

Jaringan Syaraf Tiruan

Kecerdasan Komputasional

Dr. Indah Agustien S.,S.Kom., M.Kom.
Universitas Trunojoyo Madura



1

Jaringan Syaraf Tiruan - Content

1

2

3

4

5

Jaringan Syaraf Tiruan

Neuron

Perceptron

Supervised Learning

Unsupervised Learning

Kecerdasan Komputasional
| Jaringan Syaraf Tiruan

2

Perceptron

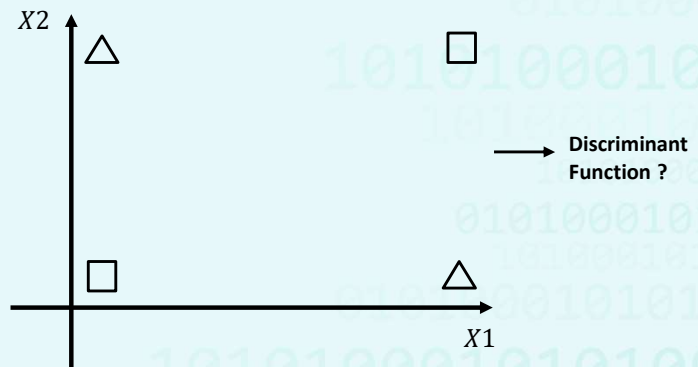
Kegagalan (single layer) Perceptron :

→ Hanya untuk data 'Linearly Separable'

Data Logika XOR :

Tidak Dapat dipisahkan Secara Linear

x1	x2	Target
0	0	0
0	1	1
1	0	1
1	1	0



Kecerdasan Komputasional | Jaringan Syaraf Tiruan

3

Multi Layer Perceptron

$$x_1 \oplus x_2 \Leftrightarrow (x_1 \wedge (\neg x_2)) \vee (x_2 \wedge (\neg x_1))$$

x_1	x_2	$\neg x_1$	$\neg x_2$	$x_1 \wedge \neg x_2$	$x_2 \wedge \neg x_1$	$(x_1 \wedge \neg x_2) \vee (x_2 \wedge \neg x_1)$
0	0	1	1	0	0	0
0	1	1	0	0	1	1
1	0	0	1	1	0	1
1	1	0	0	0	0	0

Kecerdasan Komputasional | Jaringan Syaraf Tiruan

4

Multi Layer Perceptron

Diperlukan 3 buah proses pembelajaran / *learning* :

1 $A = x_1 \wedge (\neg x_2)$

2 $B = x_2 \wedge (\neg x_1)$

3 $A \vee B$

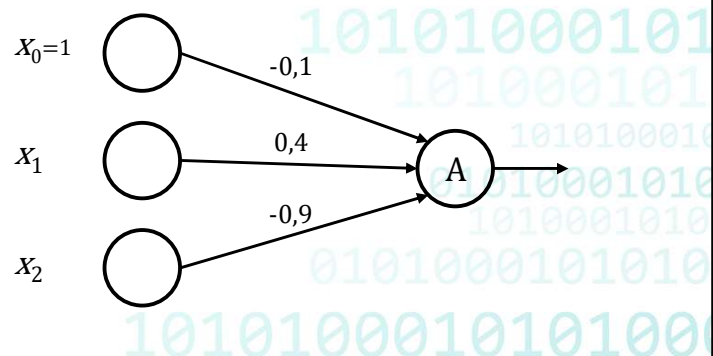
Kecerdasan Komputasional | Jaringan Syaraf Tiruan

5

Multi Layer Perceptron

$$A = x_1 \wedge (\neg x_2)$$

x_1	x_2	$x_1 \wedge (\neg x_2)$
0	0	0
0	1	0
1	0	1
1	1	0



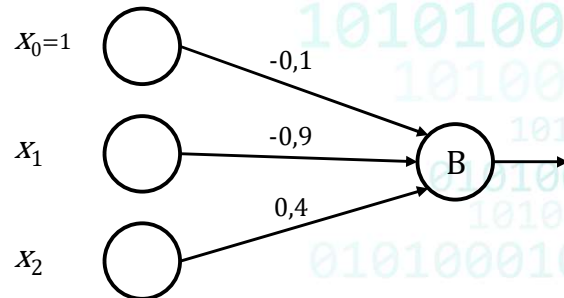
Kecerdasan Komputasional | Jaringan Syaraf Tiruan

6

Multi Layer Perceptron

$$B = x_2 \wedge (\sim x_1)$$

x_1	x_2	$x_2 \wedge (\neg x_1)$
0	0	0
0	1	1
1	0	0
1	1	0



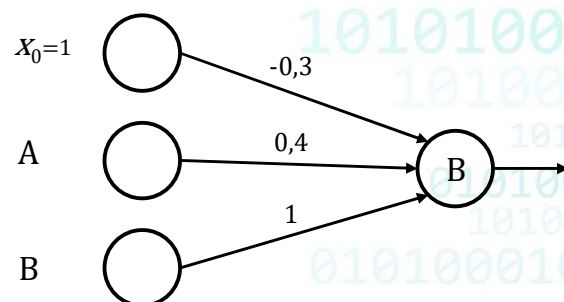
Kecerdasan Komputasional | Jaringan Syaraf Tiruan

7

Multi Layer Perceptron

$$A \vee B$$

x_1	x_2	$A \vee B$
0	0	0
0	1	1
1	0	0
1	1	1



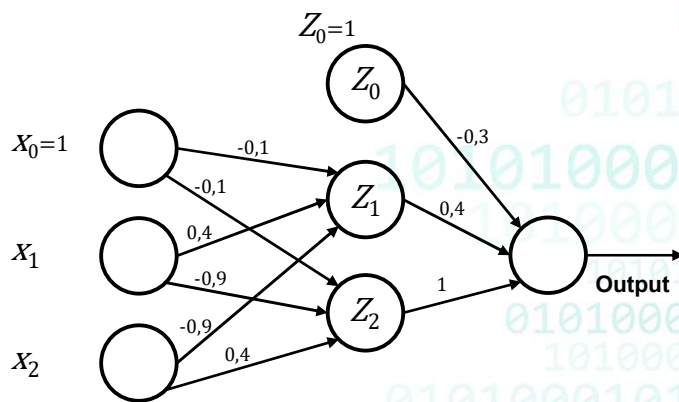
Kecerdasan Komputasional | Jaringan Syaraf Tiruan

8

Multi Layer Perceptron

Model untuk data logika XOR

x1	x2	Target
0	0	0
0	1	1
1	0	1
1	1	0



Kecerdasan Komputasional | Jaringan Syaraf Tiruan

9

Multi Layer Perceptron

Tujuan Pembelajaran untuk membangun model (mencari nilai bobot dan bias)

berdasarkan data pelatihan (*Training data*) Pembelajaran akan dilakukan

berdasarkan :

- Jumlah Epoch tertentu, atau
- Minimal Error yang ditentukan

Jenis Pembelajaran :

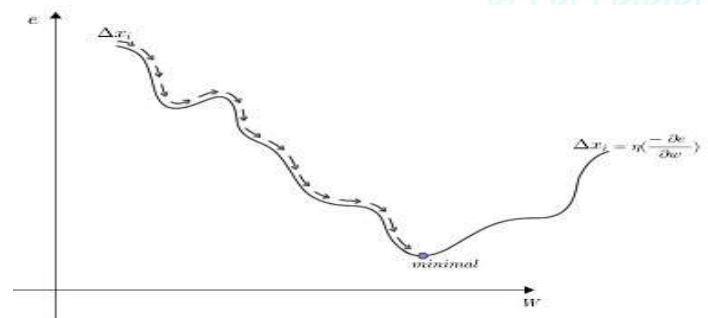
- Supervised Learning / Pembelajaran Terawasi
- Unsupervised Learning / Pembelajaran Tidak Terawasi

Kecerdasan Komputasional | Jaringan Syaraf Tiruan

10

Supervised Learning

- Data Training :
pasangan Data, yaitu sinyal input dan target output
- Salah satu Algoritma Pembelajaran berdasarkan aturan Gradient Descent Learning Rule
- Pencarian nilai bobot dan bias agar didapatkan error minimal



Kecerdasan Komputasional | Jaringan Syaraf Tiruan

11

Supervised Learning

$$E = \sum_{p=1}^n (t_p - y_p)^2$$

t_p : data target untuk data ke- p ;

y_p : output yang dihasilkan JST untuk data ke- p ,

n : jumlah data pelatihan

Kecerdasan Komputasional | Jaringan Syaraf Tiruan

12

Supervised Learning

Pembelajaran Perceptron, update bobot :

$$w_i(t + 1) = w_i(t) + \Delta x_i(t)$$

Aturan Gradient Descent Learning Rule

$$\begin{aligned} \Delta x_i(t) &= \eta \left(-\frac{\partial E}{\partial w_i} \right) \\ \frac{\partial E}{\partial w_i} &= -2(t_p - y_p) \frac{\partial f}{\partial net_p} x_{i,p} \end{aligned}$$

Kecerdasan Komputasional | Jaringan Syaraf Tiruan

13

Supervised Learning

Update Bobot :

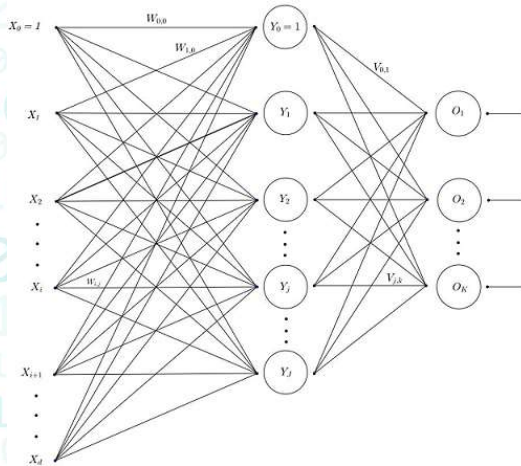
- stochastic/online learning: bobot diupdate pada setiap iterasi pembelajaran pada setiap data pelatihan (bobot diupdate pada tiap iterasi pada tiap epoch)
- batch/offline learning : perubahan bobot pada setiap iterasi diakumulasi, dan diupdate ketika seluruh data pelatihan sudah mengalami proses pembelajaran (bobot diupdate tiap epoch).

Kecerdasan Komputasional | Jaringan Syaraf Tiruan

14

Supervised Learning

Multi Layer Perceptron



Kecerdasan Komputasional | Jaringan Syaraf Tiruan

- neuron pada input layer x_0, x_1, \dots, x_i , dimana jumlah
- neuron pada input layer adalah d
- neuron pada hidden layer y_0, y_1, \dots, y_j , dimana jumlah neuron pada hidden layer adalah j
- neuron pada output layer o_0, o_1, \dots, o_k , dimana jumlah neuron pada hidden layer adalah k
- w_{ij} adalah bobot antara neuron input x_i dengan neuron hidden y_j
- v_{jk} adalah bobot antara neuron input y_j dengan neuron hidden o_k

15

Supervised Learning

FeedForward Pass

Pemrosesan sinyal input menjadi sinyal output berdasarkan bobot yang diinisialisasi sebelumnya

Backward Propagation

Update nilai bobot berdasarkan error antara sinyal output (feedforward pass) dengan target output seharusnya

Kecerdasan Komputasional | Jaringan Syaraf Tiruan

16

Backpropagation – Feedforward Pass

Hitung nilai aktivasi neuron pada masing-masing lapisan (*layer*) :

Y_j adalah nilai aktivasi neuron ke- j pada lapisan tersembunyi O_k adalah nilai aktivasi neuron ke- k pada lapisan output

$$\begin{aligned} y_j &= f_{y_j}(net_{y_j}) \\ &= \frac{1}{1 + e^{-net_{y_j}}} \\ O_k &= f_{O_k}(net_{O_k}) \\ &= \frac{1}{1 + e^{-net_{O_k}}} \\ net_{y_j} &= \sum_{i=0}^d (x_i w_{ij}) \\ net_{O_k} &= \sum_{j=0}^j (y_j v_{jk}) \end{aligned}$$

Kecerdasan Komputasional | Jaringan Syaraf Tiruan

17

Backpropagation – Feedforward Pass

Update Bobot berdasarkan Error :

$$E_p = \frac{1}{2} \left(\frac{\sum_{k=1}^K (t_{k,p} - O_{k,p})^2}{K} \right)$$

K : jumlah neuron pada output layer,
 $t_{k,p}$: target output untuk neuron ke- k pada output layer dari data pelatihan ke- p ,
 $O_{k,p}$: output yang dihasilkan oleh neuron ke- k pada output layer dari data pelatihan ke- p ,

Kecerdasan Komputasional | Jaringan Syaraf Tiruan

18

Backpropagation – Feedforward Pass

Bobot antara hidden dengan output diupdate dengan :

$$v_{jk}(t) = v_{jk}(t-1) + \Delta v_{jk}(t)$$

Backward Propagation akan mencari Δv_{jk} , yaitu :

$$\begin{aligned}\Delta v_{jk} &= \eta \left(\frac{-E}{\partial v_{jk}} \right) \\ &= -\eta \frac{\partial E}{\partial O_k} \frac{\partial O_k}{\partial v_{jk}} \\ &= -\eta \cdot -(t_k - O_{kp}) f'_{O_k} y_j \\ &= \eta \delta O_k y_j\end{aligned}$$

Backpropagation – Feedforward Pass

Bobot antara hidden dengan output diupdate dengan :

$$w_{ij}(t) = w_{ij}(t-1) + \Delta w_{ij}(t)$$

Backward Propagation akan mencari Δw_{ij} , yaitu :

$$\begin{aligned}\Delta w_{ij} &= \eta \left(-\frac{\partial E}{\partial w_{ij}} \right) \\ &= -\eta \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial w_{ij}} \\ &= -\eta - \delta O_k v_{jk} f'_{y_j} x_i \\ &= \eta \delta y_j x_i\end{aligned}$$

