

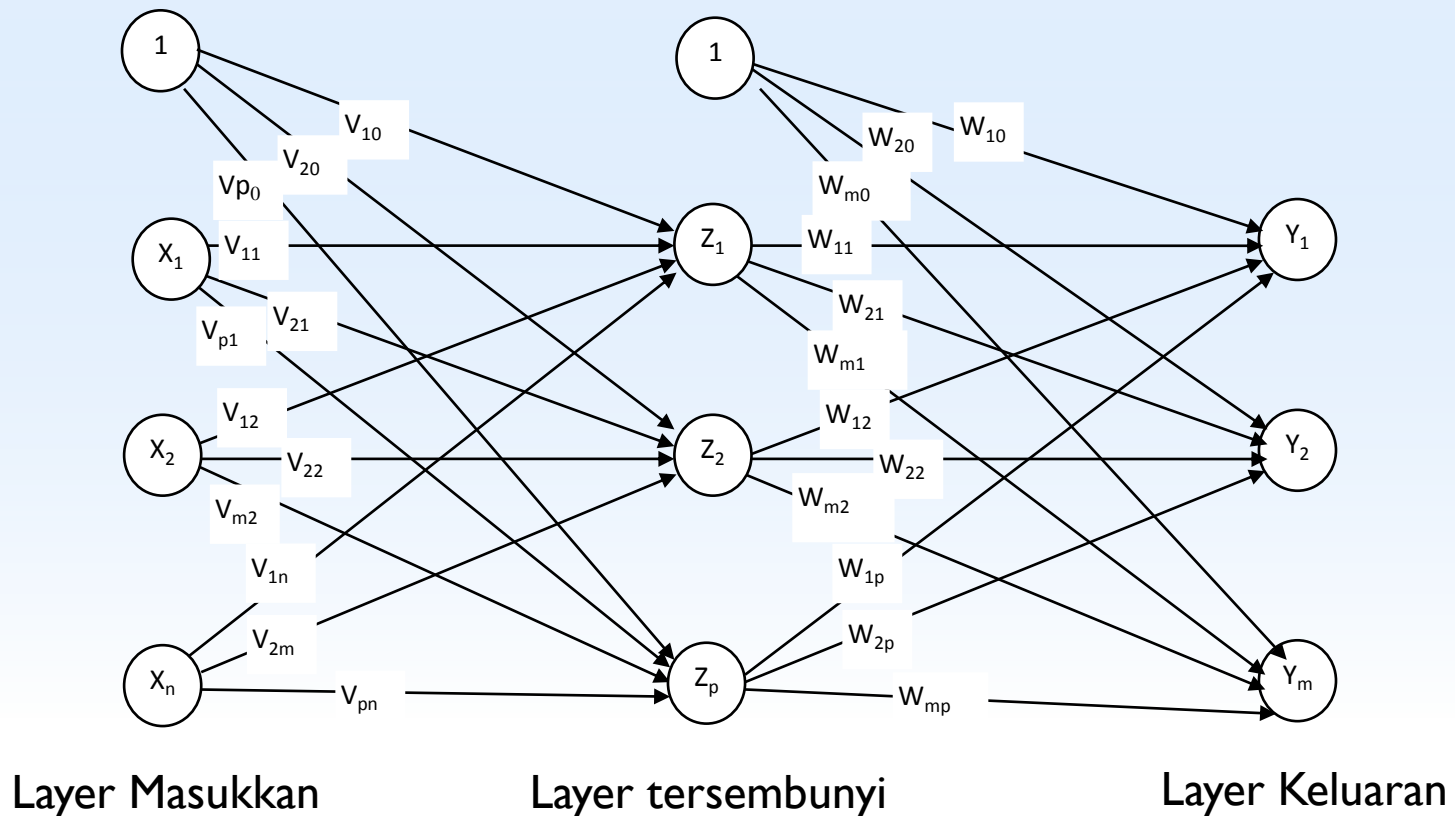
# Algoritma JST Backpropagation



# Arsitektur JST Backpropagation

Contoh Arsitektur JST Backpropagation dengan:

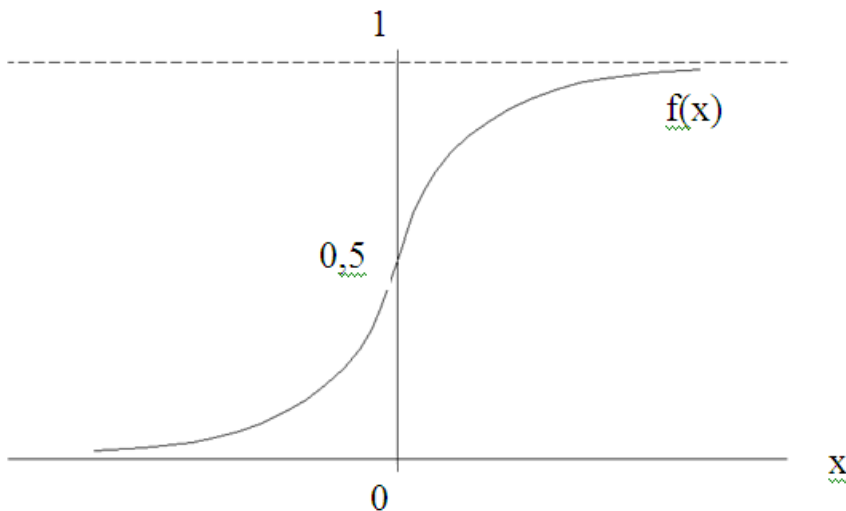
- ▣ n unit masukan
- ▣ p unit layer tersembunyi
- ▣ m unit keluaran



# Fungsi Aktifasi

Fungsi aktifasi yang digunakan pada backpropagation yaitu sigmoid biner dan sigmoid bipolar

Fungsi sigmoid biner  
Grafik



Mempunyai Range 0 sampai dengan 1

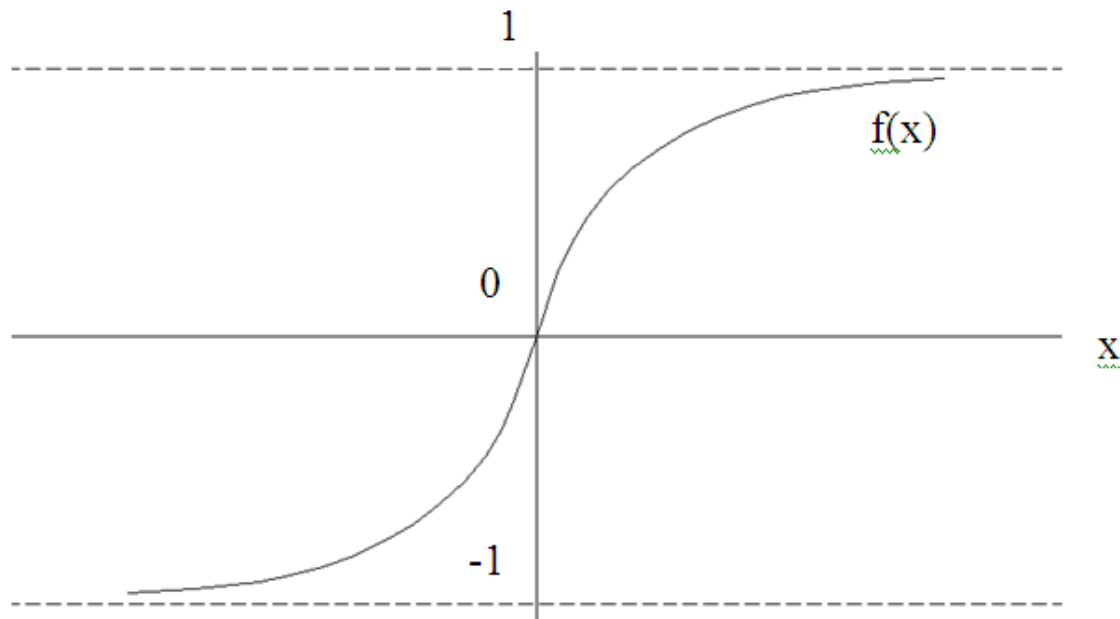
Fungsi:

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

# Fungsi Aktifasi

Fungsi sigmoid bipolar

Grafik



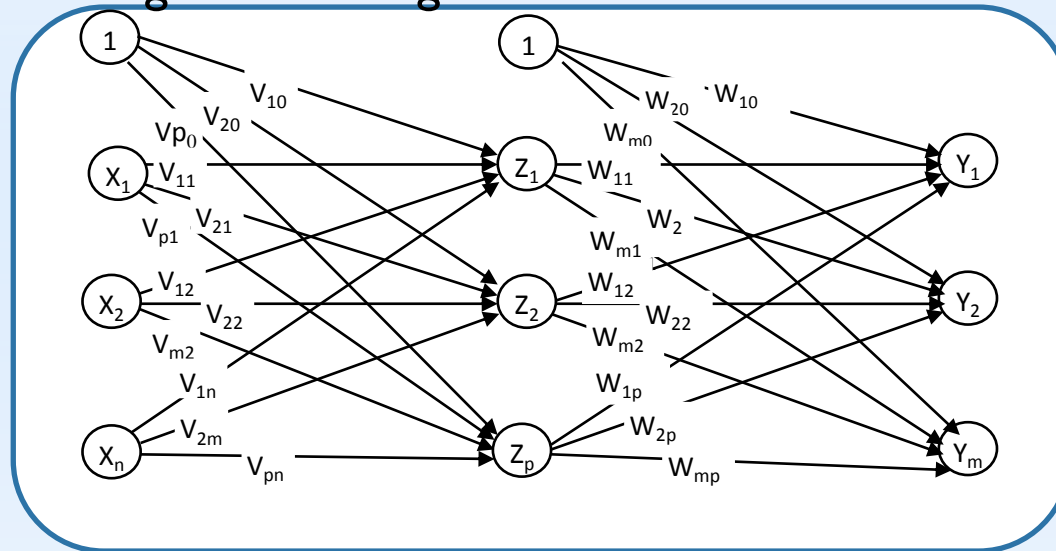
Mempunyai Range -1 sampai dengan 1

Fungsi:

$$f(x) = \frac{2}{1 + e^{-x}} - 1$$

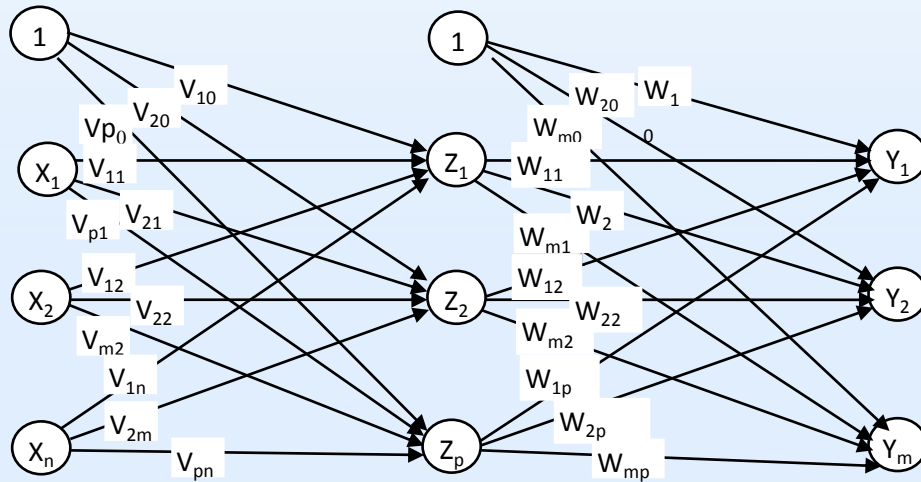
# Algoritma Pelatihan Backpropagation

Algoritma Pelatihan Backpropagation dengan satu layer tersembunyi dan dengan menggunakan fungsi aktivasi sigmoid biner



- **Langkah 0** : Inisialisasi semua bobot dengan bilangan acak kecil.
- **Langkah 1** : Jika kondisi penghentian belum terpenuhi, lakukan langkah 2 sampai dengan 8.
- **Langkah 2**: Untuk setiap pasang data pelatihan, lakukan langkah 3 sampai dengan 8

# Algoritma Pelatihan Backpropagation



Fase I: Propagasi Maju

▣ **Langkah 3**

Tiap unit masukkan menerima sinyal dan meneruskan ke unit tersembunyi

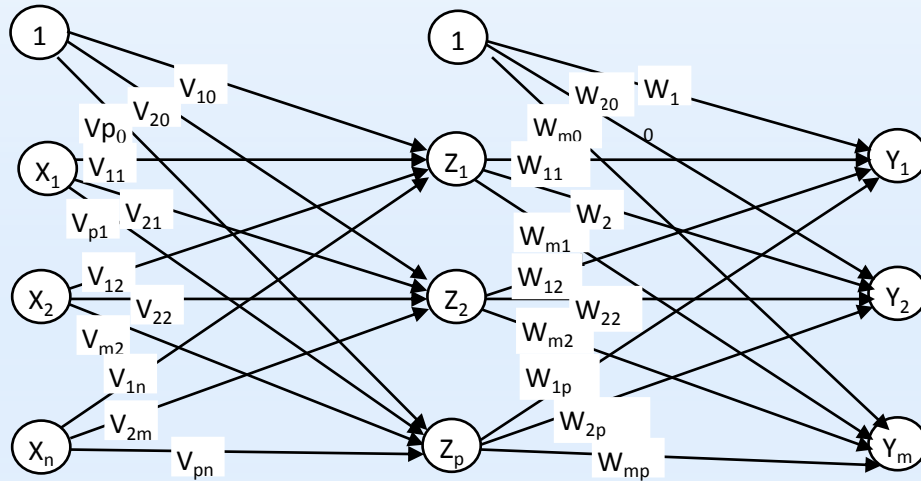
▣ **Langkah 4**

Hitung semua keluaran di unit tersembunyi ( $Z_j$ ):

$$z_{net_j} = v_{j0} + \sum_{i=1}^n x_i v_{ji}$$

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

# Algoritma Pelatihan Backpropagation



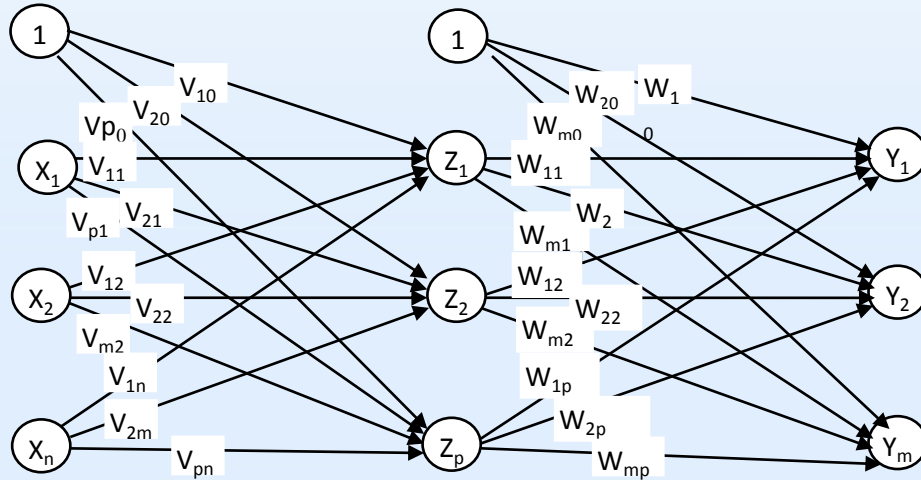
## Langkah 5

Hitung semua jaringan di unit keluaran ( $y_k$ )

$$y_{net_k} = w_{k0} + \sum_{j=1}^p x_j v_{kj}$$

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}}$$

# Algoritma Pelatihan Backpropagation



Fase II : Propagasi Maju

## Langkah 6

Hitung factor  $\delta$  unit keluaran berdasarkan kesalahan setiap unit keluaran  $y_k$  ( $k=1,2,3,\dots$ )

$$\delta_k = (t_k - y_k) f'(y_{\text{net}_k}) = (t_k - y_k) y_k (1 - y_k)$$

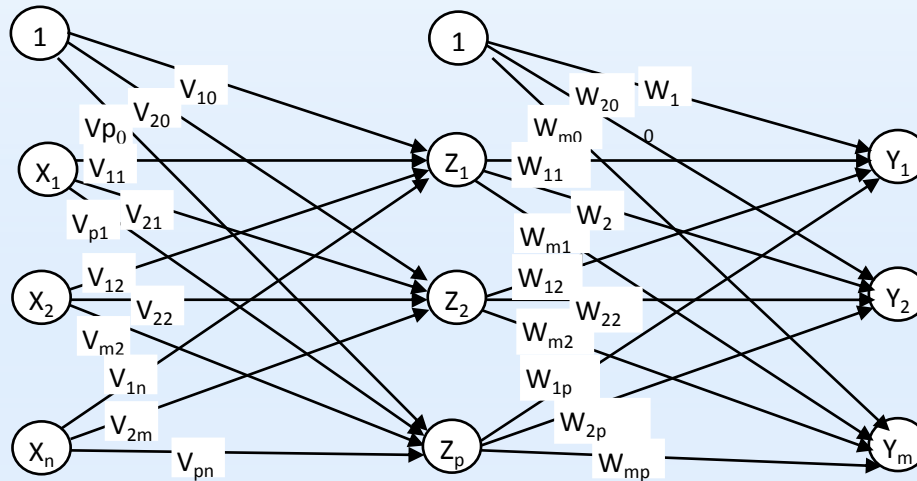
$\delta_k$  merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layer dibawahnya (langkah 7)

Hitung suku perubahan bobot  $W_{kj}$  dengan laju perubahan  $\alpha$

$$\Delta w_{kj} = \alpha \delta_k z_j \quad ; k=1,2,3,\dots,m \quad ; j=0,1,2,\dots,p$$



# Algoritma Pelatihan Backpropagation



## Langkah 7

Hitung factor  $\delta$  unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi  $z_j$  ( $j=1,2,3,\dots,p$ )

$$\delta_{net_j} = \sum_{k=1}^m \delta_k w_{kj}$$

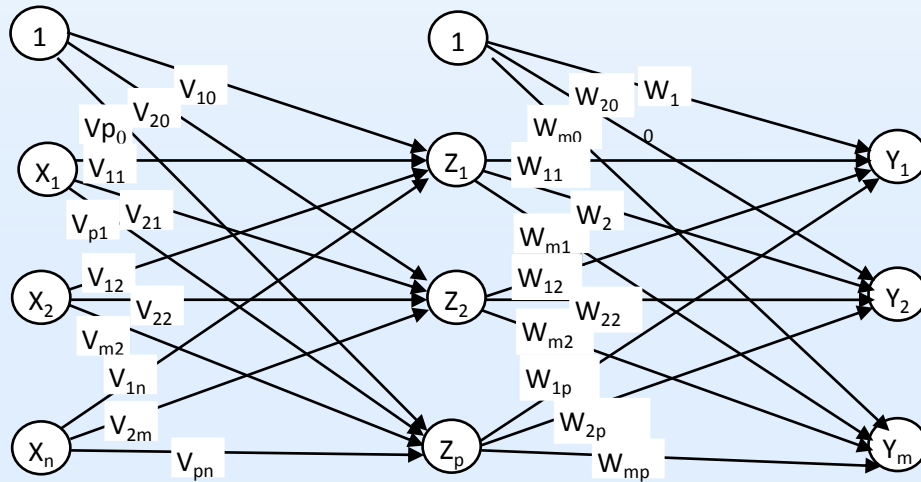
Faktor unit tersembunyi

$$\delta_j = \delta_{net_j} f'(z_{net_j}) = \delta_{net_j} z_j (1 - z_j)$$

Hitung suku perubahan bobot  $v_{ji}$

$$\Delta v_{ji} = \alpha \delta_j x_i \quad ; j=1,2,\dots,p \quad ; i=0,1,2,\dots,n$$

# Algoritma Pelatihan Backpropagation



Fase III : Perubahan Bobot

## Langkah 8

Perubahan bobot garis yang menuju unit keluaran

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

Perubahan bobot garis yang menuju ke unit tersembunyi

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

# Laju Pemahaman

- ▣ Laju pemahaman di simbolkan dengan  $\alpha$
- ▣ Laju pemahaman menentukan lama iterasi
- ▣ Nilai dari  $\alpha$  diantara 0 sd 1
- ▣ Semakin besar nilai  $\alpha$  semakin cepat lama iterasi
- ▣ Akan tetapi jika terlalu besar akan merusak pola, sehingga justru akan lebih lama iterasinya

# Epoch

- Epoch yaitu satu siklus pelatihan yang melibatkan semua pola
- Misal jika suatu arsitektur JST terdapat 4 pola masukkan dan 1 target, maka pelatihan 4 pola masukkan tersebut adalah 1 epoch