



ARTIFICIAL NEURAL NETWORK

OUTLINE

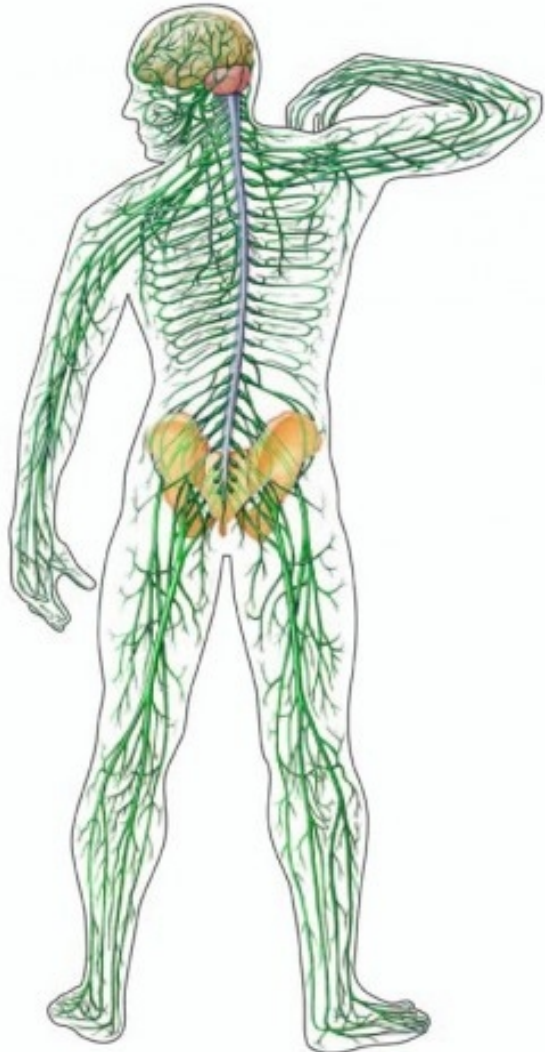
- ❖ Pengantar Neural Network
- ❖ Representasi NN
- ❖ Karakteristik task untuk NN
- ❖ Perceptron



ARTIFICIAL NEURAL NETWORK



JARINGAN SARAF MANUSIA



- ❖ Sistem saraf bekerja melalui jaringan interkoneksi miliaran neuron, 86 milyar neuron diantaranya ada di otak manusia
- ❖ Neuron ini mengirimkan informasi dalam bentuk impuls saraf
- ❖ Bertanggung jawab untuk mengendalikan dan mengkoordinasikan semua fungsi tubuh
- ❖ Sistem saraf manusia yang bersangkutan dengan menerima informasi dari dunia luar, pengolahan, dan kemudian menghasilkan respon yang tepat
- ❖ Jaringan yang mengontrol dan mengkoordinasikan semua kegiatan tubuh, dengan mengirimkan pesan atau sinyal dari otak ke bagian-bagian berbeda dari tubuh dan sebaliknya

STRUKTUR SEL

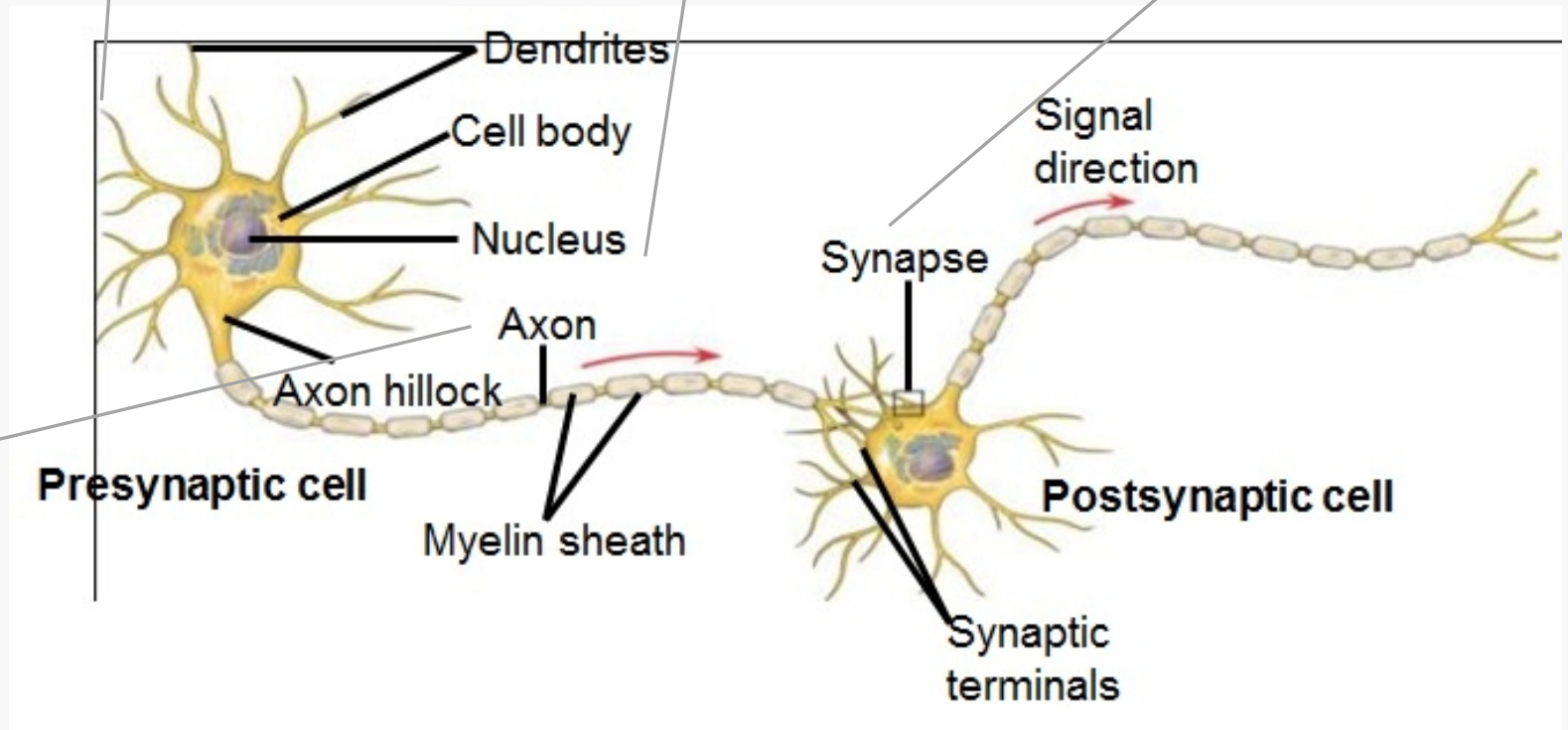
Neuron pada manusia, terdiri dari sel tubuh, sebuah axon dan banyak dendrit.

Axon : penghubung yang mengirimkan sinyal keluaran neuron menuju neuron lainnya

Dendrit : Fasilitator yang menghubungkan neuron dengan axon dari neuron lainnya

Nukleus : Inti sel pengolah informasi

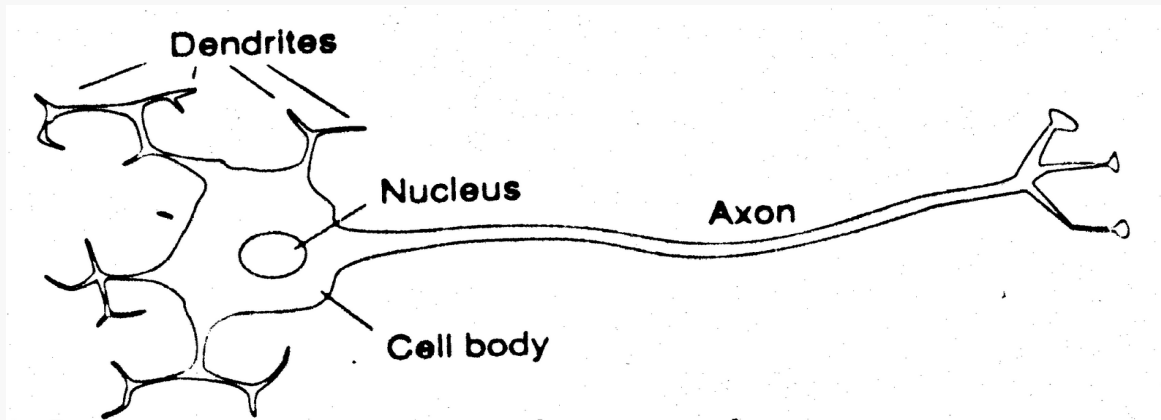
Sinapsis : Berfungsi sebagai persimpangan, di mana impuls atau informasi dapat mengalir dari satu neuron ke yang lain



CARA KERJA SISTEM SARAF

- ❖ Neuron sensorik mengirimkan rangsangan atau impuls yang diterima dari alat indera (mata, hidung, kulit) ke sistem saraf pusat (otak dan sumsum tulang belakang)
- ❖ Interneuron berfungsi membaca impuls yang diterima dari neuron sensorik dan memutuskan respon yang akan dihasilkan
- ❖ Otak memproses rangsangan tersebut dan mengirimkannya kembali ke bagian lain dari tubuh, memberitahu mereka bagaimana bereaksi terhadap jenis tertentu dari stimulus.
- ❖ Motor neuron bertanggung jawab untuk menerima sinyal dari saraf otak dan tulang belakang, dan mengirim mereka ke bagian lain dari tubuh

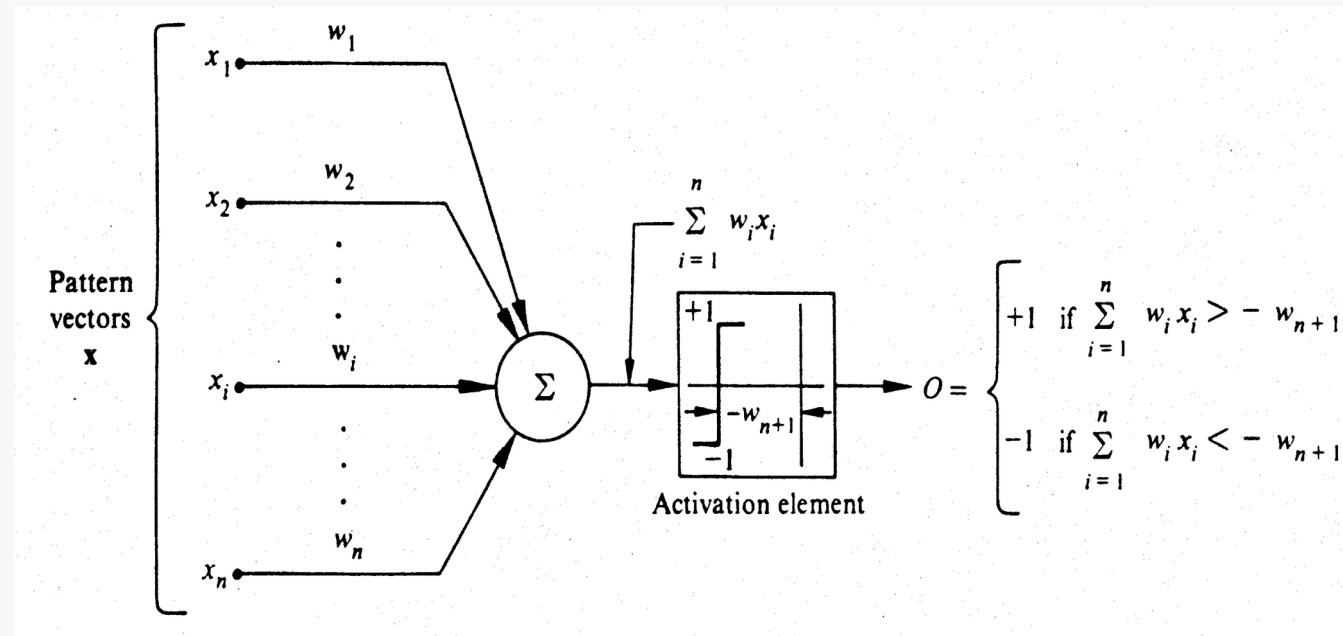
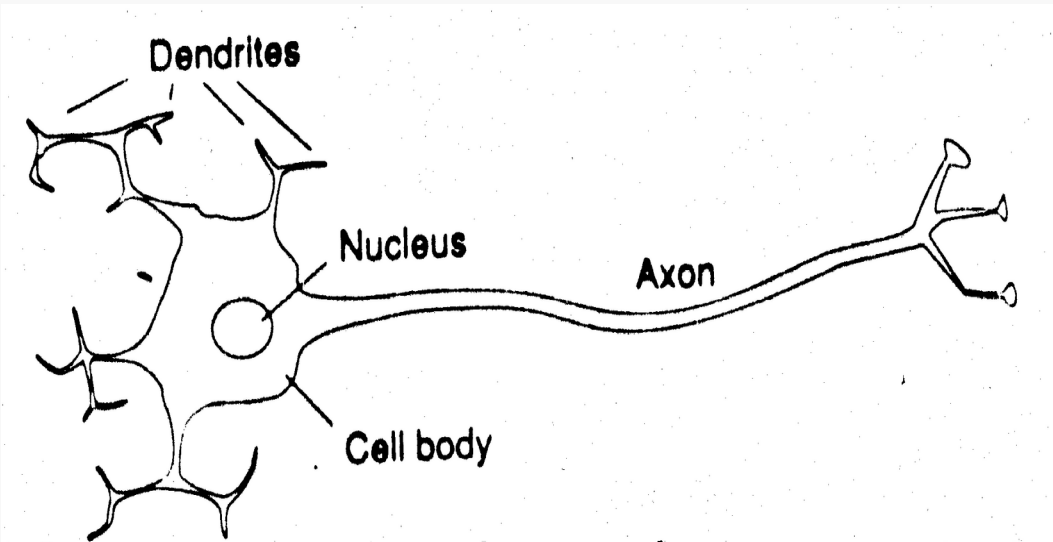
CARA KERJA NEURON



Cara kerja dari neuron adalah menggabungkan pengaruh dari semua masukan dan mempertimbangkan apakah pengaruh dari masukan tadi dapat mencapai threshold yang telah ditetapkan atau tidak. Jika nilai threshold tadi tercapai, maka neuron akan memproduksi keluaran dalam bentuk suatu pulsa tertentu yang diproses dari sel hingga dikeluarkan melalui axon. Jika hal ini terjadi, maka neuron sedang dalam keadaan aktif.

ARTIFICIAL NEURAL NETWORK

Artificial neural network (Neural Network) adalah sebuah sistem untuk pemrosesan informasi dengan “meniru” cara kerja sistem saraf biologis



NEURON PADA ARTIFICIAL NEURAL NETWORK

- ❖ Tiap penghubung diasosiasikan dengan sebuah nilai bobot (w). Seperti pada sebuah sinapsis, nilai bobot menentukan derajat pengaruh dari sebuah neuron ke neuron yang lainnya. Pengaruh dari sebuah neuron ke neuron yang lainnya merupakan hasil kali dari nilai keluaran dari neuron-neuron yang masuk ke neuron (x) dengan nilai bobot (w) yang menghubungkan neuron-neuron tadi.
- ❖ Tiap neuron dikombinasikan dengan sebuah fungsi aktivasi yang berfungsi sebagai penghubung dari penjumlahan semua nilai masukan dengan nilai keluarannya. Keluaran dari neuron inilah yang nantinya akan menentukan apakah sebuah neuron itu aktif ataukah tidak.



REPRESENTASI NEURAL NETWORK

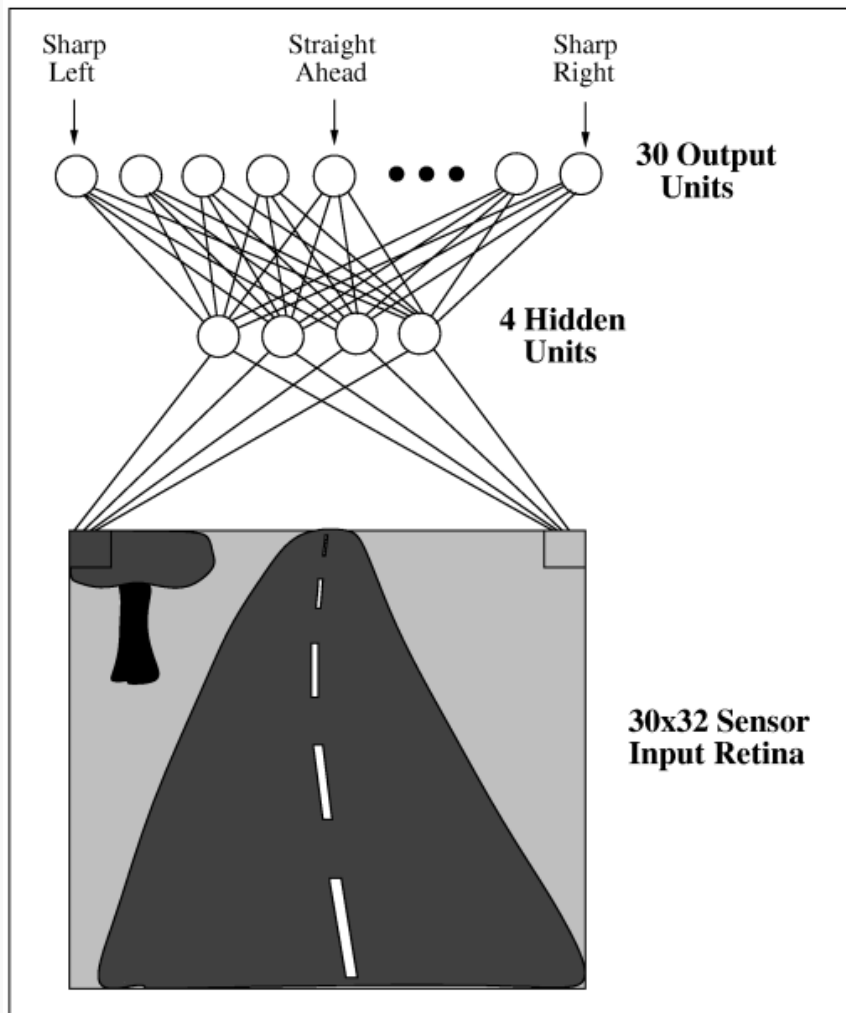


ALVINN

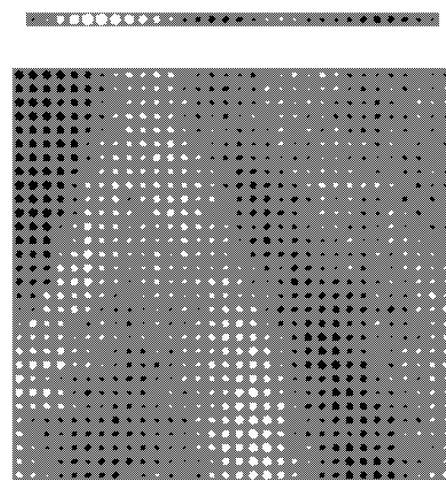


- ❖ Prototype ANN dikembangkan tahun 1993
- ❖ Sistem kemudi otomatis kendaraan dengan kecepatan normal di jalan raya
- ❖ ANN melakukan pembelajaran selama 5 menit untuk mengetahui bagaimana manusia mengendarai kendaraan
- ❖ Alvinn berhasil menjalankan kendaraan dengan kecepatan 70mph sejauh 90 mil di jalan raya (sisi kiri jalan dengan lalu lintas jalan normal)

ALVINN



- ❖ Input data berasal dari kamera dengan sensor sebanyak 960 berukuran 30x32 pixel
- ❖ Data diterima oleh 4 unit hidden layer
- ❖ Output dari hidden layer diterima oleh 30 unit output layer yang menunjukkan seberapa banyak kemudi harus diputar



Bobot dari masing-masing unit



KARAKTERISTIK TASK UNTUK NEURAL NETWORK



PROBLEM YANG BISA DISELESAIKAN DENGAN NEURAL NETWORK

- ❖ Data pelatihan terlalu banyak noisy
- ❖ Data yang kompleks
- ❖ Penggunaan representasi simbolik, misalnya decision tree

KARAKTERSTIK PROBLEM NEURAL NETWORK

Neural network bisa menyelesaikan problem yang mempunyai karakteristik :

- ❖ *Instance* direpresentasikan dalam pasangan attribute-value yang banyak
- ❖ Target output bisa bernilai diskrit, real, atau vector(diskrit/real)
- ❖ Penggunaan representasi simbolik, misalnya decision tree
- ❖ Data training bisa mengandung error
- ❖ Tidak ada batasan waktu pelatihan
- ❖ Dibutuhkannya evaluasi yang cepat terhadap proses pembelajaran
- ❖ Tidak mementingkan kemampuan manusia untuk mengerti proses pembelajaran

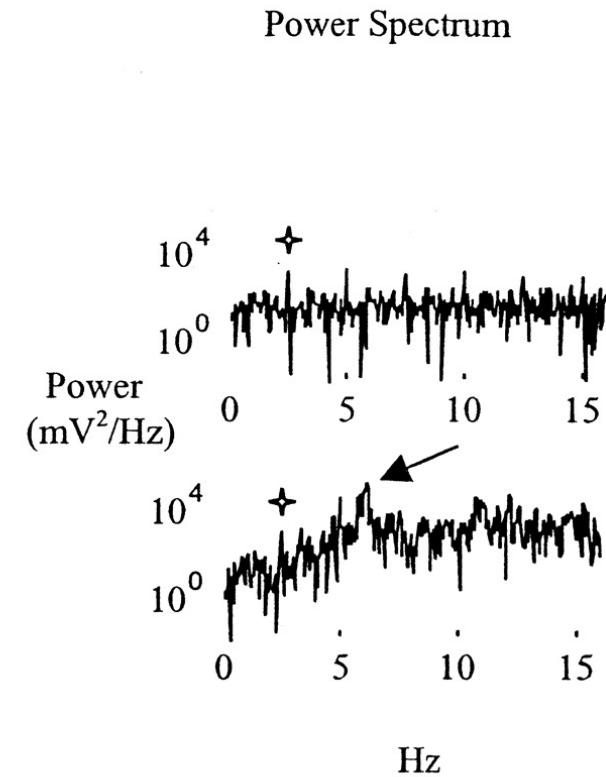
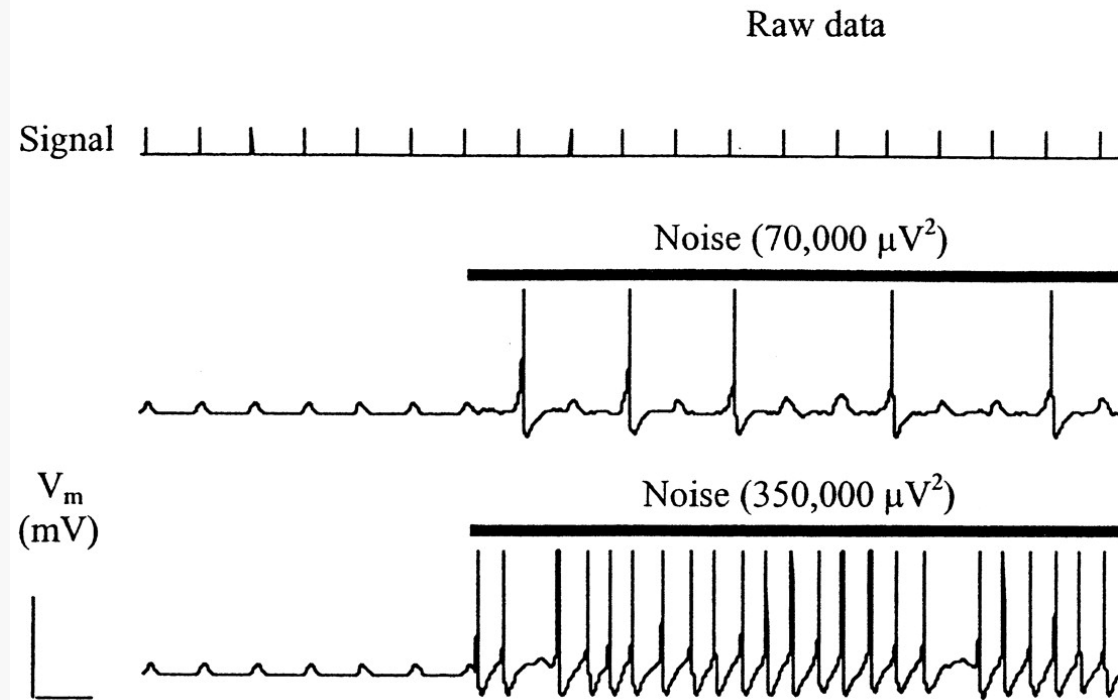
PENGGUNAAN NEURAL NETWORK

- ❖ Pemrosesan Sinyal
- ❖ Sistem kendali
- ❖ Pengenalan Pola
- ❖ Kedokteran
- ❖ Sintesa dan Pengenalan suara
- ❖ Bisnis

PEMROSESAN SINYAL

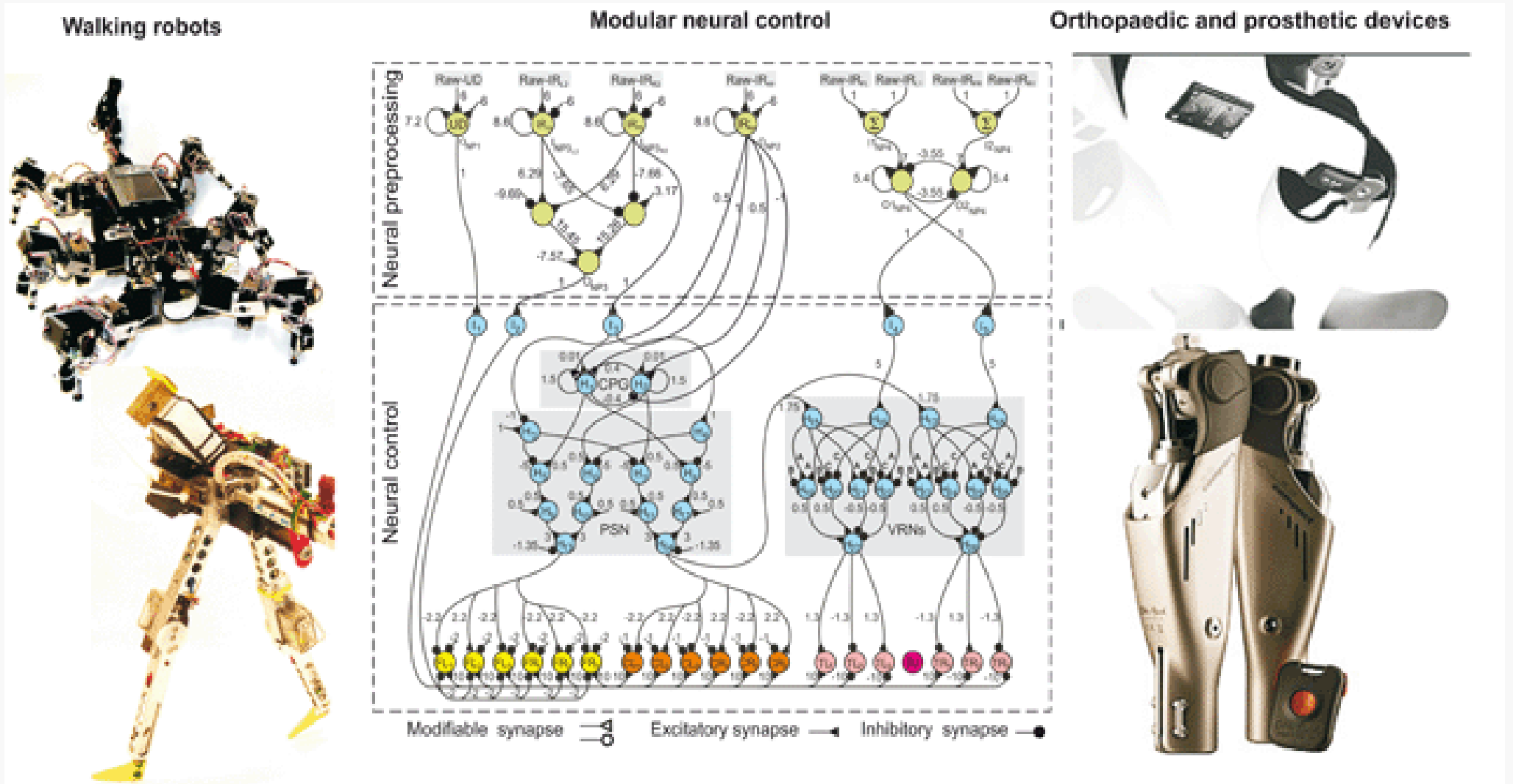
❖ Menghapus gangguan suara

❖ Menghapus gema suara



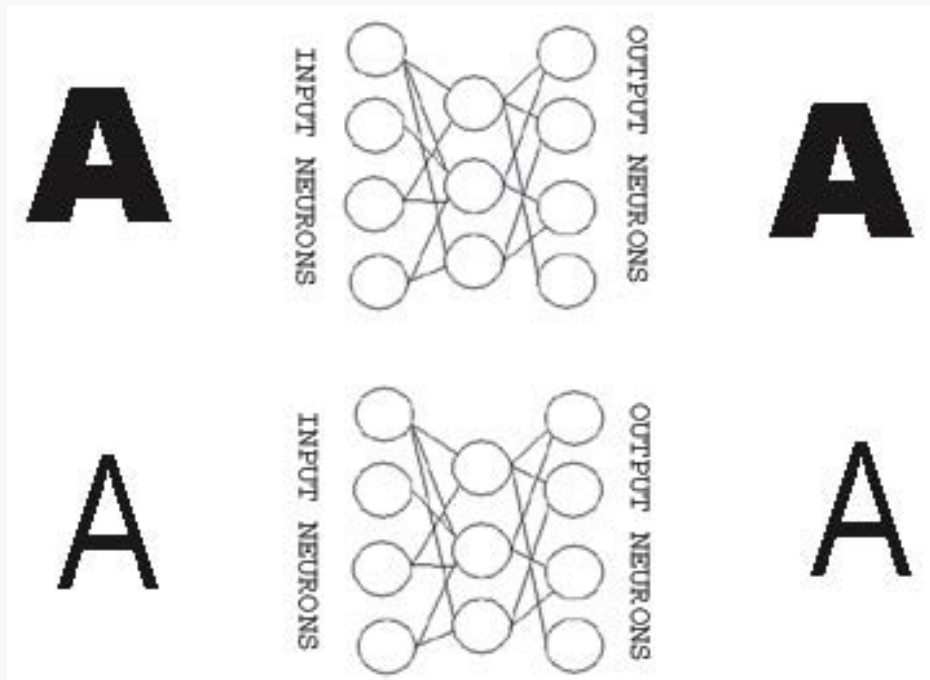
SISTEM KENDALI

Kendali robot

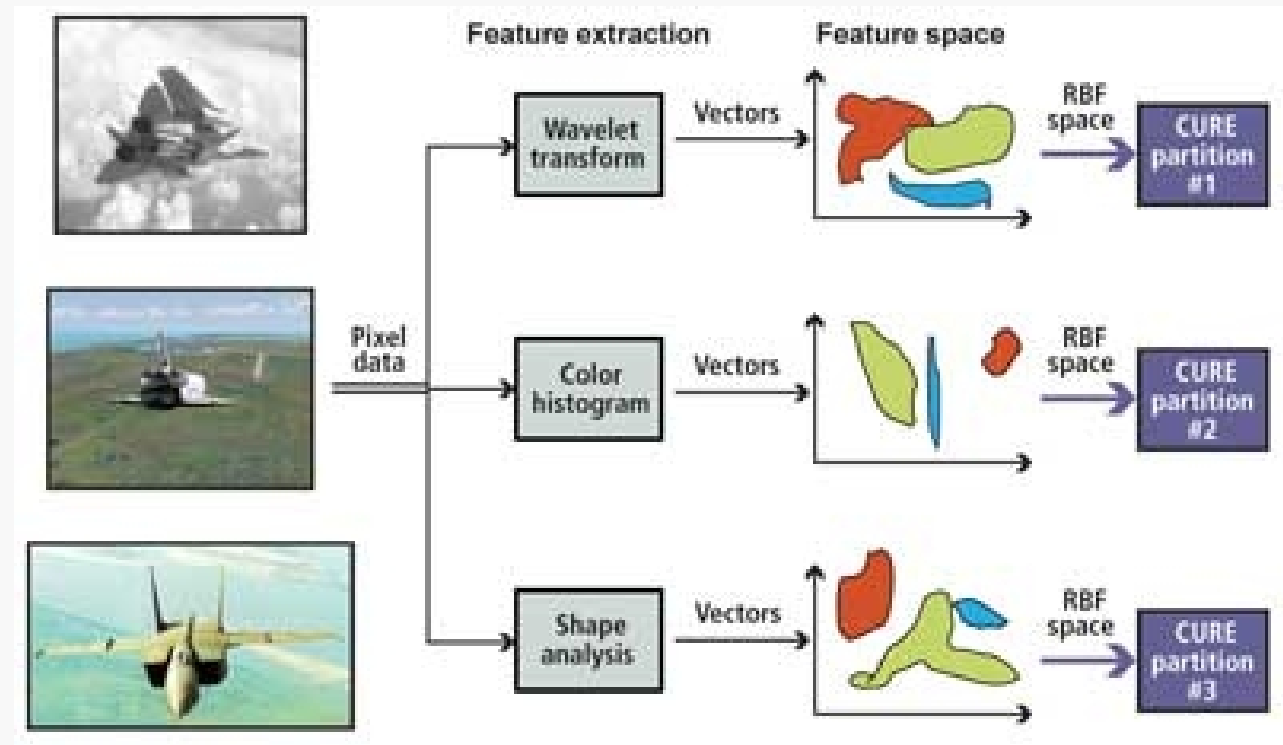


PENGENALAN POLA

Pengenalan Huruf

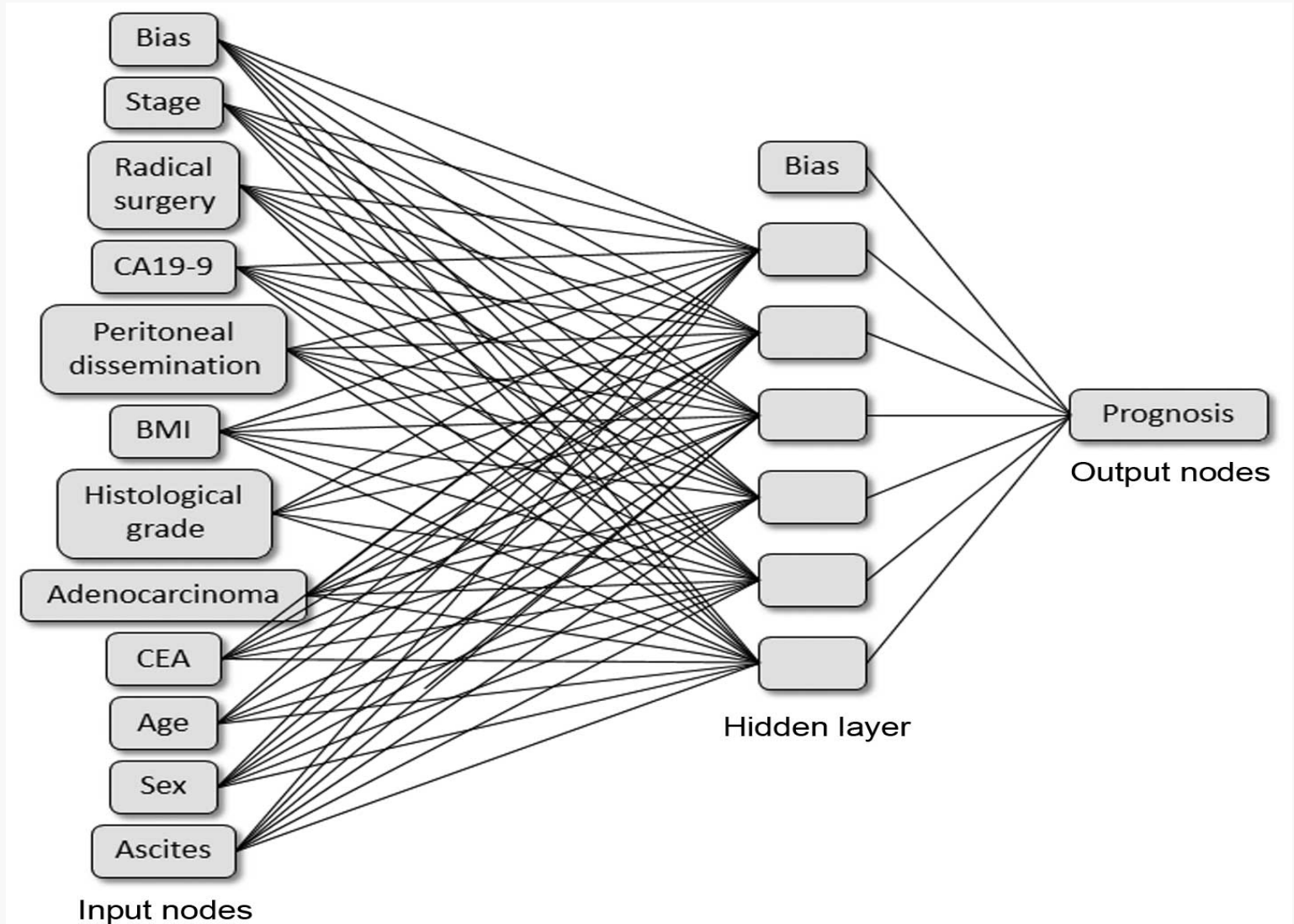


Pengenalan gambar



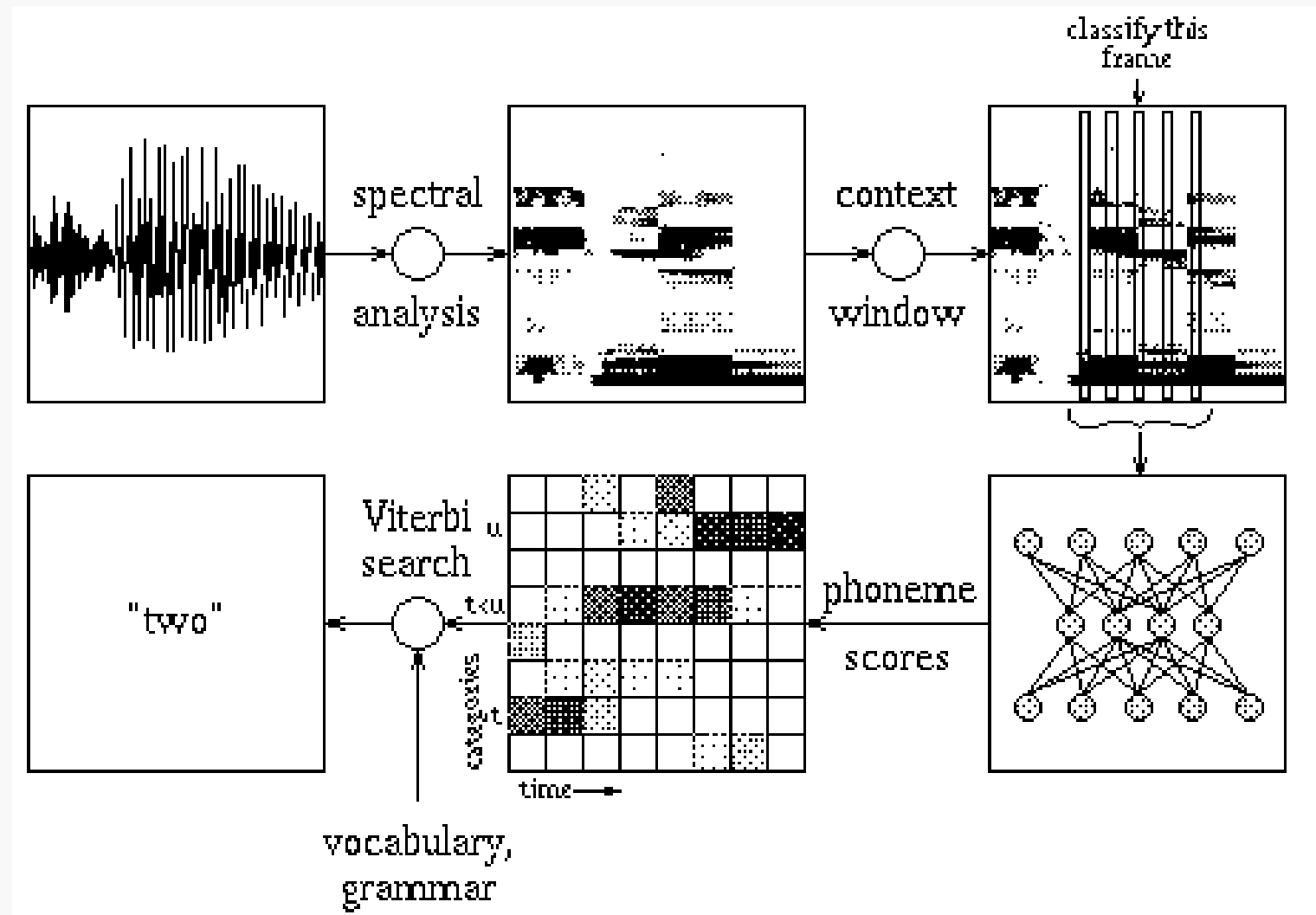
KEDOKTERAN

❖ Diagnosa Penyakit



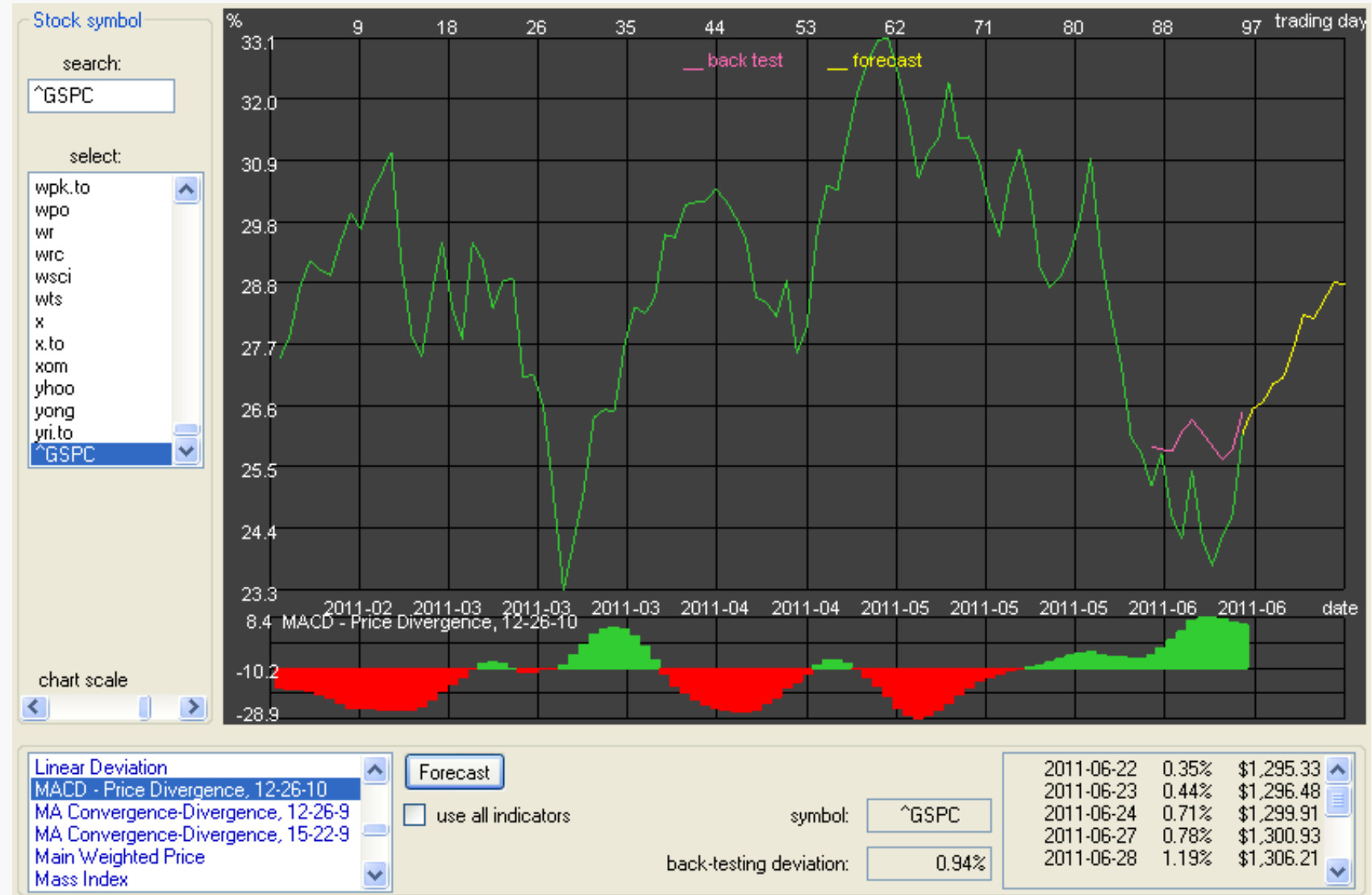
SINTESA DAN PENGENALAN SUARA

- ❖ Sintesa suara
- ❖ Pengenalan suara



BISNIS

- ❖ Prediksi harga saham
- ❖ Prediksi kegagalan bisnis



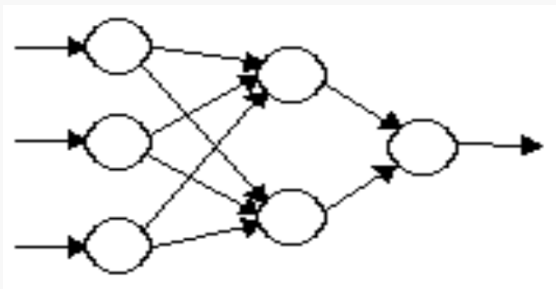


PERCEPTRON

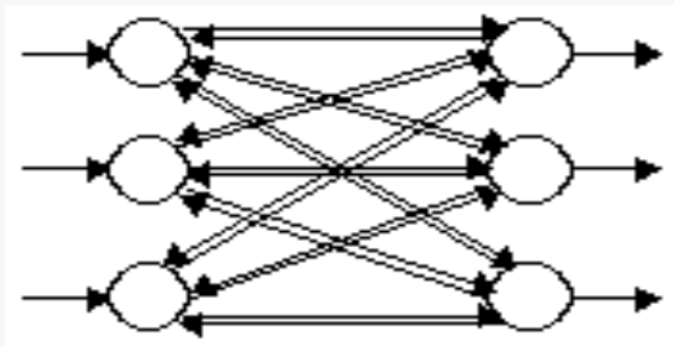


STRUKTUR ARTIFICIAL NEURAL NETWORK

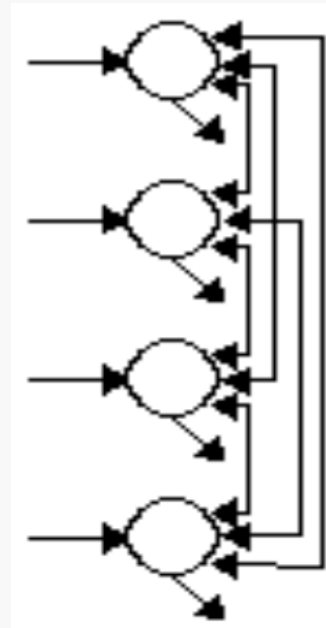
Multilayer feedforward



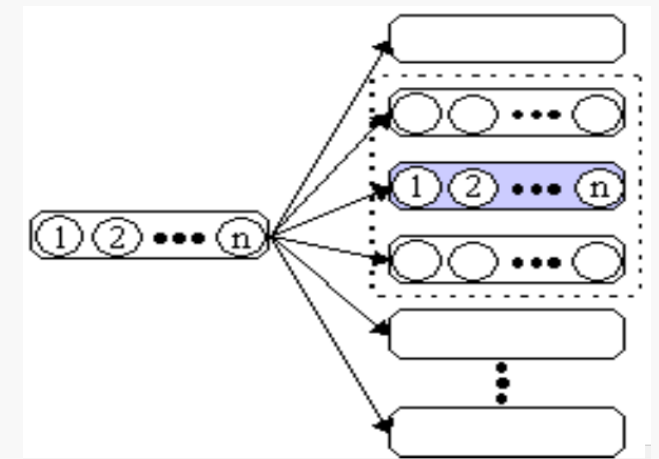
Two layer feedforward



Single layer
terhubung penuh

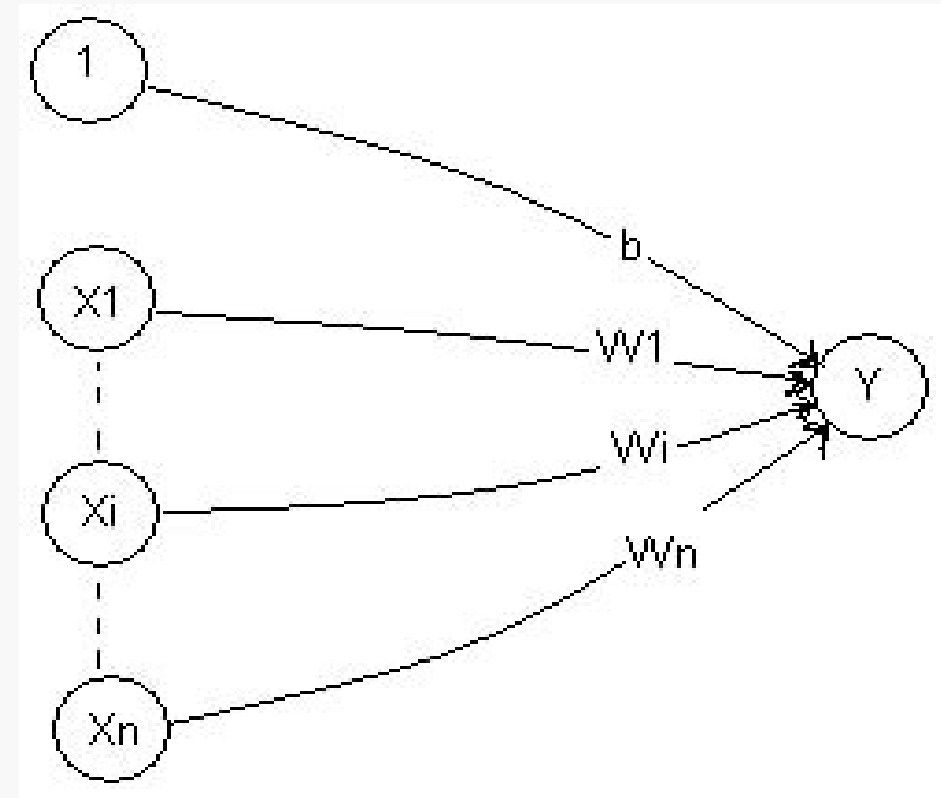


organized
feature map

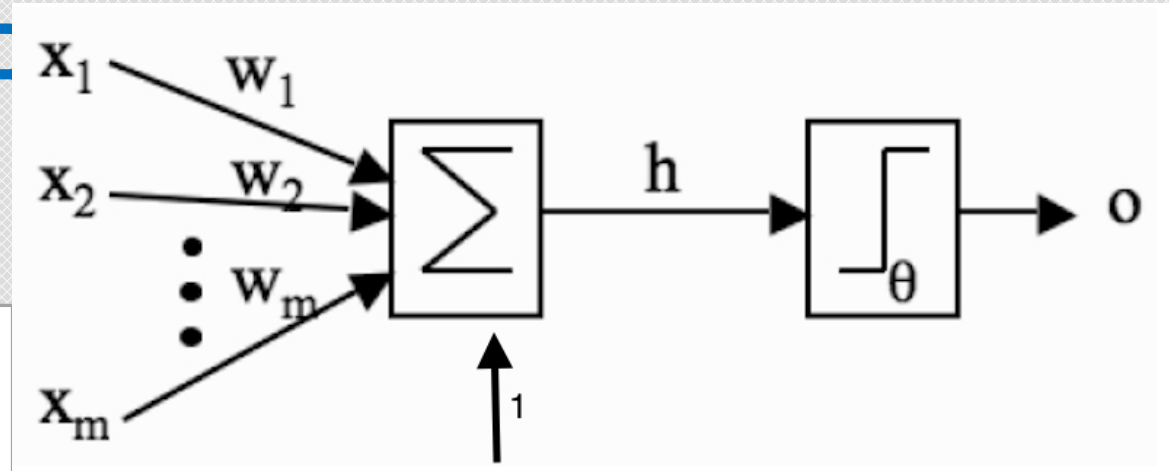
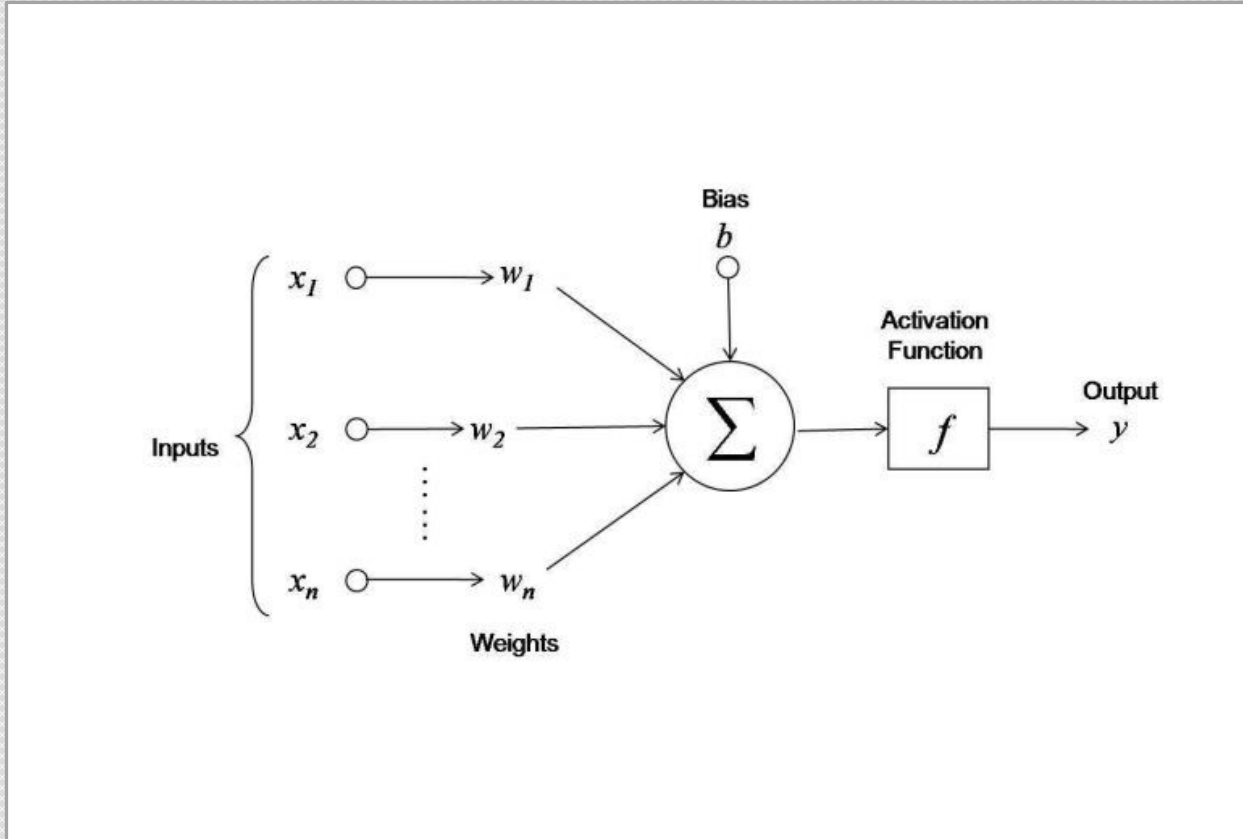


PERCEPTRON

- ❖ Arsitektur sederhana dalam jaringan syaraf tiruan terdiri dari satu layer input unit (yang jumlah neuronnya sesuai dengan banyaknya jumlah komponen dari data yang ingin dikenali) dan satu output unit



MODEL PERCEPTIF



PEMBELAJARAN PERCEPTRON

- ❖ Pembelajaran dengan perceptron memiliki asumsi bahwa prosedur pembelajaran terbukti untuk mengarahkan bobot menjadi konvergen. Perbedaan mendasar dengan Hebb net adalah pada perceptron terdapat error pada input pattern yang dilatihkan, maka bobot akan berubah sesuai dengan formula

$$w_i(\text{new}) = w_i(\text{old}) + \alpha t x_i$$

- ❖ dan jika tidak terdapat error maka nilai bobot tidak akan berubah, atau dengan kata lain bahwa bobot hanya berubah mengikuti kondisi tertentu.

PEMBELAJARAN PERCEPTRON

Algoritma

Langkah 0 Inisialisasi semua bobot dan bias

Untuk kemudahan, set bobot dan bias = 0

Set learning rate α ($0 \leq \alpha \leq 1$) Untuk kemudahan, set $\alpha = 1$

Langkah 1

selama kondisi stop tidak terpenuhi, lakukan langkah 2 – 6

Langkah 2

untuk setiap pasangan vektor input dan target output , s : t,

lakukan langkah 3 – 5

Langkah 3 Set aktivasi input :

$$x_i = s_i ,$$

Langkah 4 Hitung output :

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

PEMBELAJARAN PERCEPTRON

Algoritma

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

Langkah 5 update nilai bobot dan bias jika terjadi error

If $y \neq t$

$$w_i (\text{new}) = w_i (\text{old}) + \alpha t x_i$$

$$b (\text{new}) = b (\text{old}) + \alpha t$$

else

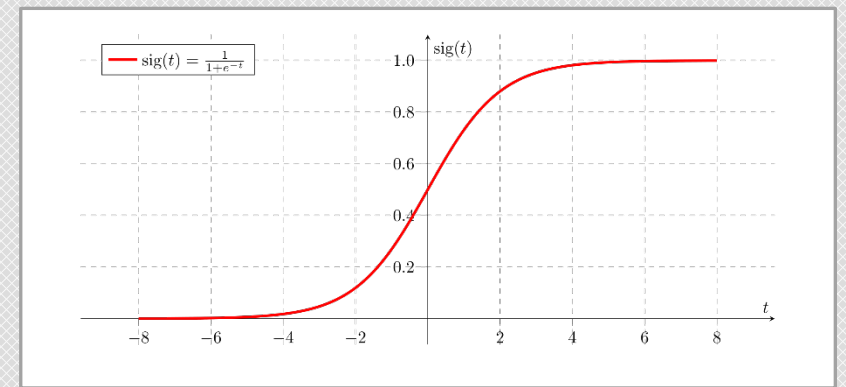
$$w_i (\text{new}) = w_i (\text{old})$$

$$b (\text{new}) = b (\text{old})$$

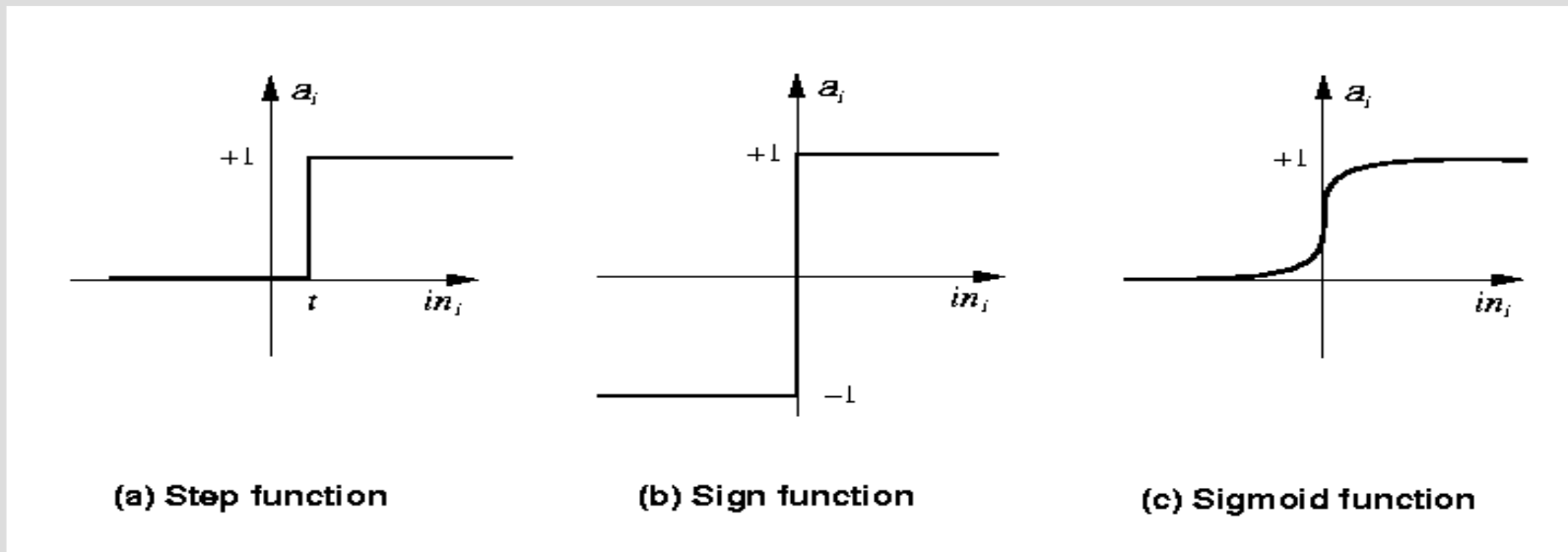
Langkah 6 Tes kondisi stop

Jika tidak ada bobot yang berubah pada step 2, maka stop, selain itu lanjutkan

FUNGSI AKTIVASI



❖ Melakukan transformasi input neuron menjadi output



```
# step function
theta <- 0.5
yin <- 2
if (yin > theta) {
  y <- 1
} else {
  y <- 0
}
```

```
# sign function
theta <- 0.5
yin <- 0.6
if (yin > theta) {
  y <- 1
} else {
  y <- -1
}
```

```
# sigmoid function
yin <- 2
exp_value <- exp(-yin)
y <- 1 / (1 + exp_value)
print(sprintf("yin=%g, y=%g", yin, y))
```

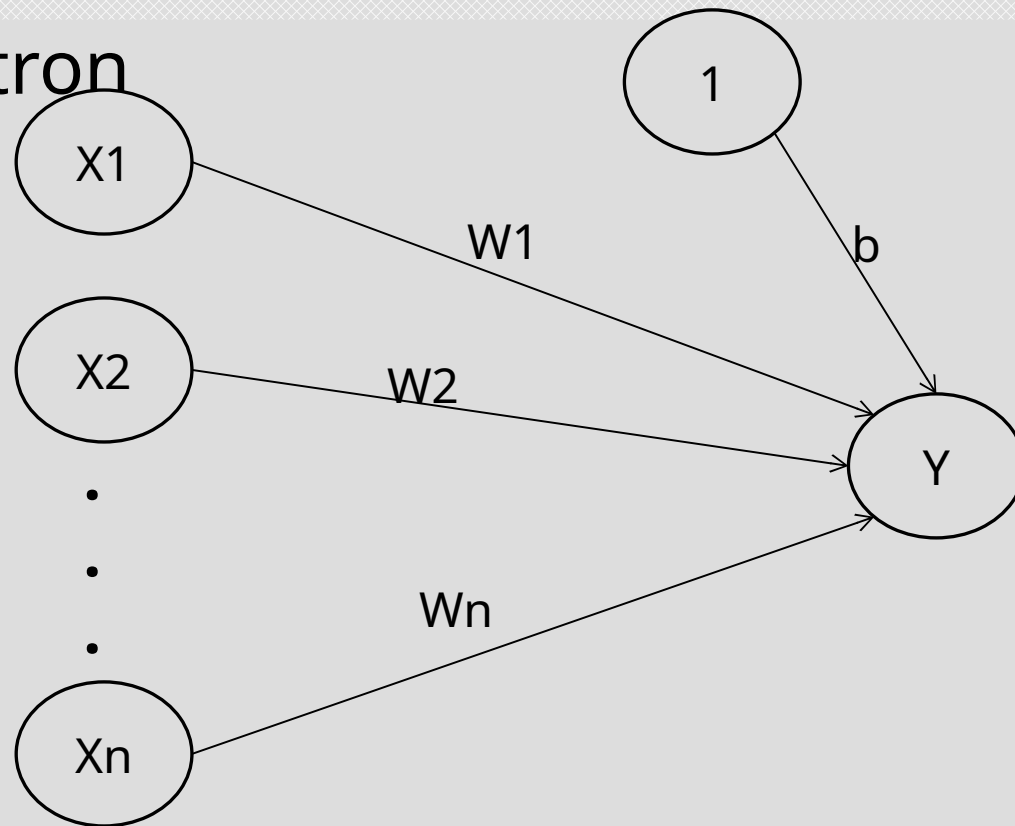


CONTOH PERHITUNGAN



ARSITEKTUR

❖ Simple Perceptron



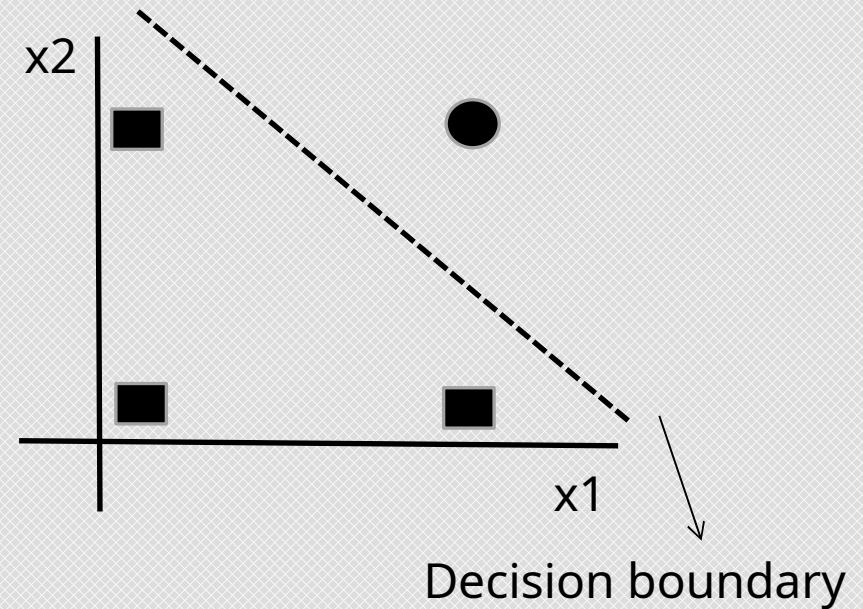
$$\mathbf{WX} + \mathbf{b} = \mathbf{0}$$

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

EXAMPLE 1

❖ Bagaimana memisahkan dua kelompok tsb?

X1	X2	1	Target
1	1	1	1
1	0	1	-1
0	1	1	-1
0	0	1	-1

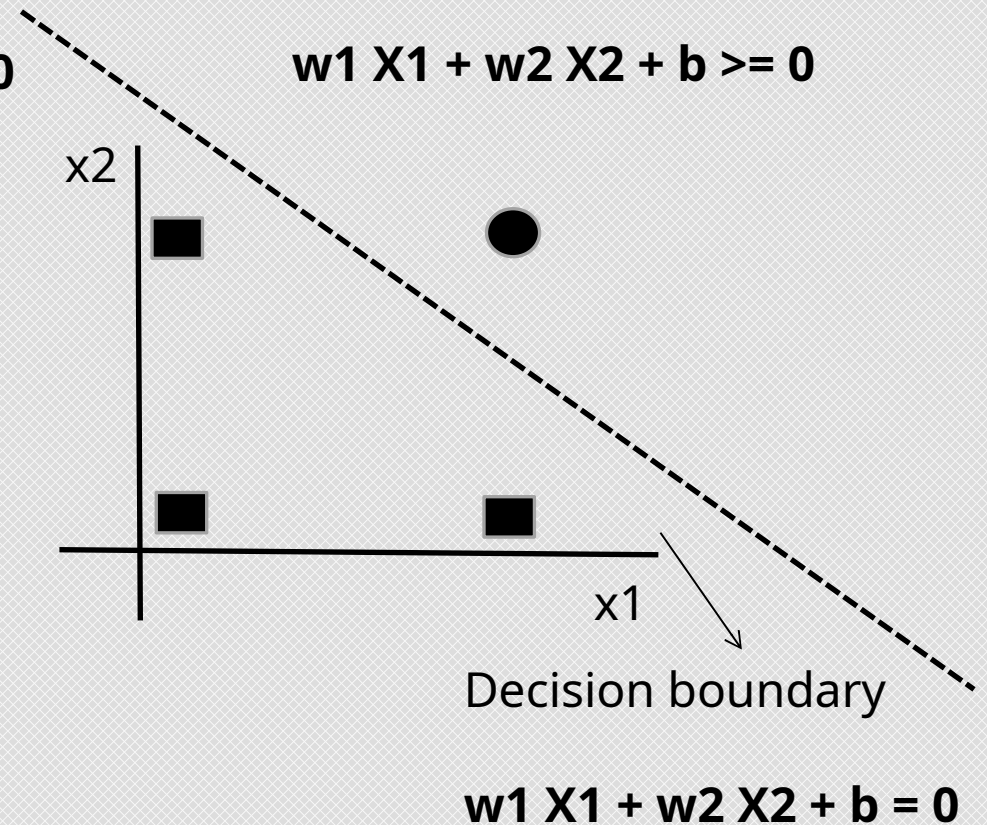


$$WX + b = 0$$

$$w_1 x_1 + w_2 x_2 + b = 0$$

EXAMPLE 1 $w_1 X_1 + w_2 X_2 + b < 0$

- ❖ Bagaimana memisahkan dua kelompok tsb ?
- ❖ Berapa nilai w_1 , w_2 , w_3 agar diperoleh decision boundary yang optimal ?

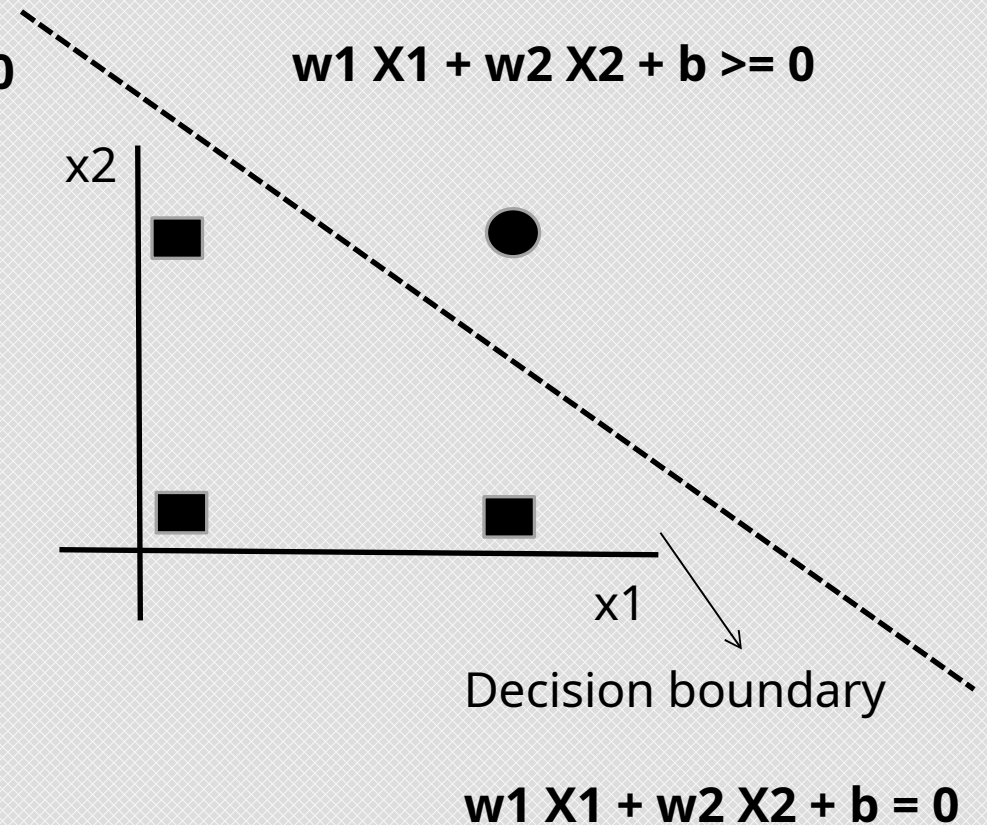


$$W X + b = 0$$

EXAMPLE 1 $w_1 x_1 + w_2 x_2 + b < 0$

❖ Berapa nilai w_1 , w_2 , w_3 agar diperoleh decision boundary yang optimal ?

❖ Boundary decision :
 $f(x_1, x_2)$



$$f(x_1, x_2) = \begin{cases} 1 & \text{if } w_1 x_1 + w_2 x_2 + b \geq 0 \\ -1 & \text{if } w_1 x_1 + w_2 x_2 + b < 0 \end{cases}$$

EXAMPLE 1

$$f(x_1, x_2) = \begin{cases} 1 & \text{if } w_1 X_1 + w_2 X_2 + b \geq 0 \\ -1 & \text{if } w_1 X_1 + w_2 X_2 + b < 0 \end{cases}$$

❖ Berapa nilai w_1 , w_2 , w_3 agar diperoleh decision boundary yang optimal ?

❖ Step 0 : Inisialisasi bobot (w) dan bias (b)

Set learning rate

❖ Step 1 : jalankan step 2-6 selama kondisi false

▪ Step 2 : Training untuk setiap record

▪ Step 3 : set aktivasi input

▪ Step 4 : Hitung respon output

▪ Step 5 : Update bobot dan bias

▪ Step 6 : Test kondisi (ada perubahan nilai bobot)

EXAMPLE

❖ Threshold = 0.8

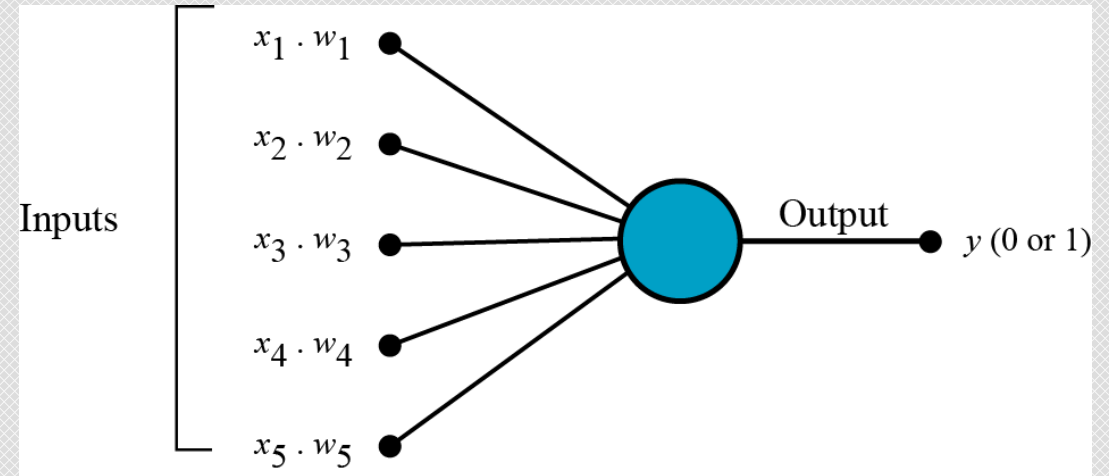
Inputs			Weight	Weighted Sum	Output produced	Output Target	Action
1	1	1	50%	1.5	1	1	None
1	0	1	50%	1	1	-1	Decrease
0	1	1	50%	1	1	-1	Decrease
0	0	1	50%	0.5	-1	-1	None

Inputs			Weight	Weighted Sum	Output produced	Actual Output	Action
1	1	1	50%	1.5	1	1	None
1	0	1	40%	0.8	-1	-1	None
0	1	1	40%	0.8	-1	-1	None
0	0	1	50%	0.5	-1	-1	None

EXAMPLE 2

❖ A perceptron is an artificial neuron similar to a single biological neuron. It takes a set of weighted inputs, sums the inputs and compares the result with a threshold value. If the result is above the threshold value, the perceptron fires, otherwise it does not. When a perceptron fires, the output is 1: when it does not fire, the output is zero.

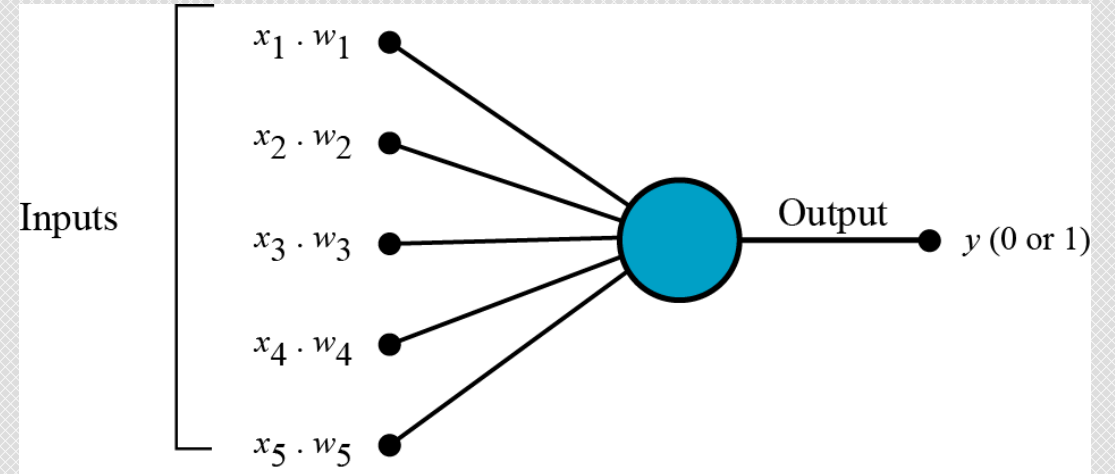
$$W_i X_i + b = 0$$



Sebuah perceptron

EXAMPLE 2

- ❖ Assume a case study with three inputs and one output. There are already four examples with known inputs and outputs
- ❖ This set of inputs is used to train a perceptron with all equal weights ($w_1 = w_2 = w_3$).
- ❖ The threshold is set to 0.8.
- ❖ The original weight for all inputs is 0.5
- ❖ The weights remain the same if the output produced is correct.



Inputs			Output
1	0	0	0
0	0	1	0
1	0	1	0
1	1	1	1

EXAMPLE 2

❖ The weights are increased by 10% if the output produced is less than the output data: the weights are decreased by 10% if the output produced is greater than the output data.

Inputs			Weight	Weighted Sum	Output produced	Actual Output	Action
1	0	0	50%	0.5	0	0	None
0	0	1	50%	0.5	0	0	None
1	0	1	50%	1	1	0	Decrease
1	1	1	50%	1.5	1	1	None

Inputs			Weight	Weighted Sum	Output produced	Actual Output	Action
1	0	0	50%	0.5	0	0	None
0	0	1	50%	0.5	0	0	None
1	0	1	40%	0.8	0	0	None
1	1	1	50%	1.5	1	1	None

ALGORITMA

- ❖ Step 0 : Inisialisasi bobot (w) dan bias (b)
Set learning rate ($0 < \alpha \leq 1$)
- ❖ Step 1 : jalankan step 2-6 selama kondisi false
 - Step 2 : Training untuk setiap record
 - Step 3 : set aktifasi input
 - Step 4 : Hitung respon output
 - Step 5 : Update bobot dan bias
 - Step 6 : Test kondisi (ada perubahan nilai bobot)

FORMULA

❖ Respon Output

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

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

❖ Update bobot dan bias

jika $y \neq t$

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

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

jika $y = t$

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

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

ITERASI

❖ Untuk penyederhanaan, $w=0$, $b=0$, $\alpha=1$, $\theta = 0$

Input			Net	Out	Target	Weight Changes			Weight		
(x1	x2	1)							(W1	w2	b)
									(0	0	0)
(1	1	1)			1						

ITERASI

❖ Epoch 1 baris 2

1 jika Net > 0
 Out = 0 jika Net = 0
 -1 jika Net < 0

$dW1 = \alpha TX1 = 1 * -1 * 1 = -1$
 $dW2 = \alpha TX2 = 1 * -1 * 0 = 0$
 $dW3 = \alpha TX3 = 1 * -1 * 1 = -1$

$W1 = W1 + dW1 = 1 - 1 = 0$
 $W2 = W2 + dW2 = 1 + 0 = 1$
 $W3 = W3 + dW3 = 1 - 1 = 0$

Net = 1 +
 $(x1 * w1 + x2 * w2)$
 $Net = 1 + (1 * 1 + 0 * 1)$
 Net = 2

	Out	Target	Weight Changes	Weight
(x1 x2)			(W1 w2 b)	
(1 0 1)	2	1	(-1 0 -1)	(0 1 0)

ITERASI

❖ Epoch 1 baris 3

1 jika Net > 0
 Out = 0 jika Net = 0
 -1 jika Net < 0

$dW1 = \alpha TX1 = 1 * -1 * 0 = 0$
 $dW2 = \alpha TX2 = 1 * -1 * 1 = -1$
 $dW3 = \alpha TX3 = 1 * -1 * 1 = -1$

$W1 = W1 + dW1 = 0 + 0 = 0$
 $W2 = W2 + dW2 = 1 - 1 = 0$
 $W3 = W3 + dW3 = 0 - 1 = -1$

$Net = b + (x1 * w1 + x2 * w2)$
 $Net = 0 + (0 * 0 + 1 * 1)$
 $Net = 1$

(x1	x2	le	Out	Targ et	Weight	anges	Weight	ct
(0	1	1)	1	1	-1	(0	-1	-1)
(W1	w	2	b)					
(0	1	0)						
(0	0	-1)						

ITERASI

❖ Epoch 1 baris 4

1 jika Net > 0
 0 jika Net = 0
 -1 jika Net < 0

Karena out=target, maka tidak ada perubahan bobot
 $W1 = W1 + dW1 = 0 + 1 = 1$
 $W2 = W2 + dW2 = 0 + 1 = 1$
 $W3 = W3 + dW3 = 0 + 1 = 1$

Net = b + (x1*w1+x2*w2)
 Net = -1 + (0*0+0*0)
 Net = -1

(x1	x2	r	Out	Target	Weight Changes	W1	W2	b)
(0	0	1)	-1	-1	(0 0 0)	(0	0	-1)

ITERASI

❖ Epoch 1

Input			Net	Out	Target	Weight Changes			Weight		
(x1	x2	1)							(W1	w2	b)
(1	1	1)	0	0	1	(1	1	1)	(1	1	1)
(1	0	1)	2	1	-1	(-1	0	-1)	(0	1	0)
(0	1	1)	1	1	-1	(0	-1	-1)	(0	0	-1)
(0	0	1)	-1	-1	-1	(0	0	0)	(0	0	-1)

ITERASI

❖ Epoch 2

Input			Net	Out	Target	Weight Changes			Weight		
(x1	x2	1)							(W1	w2	b)
(1	1	1)	-1	-1	1	(1	1	1)	(1	1	0)
(1	0	1)	1	1	-1	(-1	0	-1)	(0	1	-1)
(0	1	1)	0	0	-1	(0	-1	-1)	(0	0	-2)
(0	0	1)	-2	-1	-1	0	0	0	(0	0	-2)

ITERASI

❖ Epoch 3

Input			Net	Out	Target	Weight Changes			Weight		
x1	x2	1)							(W1	w2	b)
(1	1	1)	-2	-1	1	(1	1	1)	(1	1	-1)
(1	0	1)	0	0	-1	(-1	0	-1)	(0	1	-2)
(0	1	1)	-1	-1	-1	(0	0	0)	(0	1	-2)
(0	0	1)	-2	-1	-1	(0	0	0)	(0	1	-2)

ITERASI

❖ Epoch 4

Input			Net	Out	Target	Weight Changes			Weight		
x1	x2	1)							(W1	w2	b)
(1	1	1)	-1	-1	1	(1	1	1)	(1	2	-1)
(1	0	1)	0	0	-1	(-1	0	-1)	(0	2	-2)
(0	1	1)	0	0	-1	(0	-1	-1)	(0	1	-3)
(0	0	1)	-3	-1	-1	(0	0	0)	(0	1	-3)

ITERASI

❖ Epoch 5

Input			Net	Out	Target	Weight Changes			Weight		
(x1	x2	1)							(W	w2	b)
									1		
(1	1	1)	-2	-1	1	(1	1	1)	(1	2	-2)
(1	0	1)	-1	-1	-1	(0	0	0)	(1	2	-2)
(0	1	1)	0	0	-1	(0	-1	-1)	(1	1	-3)
(0	0	1)	-3	-1	-1	(0	0	0)	(1	1	-3)

ITERASI

❖ Epoch 6

Input			Net	Out	Target	Weight Changes			Weight		
(x1	x2	1)							(W1	w	b)
									2		
(1	1	1)	-1	-1	1	(1	1	1)	(2	2	-2)
(1	0	1)	0	0	-1	(-1	0	-1)	(1	2	-3)
(0	1	1)	-1	-1	-1	(0	0	0)	(1	2	-3)
(0	0	1)	-3	-1	-1	(0	0	0)	(1	2	-3)

ITERASI

❖ Epoch 7

Input			Net	Out	Target	Weight Changes			Weight		
(x1	x2	1)							(W1	w	b)
									2		
(1	1	1)	0	0	1	(1	1	1)	(2	3	-2)
(1	0	1)	0	0	-1	(-1	0	-1)	(1	3	-3)
(0	1	1)	0	0	-1	(0	-1	-1)	(1	2	-4)
(0	0	1)	-4	-1	-1	(0	0	0)	(1	2	-4)

ITERASI

❖ Epoch 8

Input			Net	Out	Target	Weight Changes			Weight		
x1	x2	1)							(W1	w2	b)
(1	1	1)	-1	-1	1	(1	1	1)	(2	3	-3)
(1	0	1)	-1	-1	-1	(0	0	0)	(2	3	-3)
(0	1	1)	0	0	-1	(0	-1	-1)	(2	2	-4)
(0	0	1)	-4	-1	-1	(0	0	0)	(2	2	-4)

ITERASI

❖ Epoch 9

Input			Net	Out	Target	Weight Changes			Weight		
x1	x2	1)							(W1	w	b)
									2		
(1	1	1)	0	0	1	(1	1	1)	3	3	-3
(1	0	1)	0	0	-1	(-1	0	-1)	(2	3	-4)
(0	1	1)	-1	-1	-1	(0	0	0)	(2	3	-4)
(0	0	1)	-4	-1	-1	(0	0	0)	(2	3	-4)

ITERASI

❖ Epoch 10

Input			Net	Out	Target	Weight Changes			Weight		
x1	x2	1)							(W1	w2	b)
(1	1	1)	1	1	1	(0	0	0)	(2	3	-4)
(1	0	1)	-2	-1	-1	(0	0	0)	(2	3	-4)
(0	1	1)	-1	-1	-1	(0	0	0)	(2	3	-4)
(0	0	1)	-4	-1	-1	(0	0	0)	(2	3	-4)

Coming Up:

ARTIFICIAL NEURAL
NETWORK
backpropagation

REFERENCES

- 1. Stuart Russell, Peter Norvig. 2009. Artificial Intelligence: A Modern Approach (3rd Edition)**
- 2. Dan Klein, Pieter Abbeel. 2013. Courseware edX Artificial Intelligence University of California at Berkeley**
- 3. D. Poole & A. Mackworth. 2010. Artificial Intelligence: Foundations of Computational Agents. Cambridge University Press**
- 4. Tom M Michele, 1997, Machine Learning**
- 5. Laurene Fausett, 1993, Fundamental of Neural Network : Architecture, Algorithm and Applications**