z_j w_{kj} \quad \left| \quad y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}}\right.$$