

Perceptron

Dikompilasi Oleh:

Agust Isa Martinus

BACK

Agust Isa Martinus @2008/2022

<aimxx@yahoo.com>

Karakteristik JST Perceptron (Rosenblatt's Perceptron)

- **Aktifasi Bipolar.**
 - Output **+1** (**fires**) atau **-1** (*does not fire*)
- Setiap neuron dihubungkan dengan **jalur berarah dan berbobot** (vektor).
 - Positif: excitatory ($w > 0$)
 - Negatif: inhibitory ($-p, p > 0$)
- Setiap neuron mempunyai **satu threshold** (θ , ambang batas) **yang tetap**.
 - Fires, jika $net \geq \theta$.
 - Threshold diset sedemikian rupa sehingga inhibition adalah absolut. (any nonzero inhibitory input will prevent the neuron from firing.)
- Setiap sinyal membutuhkan **satu satuan waktu** untuk mengalir pada jalur koneksi dari satu neuron ke neuron berikutnya.

Karakteristik JST Perceptron

Perceptron has the following characteristics:

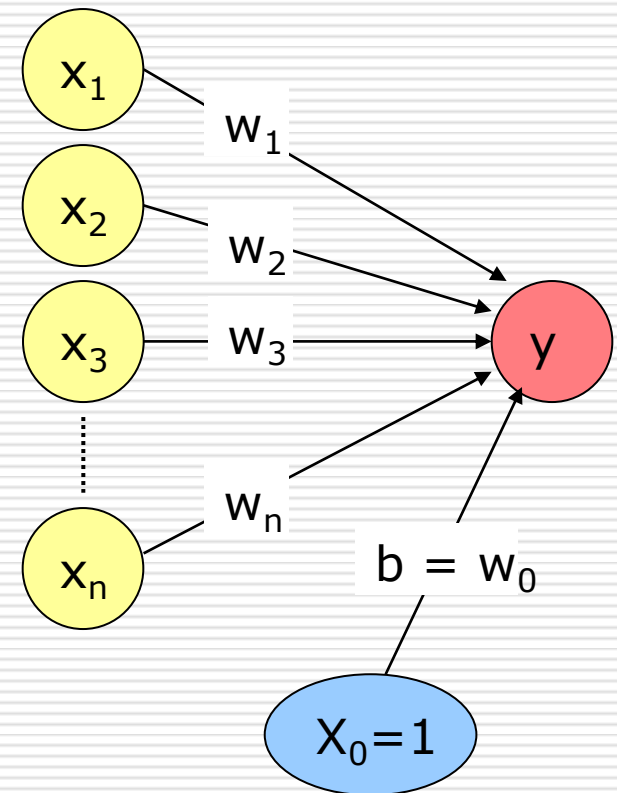
- ❑ Perceptron is an algorithm for Supervised Learning of single layer binary linear classifiers.
- ❑ Optimal weight coefficients are automatically learned.
- ❑ Weights are multiplied with the input features and decision is made if the neuron is fired or not.
- ❑ Activation function applies a step rule to check if the output of the weighting function is greater than zero.
- ❑ Linear decision boundary is drawn enabling the distinction between the two linearly separable classes +1 and -1.
- ❑ If the sum of the input signals exceeds a certain threshold, it outputs a signal; otherwise, there is no output.

<https://www.simplilearn.com/tutorials/deep-learning-tutorial/perceptron#:~:text=Perceptron%20has%20the%20following%20characteristics,neuron%20is%20fired%20or%20not.>

Arsitektur Jaringan Perceptron

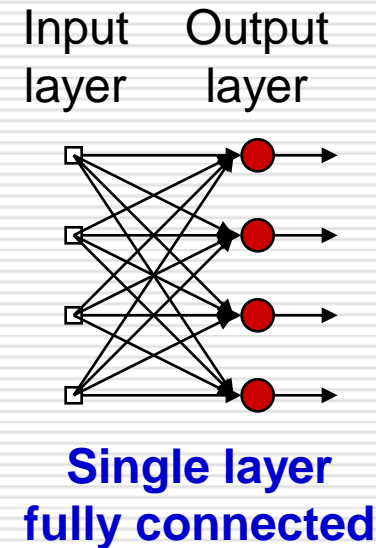
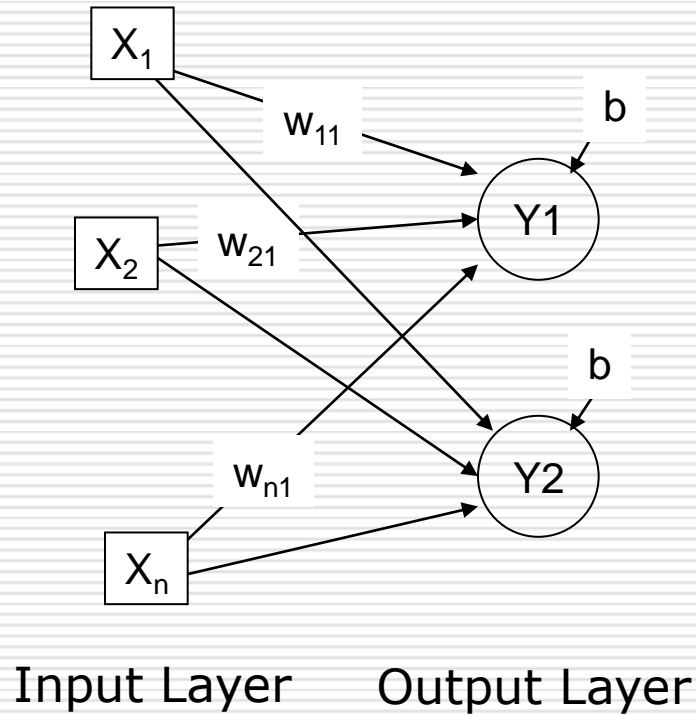
- Jaringan satu layer
 - Beberapa neuron masukan dihubungkan langsung dengan sebuah neuron keluaran
 - Ditambah satu buah bias
- Fungsi aktivasi memiliki nilai -1, 0 dan 1

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



$$net = \sum_{i=1}^n w_i x_i + b$$

Arsitektur Jaringan Perceptron



$$net = \sum_{i=1}^n w_i x_i + b$$

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

Algoritma Pembelajaran Perceptron

The perceptron learning rule is very simple and can be stated as follows:

A. Inisialisasi.

- Start with random weights for the connections;

B. Iterasi.

1. Select an input vector X from the set of training samples;
2. If $y \neq d(X)$ (the perceptron gives an incorrect response), modify all connections w_i according to: $\Delta w_i = d(X) \cdot x_i$;
3. Go back to 2.

Suppose we have a set of learning samples consisting of an input vector X and a desired output $d(X)$. For a classification task the $d(X)$ is usually $+1$ or -1 .

Pelatihan Perceptron

- ❑ Iterasi dilakukan terus menerus hingga seluruh keluaran sama dengan target yang ditentukan
 - Jaringan sudah memahami pola
 - Pada Hebb, iterasi berhenti setelah semua pola dimasukkan
- ❑ Perubahan bobot hanya dilakukan bila keluaran jaringan tidak sama dengan target
 - Yaitu bila $y = f(\text{net}) \neq t$
- ❑ Modifikasi bobot menggunakan laju pemahaman (learning rate, α) yang nilainya dapat diatur
 - Modifikasi bobot tidak hanya ditentukan oleh perkalian antara target dan masukan saja
 - Umumnya, $0 < \alpha < 1$
- ❑ Satu siklus pelatihan yang melibatkan semua pola disebut epoch
 - Pada Hebb, pelatihan hanya dilakukan dalam satu epoch saja

Algoritma Pelatihan Perceptron

□ Langkah 0 : Inisialisasi

Inisialisasi semua bobot dan bias

$$w_i = 0$$

($i=1$ to n)

$$b = 0$$

($b = w_0$)

Set laju belajar (learning rate), α ($0.1 < \alpha \leq 1$)

$$0.1 < \alpha \leq 1$$

($\alpha = 1$, untuk kesederhanaan)

□ Langkah 1 : Iterasi

(*iterasi*)

Laksanakan langkah 2, **hingga semua respons sama dengan targetnya masing-masing**

■ Langkah 2

(*epoch*)

Untuk setiap pasangan masukan dan keluaran, s:t, laksanakan langkah 3 – 5,

□ Langkah 3

Set aktifasi unit masukan

$$x_i = s_i$$

($i=1$ to n)

□ Langkah 4

Hitung respons unit keluaran

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

($i=1$ to n)

$$y =$$

- +1 jika $net > +\theta$
- 0 jika $-\theta \leq net \leq +\theta$
- -1 jika $net < -\theta$

□ Langkah 5

Jika respons unit keluaran tidak sama dengan target pada pola s:t ini, perbaharui bobot dan bias

if $y \neq t$ then

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

($i=1$ to n)

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

else

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

($i=1$ to n)

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

end if

Keunggulan Perceptron

- ❑ Seluruh pola masukan dibandingkan responsnya dengan target yang diinginkan
 - Bobot akan dimodifikasi hanya jika terdapat perbedaan antara keluaran dengan target yang diinginkan
 - Bobot tidak selalu dimodifikasi pada setiap iterasi
- ❑ Kecepatan iterasi ditentukan oleh laju pemahaman (learning rate)
 - Semakin besar α , semakin sedikit iterasi yang diperlukan
 - Tetapi bila α terlalu besar dapat merusak pola yang sudah benar dan mengakibatkan pemahaman menjadi lama
- ❑ Pelatihan dilakukan secara terus menerus hingga jaringan dapat mengerti pola yang ditentukan
 - Teorema konvergensi perceptron menyatakan bahwa apabila ada bobot yang tepat, maka proses pelatihan akan konvergen ke bobot yang tepat tersebut

JST Perceptron

☐ Arsitektur

- Single Layer, Fully Connected, Feed Forward

☐ Metode Pembelajaran

- Perceptron
 - ☐ (target bipolar)
 - ☐ (input binary or bipolar)

☐ Fungsi Aktivasi

- Signum Function
 - ☐ If $net > \theta$ then Output = +1
 - ☐ If $-\theta < net < \theta$ then Output = 0
 - ☐ If $net < -\theta$ then Output = -1

Pelatihan dan Penggunaan

Pelatihan

- Latih JST menggunakan Algoritma Perceptron.
 - Iterasi s.d. semua respons sama dengan targetnya.

Penggunaan

- Operasikan JST menggunakan bobot-bobot (pengetahuan) hasil latihan yang telah dilakukan.

Prosedur Penggunaan Extended Hebb-net

□ Langkah 0

Latih dengan algoritma Perceptron untuk mendapatkan bobot dan bias.

□ Langkah 1

Untuk setiap pasangan masukan dan keluaran, $s:t$, yang akan diklasifikasi/dikenali, laksanakan langkah 2 – 3,

■ Langkah 2

Set aktivasi unit masukan

$$x_i = s_i \quad (i=1 \text{ to } n)$$

■ Langkah 3

Hitung respons unit keluaran, y

$$\blacksquare \quad \text{net} = b + \sum x_i w_i \quad (i=1 \text{ to } n)$$

$y =$

- $+1$ jika $\text{net} > \theta$
- 0 jika $-\theta \leq \text{net} \leq +\theta$
- -1 jika $\text{net} < -\theta$

Contoh Kasus 1 : Fungsi Logika

□ Kasus :

- Buat perceptron yang dapat menyatakan fungsi logika AND

□ Gunakan representasi masukan/keluaran :

- A. Masukan dan keluaran bipolar (-1 atau 1)
- B. Masukan biner (0 atau 1) dan keluaran bipolar (-1 atau 1)

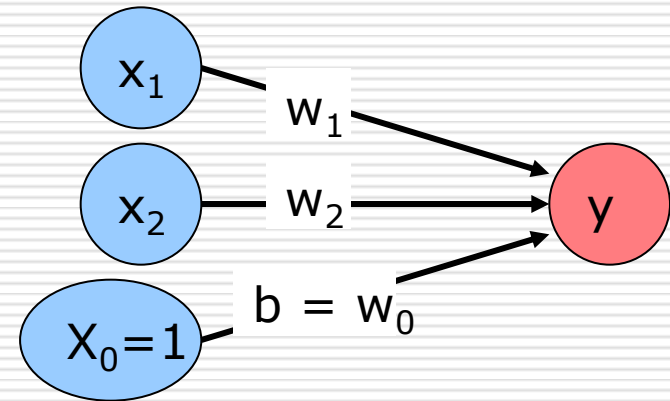
□ Inisialisasi :

- Bobot dan bias awal $w_i = 0$, $b = 0$
- Learning rate $\alpha = 1$ (penyederhanaan)
- Threshold $\theta = 0$

A. Representasi Bipolar

Tabel masukan - target fungsi logika AND Arsitektur jaringan Perceptron :

	Masukan			Target
	s_1	s_2	s_0	t
Pola 1	1	1	1	1
Pola 2	1	-1	1	-1
Pola 3	-1	1	1	-1
Pola 4	-1	-1	1	-1



Masukan bipolar dan target bipolar

Fungsi aktivasi untuk $\theta = 0$:

Parameter yang digunakan :

$$\alpha = 1$$

$$\theta = 0$$

$$w_i = w_i + \alpha \cdot t \cdot x_i$$

$$net = \sum_{i=0}^2 x_i w_i$$

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

Epoch Pertama (Bipolar)

Masukan			Target	Keluaran		Perubahan Bobot			Bobot Baru		
x_1	x_2	bias	t	net	f(net)	Δw_1	Δw_2	Δb	w_1	w_2	b
									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

$$net = \sum_{i=1}^2 x_i w_i + b$$

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

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

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

- ☐ Epoch pertama terdiri dari empat iterasi
- ☐ Pada iterasi pertama-ketiga, keluaran $y = f(net)$ tidak sama dengan target \rightarrow bobot diubah.
- ☐ Pada iterasi keempat, nilai $f(net)$ sama dengan target ($f(net) = -1$) \rightarrow bobot tidak diubah.
- ☐ Pada epoch pertama, belum seluruh $f(net)$ sama dengan target \rightarrow iterasi dilanjutkan pada epoch kedua

Epoch Kedua (Bipolar)

Masukan			Target	Keluaran		Perubahan Bobot			Bobot Baru		
x_1	x_2	bias	t	net	f(net)	Δw_1	Δw_2	Δb	w_1	w_2	b
									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

$$net = \sum_{i=1}^2 x_i w_i + b$$

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

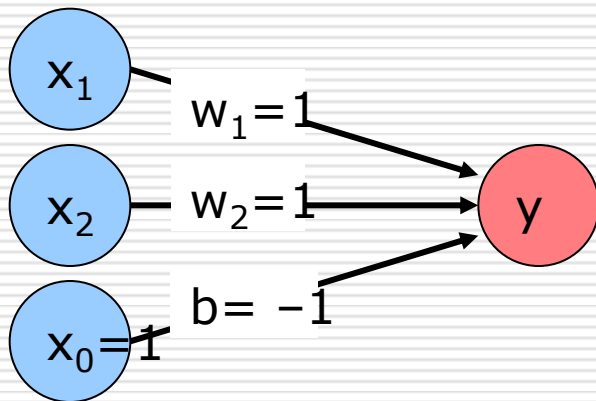
$$w_i(\text{baru}) = w_i(\text{lama}) + \alpha \cdot t \cdot x_i$$

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

- ☐ Bobot awal diperoleh dari epoch pertama
- ☐ Pada setiap iterasi dalam epoch kedua, semua pola f(net) sama dengan target t → tidak dilakukan perubahan bobot lagi
- ☐ Jaringan sudah mengenal pola, iterasi dihentikan

Arsitektur Perceptron Diperoleh

Arsitektur jaringan Perceptron :



Tabel masukan - keluaran fungsi logika AND

Masukan			Keluaran
x_1	x_2	x_0	y
1	1	1	1
1	-1	1	-1
-1	1	1	-1
-1	-1	1	-1

Masukan bipolar dan target bipolar

Fungsi aktivasi untuk $\theta = 0$:

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

Epoch Pertama (Bipolar)

Masukan			Target	Keluaran		Perubahan Bobot			Bobot Baru		
x_1	x_2	bias	t	net	f(net)	Δw_1	Δw_2	Δb	w_1	w_2	b
									0	0	0
-1	-1	1	-1	0	0	1	1	-1	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

$$net = b + \sum_{i=1}^2 x_i w_i$$

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

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

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

- ☐ Epoch pertama terdiri dari empat iterasi
- ☐ Pada iterasi pertama-ketiga, keluaran $y = f(net)$ tidak sama dengan target \rightarrow bobot diubah.
- ☐ Pada iterasi keempat, nilai $f(net)$ sama dengan target ($f(net) = -1$) \rightarrow bobot tidak diubah.
- ☐ Pada epoch pertama, belum seluruh $f(net)$ sama dengan target \rightarrow iterasi dilanjutkan pada epoch kedua

Epoch Kedua (Bipolar)

Masukan			Target	Keluaran		Perubahan Bobot			Bobot Baru		
x_1	x_2	bias	t	net	f(net)	Δw_1	Δw_2	Δb	w_1	w_2	b
									1	1	-1
-1	-1	1	-1	-3	-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	1	1	0	0	0	1	1	-1

$$net = \sum_{i=1}^2 x_i w_i + b$$

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

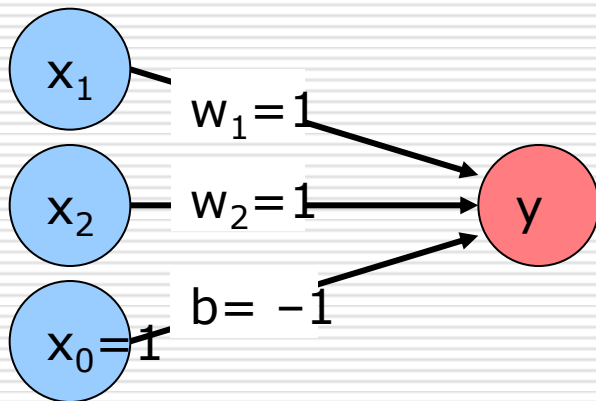
$$w_i(\text{baru}) = w_i(\text{lama}) + \alpha \cdot t \cdot x_i$$

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

- ☐ Bobot awal diperoleh dari epoch pertama
- ☐ Pada setiap iterasi dalam epoch kedua, semua pola f(net) sama dengan target t → tidak dilakukan perubahan bobot lagi
- ☐ Jaringan sudah mengenal pola, iterasi dihentikan

Arsitektur Perceptron Diperoleh

Arsitektur jaringan Perceptron :



Tabel masukan - keluaran fungsi logika AND

Masukan			Keluaran
x_1	x_2	bias	y
1	1	1	1
1	-1	1	-1
-1	1	1	-1
-1	-1	1	-1

Masukan bipolar dan target bipolar

Fungsi aktivasi untuk $\theta = 0.3$:

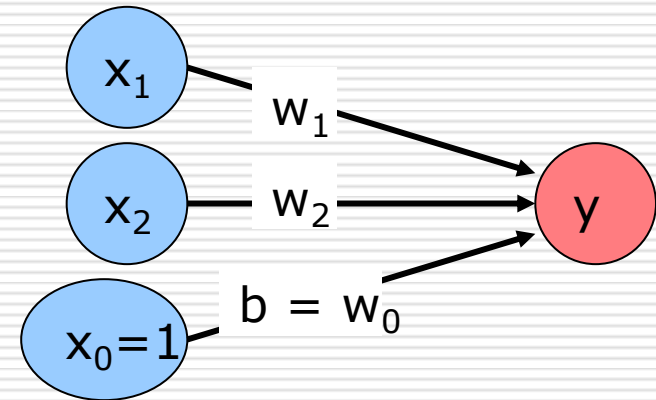
$$y = f(net) = \begin{cases} 1, & net > 0.3 \\ 0, & -0.3 \leq net \leq 0.3 \\ -1, & net < -0.3 \end{cases}$$

B. Representasi Biner-Bipolar

Tabel masukan - target fungsi logika AND :

Masukan			Target
s_1	s_2	s_0 (Tetap)	t
1	1	1	1
1	0	1	-1
0	1	1	-1
0	0	1	-1

Arsitektur jaringan Perceptron :



Masukan biner dan target bipolar

Parameter yang digunakan :

$$\alpha = 1$$
$$\theta = 0,2$$

$$w_i = w_i + \alpha \cdot t \cdot x_i$$

$$net = \sum_{i=0}^2 x_i w_i$$

Fungsi aktivasi untuk $\theta = 0,2$:

$$y = f(net) = \begin{cases} 1, & net > 0,2 \\ 0, & -0,2 \leq net \leq 0,2 \\ -1, & net < -0,2 \end{cases}$$

Epoch Pertama (Biner-Bipolar)

Masukan			Target	Keluaran		Perubahan Bobot			Bobot Baru		
x_1	x_2	x_0	t	net	f(net)	Δw_1	Δw_2	Δb	w_1	w_2	b
									0	0	0
1	1	1	1	0	0	1	1	1	1	1	1
1	0	1	-1	2	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	-1

$$net = \sum_{i=1}^2 x_i w_i + b$$

$$y = f(net) = \begin{cases} 1, & net > 0,2 \\ 0, & -0,2 \leq net \leq 0,2 \\ -1, & net < -0,2 \end{cases}$$

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

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

- Hanya pola masukan terakhir saja yang $f(net) = \text{target}$, (masih ada yang berbeda).

- Iterasi harus dilanjutkan ke epoch berikutnya

Parameter yang digunakan :

$$\alpha = 1$$

$$\theta = 0,2$$

$$y = f(net) = \begin{cases} 1, & net > 0,2 \\ 0, & -0,2 \leq net \leq 0,2 \\ -1, & net < -0,2 \end{cases}$$

Epoch Kedua (Biner-Bipolar)

Masukan			Target	Keluaran		Perubahan Bobot			Bobot Baru		
x_1	x_2	bias	t	net	f(net)	Δw_1	Δw_2	Δb	w_1	w_2	b
									0	0	-1
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	-1	0	0	0	-1	-1	0	0	-2
0	0	1	-1	-2	-1				0	0	-2

$$net = \sum_{i=1}^2 x_i w_i + b$$

$$y = f(net) = \begin{cases} 1, & net > 0,2 \\ 0, & -0,2 \leq net \leq 0,2 \\ -1, & net < -0,2 \end{cases}$$

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

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

- ☐ Bobot awal diperoleh dari epoch pertama
- ☐ Pada epoch kedua ini, hanya pola masukan terakhir saja dimana $f(net) = target$
- ☐ Iterasi masih harus dilanjutkan ke epoch berikutnya (ketiga)

Epoch Ke-10 (Akhir)

Masukan			Target	Keluaran		Perubahan Bobot			Bobot Baru		
x_1	x_2	bias	t	net	f(net)	Δw_1	Δw_2	Δb	w_1	w_2	b
									2	3	-4
1	1	1	1	1	1	0	0	0	2	3	-4
1	0	1	-1	1	-1	0	0	0	2	3	-4
0	1	1	-1	-1	-1	0	0	0	2	3	-4
0	0	1	-1	-4	-1	0	0	0	2	3	-4

$$net = \sum_{i=1}^2 x_i w_i + b$$

$$y = f(net) = \begin{cases} 1, & net > 0,2 \\ 0, & -0,2 \leq net \leq 0,2 \\ -1, & net < -0,2 \end{cases}$$

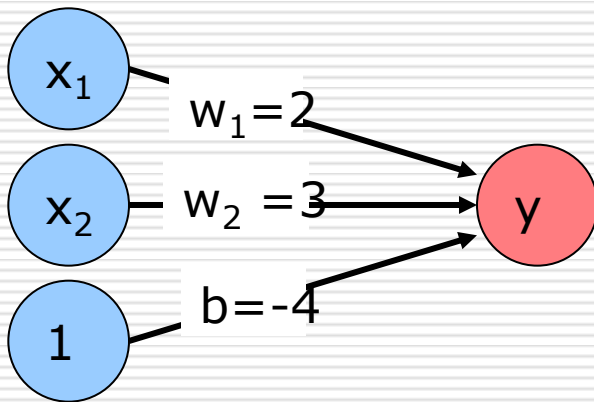
$$w_i(\text{baru}) = w_i(\text{lama}) + \alpha \cdot t \cdot x_i$$

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

- ☐ Bobot awal diperoleh dari epoch sebelumnya
- ☐ Pada setiap iterasi dalam epoch ke-10, semua pola f(net) sudah sama dengan target t
- ☐ Jaringan sudah mengenal pola, iterasi dihentikan

Arsitektur Perceptron Diperoleh

Arsitektur jaringan Perceptron :



Masukan biner dan target bipolar

Tabel masukan dan keluaran fungsi logika

Masukan			Keluaran
x_1	x_2	bias	y
1	1	1	1
1	0	1	-1
0	1	1	-1
0	0	1	-1

Fungsi aktivasi untuk $\theta = 0,2$:

$$y = f(net) = \begin{cases} 1, & net > 0,2 \\ 0, & -0,2 \leq net \leq 0,2 \\ -1, & net < 0,2 \end{cases}$$

Pengenalan Pola Karakter

- ❑ Konsep pengenalan pola karakter menggunakan perceptron :
 - Masukan berbentuk pola yang menyerupai huruf alfabet
 - Perceptron hendak dilatih untuk mengenal pola tersebut
- ❑ Algoritma pengenalan karakter :
 - Nyatakan setiap pola masukan sebagai vektor bipolar yang elemennya adalah tiap titik dalam pola tersebut.
 - Berikan nilai target = 1 jika pola masukan menyerupai huruf yang diinginkan. Jika tidak, beri nilai target = -1.
 - Tentukan inisialisasi bobot, bias, learning rate dan threshold
 - Lakukan proses pelatihan perceptron

Contoh Kasus 2 : Pola Karakter

Pengenalan sebuah pola karakter

- ❑ Diketahui 6 buah pola masukan seperti pada slide berikut
- ❑ Buat model perceptron untuk mengenali pola menyerupai huruf "A"

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

Pola 1

Pola 2

Pola 3

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

Pola 4

Pola 5

Pola 6

Representasi Kasus

- Setiap karakter pola dianggap sebagai sebuah unit masukan
 - Karakter “#” diberi nilai 1, karakter “.” diberi nilai -1
 - Pembacaan pola dilakukan dari kiri ke kanan, dimulai dari baris paling atas

1	.	.	#	#	.	.	.
2	.	.	.	#	.	.	.
3	.	.	.	#	.	.	.
4	.	.	#	.	#	.	.
5	.	.	#	.	#	.	.
6	.	#	#	#	#	#	.
7	.	#	.	.	.	#	.
8	.	#	.	.	.	#	.
9	#	#	#	.	#	#	#

→

-1	-1	1	1	-1	-1	-1
-1	-1	-1	1	-1	-1	-1
-1	-1	-1	1	-1	-1	-1
-1	-1	1	-1	1	-1	-1
-1	-1	1	-1	1	-1	-1
-1	1	1	1	1	1	-1
-1	1	-1	-1	-1	1	-1
-1	1	-1	-1	-1	1	-1
1	1	1	-1	1	1	1

- Setiap pola terdiri dari 9 baris dan 7 kolom
 - Perceptron terdiri dari 63 unit masukan (x1 s/d x63) dan sebuah bias bernilai = 1

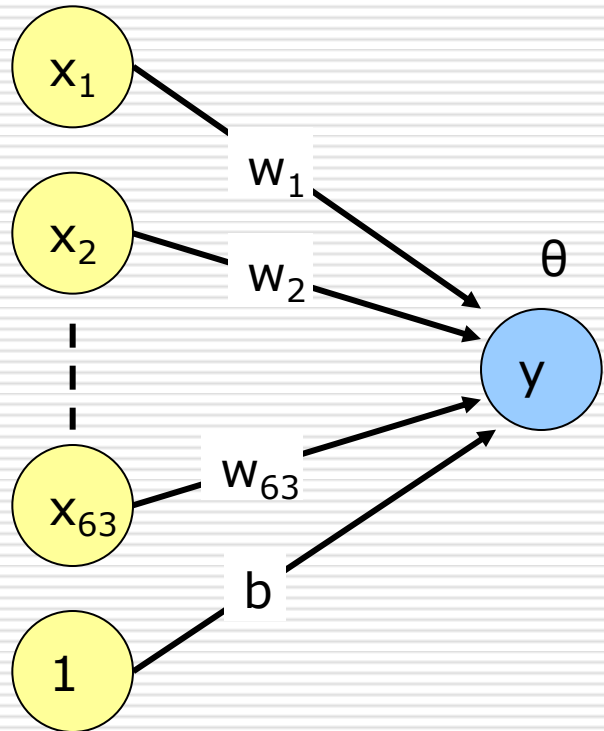
Representasi Kasus

□ Target

- Keluaran jaringan bernilai 1 jika diberi masukan menyerupai huruf "A" dan bernilai -1 jika tidak menyerupai huruf "A"
- Pola yang menyerupai huruf "A" adalah pola 1 dan 4.

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

Arsitektur Perceptron



Perceptron memiliki 63 unit masukan,
Sebuah bias dan sebuah unit keluaran

✚ Asumsi parameter :

- Bobot awal = 0
- Learning rate $\alpha = 1$
- Threshold $\theta = 0,5$

✚ Pelatihan dilakukan dengan memasukkan seluruh pola huruf

✚ Hitung

$$net = \sum_{i=1}^{63} x_i w_i + b$$

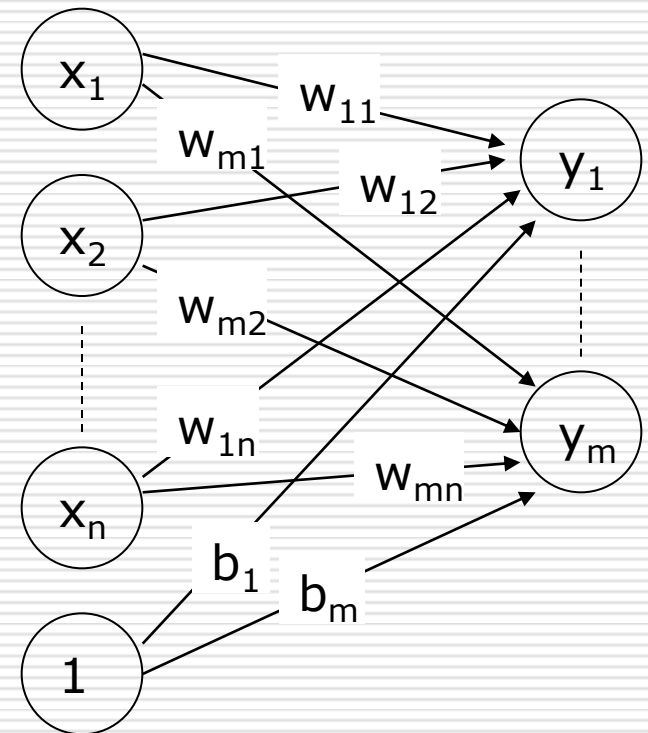
$$y = f(net) = \begin{cases} 1, & net > 0,5 \\ 0, & -0,5 \leq net \leq 0,5 \\ -1, & net < 0,5 \end{cases}$$

✚ Bila $f(net) \neq \text{target}$, bobot dan bias diubah

✚ Proses pelatihan terus dilakukan hingga semua keluaran sama dengan target.

Pengenalan Beberapa Karakter

- Pengenalan beberapa pola sekaligus dilakukan dengan menggabungkan beberapa perceptron
 - Terdapat beberapa unit keluaran
- Setiap unit masukan dan bias dihubungkan dengan setiap target
 - Bobot koneksi dari unit x_i ke y_j diberi label w_{ji}
 - Bobot bias diberi label b_1, b_2, \dots, b_m



Algoritma Pelatihan

- Algoritma pelatihan perceptron untuk pengenalan beberapa pola :
 - 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, dan berikan nilai $t_j = -1$ jika sebaliknya ($j=1, 2, \dots, m$)
 - Inisialisasi semua bobot, bias dan learning rate
 - Lakukan proses pelatihan perceptron seperti dibahas sebelumnya
 - Untuk setiap unit keluaran, hitung respon unit keluaran net dan $y_j = f(\text{net}_j)$
 - Perbaiki bobot pola bila respon keluaran tidak sama dengan target ($y_j \neq t_j$) menurut persamaan
$$w_{ji}(\text{baru}) = w_{ji}(\text{lama}) + \alpha \cdot t_j \cdot x_i$$
$$b_j(\text{baru}) = b_j(\text{lama}) + \alpha \cdot t_j$$
 - Lakukan proses pelatihan hingga $y_j = t_j$ ($j=1, 2, \dots, m$)

Kasus 3 : Pengenalan Pola "A B C"

- Bila diketahui 6 buah pola masukan seperti pada kasus 2, buat model perceptron untuk mengenali pola menyerupai huruf "A", "B" atau "C".

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

Pola 1

Pola 2

Pola 3

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

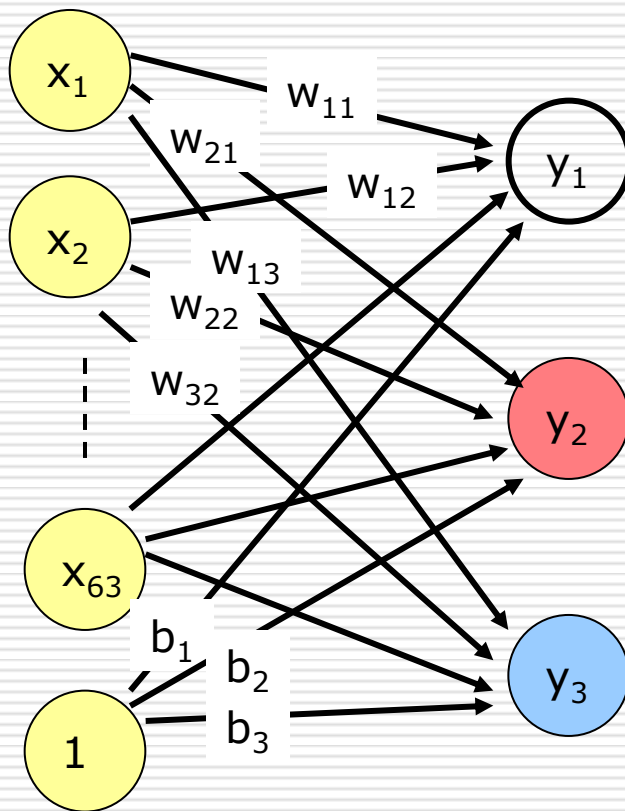
Pola 4

Pola 5

Pola 6

Arsitektur Perceptron

Arsitektur jaringan perceptron pengenalan pola menyerupai huruf A, B dan C dengan 6 pola masukan



Tabel pasangan masukan - target

Pola Masukan	Target		
	t_1	t_2	t_3
Pola 1	1	-1	-1
Pola 2	-1	1	-1
Pola 3	-1	-1	1
Pola 4	1	-1	-1
Pola 5	-1	1	-1
Pola 6	-1	-1	1

Tabel Pelatihan

Masukan		Target	Keluaran		Perubahan Bobot		Bobot Baru	
x_1 - x_{63}	bias	t_1 - t_3	net_1 - net_3	$f(net_1)$ - $f(net_3)$	Δw_{11} - $\Delta w_{1.63}$, Δw_{21} - $\Delta w_{2.63}$, Δw_{31} - $\Delta w_{3.63}$	Δb_1 - Δb_3	w_{11} - $w_{1.63}$, w_{21} - $w_{2.63}$, w_{31} - $w_{3.63}$	b_1 - b_3
							00.....00	0..0
...

Terdiri dari :

- ☐ 63 unit masukan dan sebuah bias b ($x_1, x_2, \dots x_{63}$ dan b)
- ☐ 3 kolom target (t_1, t_2 dan t_3)
- ☐ 3 kolom net (net_1, net_2 dan net_3)
- ☐ 3 kolom fungsi aktivasi ($y_1=f(net_1), y_2=f(net_2), y_3=f(net_3)$)
- ☐ 3*63 kolom perubahan bobot
 - ☒ $\Delta w_{11}, \Delta w_{12}, \dots, \Delta w_{1.63}, \Delta w_{21}, \Delta w_{22}, \dots \Delta w_{2.63}, \Delta w_{31}, \Delta w_{32}, \dots \Delta w_{3.63}$
- ☐ 3*63 kolom bobot
 - ☒ $w_{11}, w_{12}, \dots, w_{1.63}, w_{21}, w_{22}, \dots w_{2.63}, w_{31}, w_{32}, \dots w_{3.63}$

Ringkasan

- ❑ Perceptron merupakan jaringan satu layer yang menyerupai arsitektur jaringan Hebb.
- ❑ Perbedaan perceptron dengan jaringan Hebb :
 - Fungsi aktivasi memiliki nilai -1, 0 dan 1
 - Iterasi dilakukan terus menerus hingga seluruh keluaran sama dengan target yang ditentukan
 - Perubahan bobot hanya dilakukan bila keluaran jaringan tidak sama dengan target
 - Terdapat parameter laju pemahaman yang menentukan kecepatan iterasi
- ❑ Perceptron memiliki keunggulan dibandingkan dengan jaringan Hebb karena adanya beberapa hal :
 - Pelatihan dilakukan terus hingga jaringan dapat mengerti pola yang ditentukan.
 - Bobot tidak selalu dimodifikasi pada setiap iterasi. Jika terdapat perbedaan, baru bobot tersebut dimodifikasi
 - Modifikasi bobot menggunakan laju pemahaman yang dapat diatur, tidak hanya ditentukan oleh perkalian antara target dan masukan saja

Tugas

1. Buat JST Perceptron pengenalan pola untuk mengenali Pola Karakter **Angka Terakhir NIM** anda.
 - Pola untuk pembelajaran sebanyak 12 Pola Karakter:
 - Ukuran karakter 7x5 pixel,
 - Pola karakter (satu angka terakhir NIM) yang betul untuk latihan, buat 3-pola, target output +1,
 - Pola karakter yang salah untuk latihan, sebanyak pola angka sisanya (9-angka selain angka terakhir NIM anda) masing-masing 1-pola, target -1,
 - Representasi input-output: Bipolar-Bipolar,
 - Inisialisasi:
 - Fungsi aktivasi adalah Signum Function (+1, 0, -1) dengan nilai threshold, $\theta = 0.5$,
 - Laju Pembelajaran (Learning Rate), $\alpha = 1$,
 - Bias dan semua bobot diset masing-masing menjadi 0 (NOL),
 - Algoritma pembelajaran yang digunakan adalah Perceptron,
 - Tugas:
 - Buat dan tunjukkan tabel pelatihannya (buat menggunakan Excell),
 - Buat Makalah berisi pembahasan tugas ini (mulai dari deskripsi masalah, perancangan pelatihan, pengujian, ... s.d. Kesimpulan)
 - Kumpulkan (a) File Makalah (.pdf) dan (b) File Excell hasil kerja anda.
2. Gunakan JST Perceptron Pengenal Pola yang terbentuk tadi untuk menilai karakter 0,1,2,...,9.
 - Bentuk Pola Karakter tidak harus persis sama dengan yang untuk pelatihan, bisa dicoba diubah-ubah/dibuat baru,
 - Buat dan tunjukkan tabel penggunaannya.

Pustaka Acuan

- Jong Jek Siang, Drs. M.Sc., *Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan Matlab*, ANDI, Jogjakarta, 2005.
- Laurene Fausett, *Fundamentals Of Neural Networks: Architectures, Algorithms, and Applications*, Prentice-Hall, New Jersey, 1994.
- Simon Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd, Prentice-Hall, New Jersey, 1999.

Sekian

Agust Isa Martinus

aimxx@yahoo.com

<http://www.GusMartinus.mine.nu>

BACK

Agust Isa Martinus @2008/2022

<aimxx@yahoo.com>