

JARINGAN SYARAF TIRUAN



Nurul Khairina, S.Kom, M.Kom

**UNIVERSITAS MEDAN AREA
MEDAN
2019**

BAB III

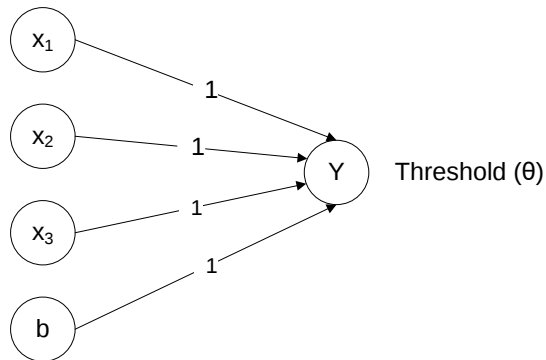
JARINGAN PERCEPTRON

Capaian Pembelajaran Mata Kuliah :

- Mahasiswa mampu memahami arsitektur jaringan Perceptron
- Mahasiswa mampu melakukan pelatihan pengenalan pola dengan jaringan Perceptron

2.1. Konsep Dasar Jaringan Perceptron

Jaringan Perceptron merupakan jaringan yang dapat melakukan perubahan bobot apabila hasil pelatihan (*training*) tidak sama dengan target. Laju pemahaman (*learning rate* (α)) merupakan salah satu variabel yang akan digunakan di dalam proses perbaikan bobot pola agar hasil pelatihan dapat sama dengan target. Jaringan Perceptron termasuk salah satu jaringan yang mampu mengenali pola dengan baik, dan lebih baik dari pada jaringan Hebb yang telah dipelajari pada bab sebelumnya. Berikut arsitektur jaringan Perceptron :



Gambar 1. Arsitektur Jaringan Perceptron

2.2. Pengenalan Pola dengan Jaringan Perceptron

Adapun algoritma pelatihan jaringan Perceptron adalah sebagai berikut :

1. Inisialisasi :

Bobot awal = 0

Bias awal = 0

2. Tentukan *Learning rate* (α), dimana $0 \leq \alpha \leq 1$
3. Hitung nilai $Net = \sum_{i=1}^n x_i \cdot w_i + b$
4. Hitung output y dengan fungsi aktivasi :

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

Catatan : *Threshold* dapat diubah agar $y = t$.

5. Cek apakah output sama dengan target

6. Perbaiki bobot apabila output tidak sama dengan target ($y \neq t$) dengan :

$$w_i (\text{baru}) = w_i (\text{lama}) + \Delta w$$

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

Dimana :

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

$$\Delta b = \alpha \cdot t$$

7. Ulangi tahap 6 apabila output $\neq$ target.

Dalam pelatihan jaringan syaraf tiruan, satu kali iterasi pelatihan disebut dengan *epoch*. Pada jaringan Perceptron, pelatihan bisa terjadi lebih dari satu kali, sehingga hasil pelatihannya nanti akan memiliki beberapa *epoch*.

Contoh Soal :

Lakukan pelatihan pengenalan pola fungsi logika “OR” dengan input dan output biner, dimana bobot awal = 0, bias awal = 0, *learning rate* = 1 dan *threshold* = 0.

Masukan		bias	target
x1	x2		
1	1	1	1
0	1	1	1

1	0	1	1
0	0	1	0

Penyelesaian Contoh Soal :

Dari soal diketahui bahwa *threshold* (θ) = 0, sehingga fungsi aktivasi menjadi :

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

➤ Langkah perhitungan :

❖ Baris ke - 1 :

- $\text{net} = x_1 \cdot w_{\text{awal}} + x_2 \cdot w_{\text{awal}} + b_{\text{awal}}$
 $= 1 \cdot 0 + 1 \cdot 0 + 0 = 0$
- $f(\text{net}) = \text{if } \text{net} = 0, \text{ then } f(\text{net}) = 0.$
- Jika $f(\text{net}) \neq t$, maka lakukan perubahan bobot :

○ Perubahan bobot w_1 :

$$\begin{aligned} \text{➤ } \Delta w_1 &= \alpha \cdot t \cdot x_1 \\ &= 1 \cdot 1 \cdot 1 = 1 \end{aligned}$$

$$\begin{aligned} \text{➤ Bobot } w_1 \text{ baru} &= \text{bobot } w_1 \text{ lama} + \Delta w_1 \\ &= 0 + 1 = 1 \end{aligned}$$

○ Perubahan bobot w_2 :

$$\begin{aligned} \text{➤ } \Delta w_2 &= \alpha \cdot t \cdot x_1 \\ &= 1 \cdot 1 \cdot 1 = 1 \end{aligned}$$

$$\begin{aligned} \text{➤ Bobot } w_1 \text{ baru} &= \text{bobot } w_1 \text{ lama} + \Delta w_1 \\ &= 0 + 1 = 1 \end{aligned}$$

○ Perubahan bobot b :

$$\begin{aligned} \text{➤ } \Delta b &= \alpha \cdot t \\ &= 1 \cdot 1 = 1 \end{aligned}$$

$$\begin{aligned} \text{➤ Bobot } b \text{ baru} &= \text{bobot } b \text{ lama} + \Delta b \\ &= 0 + 1 = 1 \end{aligned}$$

❖ Lanjutkan ke baris ke – 2 dan seterusnya, dan perubahan bobot Δw_i tetap dilakukan dari bobot baru sebelumnya.

❖ Apabila $f(\text{net}) = t$, maka : $\Delta w_1 = 0$; $\Delta w_2 = 0$; $\Delta b = 0$

➤ Hasil pelatihan :

Masukan			Target		y	Perubahan Bobot			Bobot Baru		
x_1	x_2	b	t	net	f(net)	Δw_1	Δw_2	Δb	x_1	x_2	b
Epoch ke - 1											
Inisialisasi									0	0	0
1	1	1	1	0	0	1	1	1	1	1	1
0	1	1	1	2	1	0	0	0	1	1	1
1	0	1	1	2	1	0	0	0	1	1	1
0	0	1	0	1	1	0	0	0	1	1	1
Epoch ke - 2											
1	1	1	1	3	1	0	0	0	1	1	1

0	1	1	1	2	1	0	0	0	1	1	1
1	0	1	1	2	1	0	0	0	1	1	1
0	0	1	0	1	1	0	0	0	1	1	1
Epoch ke - 3											
1	1	1	1	3	1	0	0	0	1	1	1
0	1	1	1	2	1	0	0	0	1	1	1
1	0	1	1	2	1	0	0	0	1	1	1
0	0	1	0	1	1	0	0	0	1	1	1

❖ Pada hasil pelatihan diatas, proses pelatihan sudah mencapai *epoch* ke – 3, namun target masih belum sama dengan output (target $\neq$ output).

Untuk mendapatkan target = output, maka dilakukan uji coba dengan mengubah threshold (θ) = 1, sehingga fungsi aktivasi menjadi :

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

➤ **Hasil pelatihan :**

Masukan			Target		y	Perubahan Bobot			Bobot Baru		
x ₁	x ₂	b	t	net	f(net)	Δw_1	Δw_2	Δb	x ₁	x ₂	b
Epoch ke – 1											
Inisialisasi									0	0	0
1	1	1	1	0	0	1	1	1	1	1	1
0	1	1	1	2	1	0	0	0	1	1	1
1	0	1	1	2	1	0	0	0	1	1	1
0	0	1	0	1	0	0	0	0	1	1	1
Epoch ke – 2											
1	1	1	1	3	1	0	0	0	1	1	1
0	1	1	1	2	1	0	0	0	1	1	1
1	0	1	1	2	1	0	0	0	1	1	1
0	0	1	0	1	0	0	0	0	1	1	1

- ❖ Pada hasil pelatihan diatas, proses pelatihan sudah mencapai *epoch* ke – 2, dan target sudah sama dengan output (target = output), sehingga pelatihan berhenti.
- ❖ Dari hasil pelatihan diperoleh : $w_1 = 1$; $w_2 = 1$; $b = 1$.

Rangkuman :

1. Jaringan Perceptron dapat mengenali pola lebih baik dari jaringan Hebb
2. Pada proses pelatihan, jaringan Perceptron dapat merubah bobot dan bias apabila output $\neq$ target.
3. *Threshold* (θ) pada jaringan Perceptron dapat diubah antara $0 \leq \theta \leq 1$, agar output dapat sesuai dengan target pada *epoch* tertentu.

Tugas :

1. Lakukan pelatihan Perceptron untuk mengenali fungsi logika “AND” dengan input dan output biner. Dimana bobot awal = 0, bias awal = 0, *learning rate* = 1, dan *threshold* = 0 (Apabila belum mendapatkan hasil, lakukan uji coba pelatihan dengan *threshold* yang berbeda).
2. Lakukan pelatihan Perceptron untuk mengenali fungsi logika “XOR” dengan input biner dan output bipolar. Dimana bobot awal = 0, bias awal = 0, *learning rate* = 1, dan *threshold* = 0 (Apabila belum mendapatkan hasil, lakukan uji coba pelatihan dengan *threshold* yang berbeda).