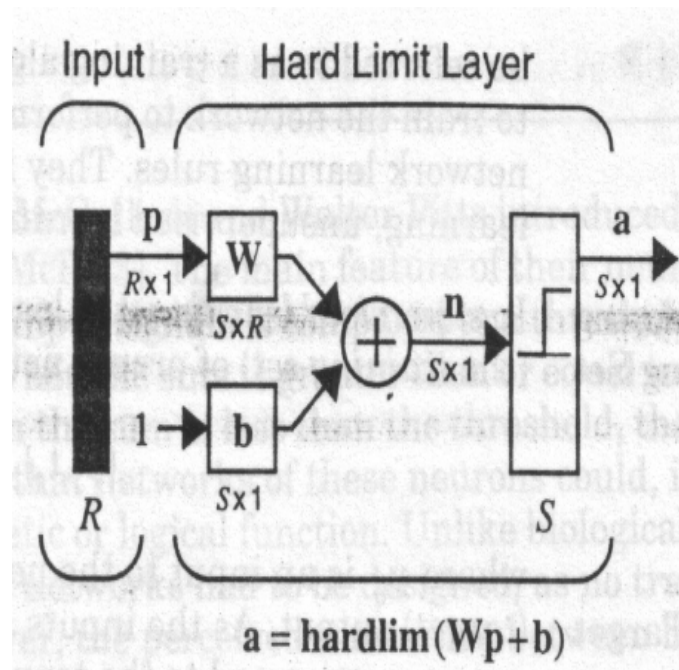


BAB IV

PERSEPTRON

- **Aturan belajar :**
prosedur pengubahan bobot dan bias dalam jaringan.
- **Pembelajaran tersupervisi :**
aturan belajar yang menggunakan contoh, yang digunakan sebagai target.
- **Reinforcement learning :**
Mirip pembelajaran tersupervisi, namun keluarannya dinyatakan dalam bentuk penilaian terhadap sekuens input
- **Pembelajaran tak tersupervisi :**
Modifikasi bobot dan bias tergantung pada respons jaringan terhadap input.

4.1 Arsitektur Perseptron



Matriks bobot untuk jaringan ini :

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \vdots & \vdots & \ddots & \vdots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix}$$

composed of the elements

$${}_iW = \begin{bmatrix} w_{i,1} \\ w_{i,2} \\ \vdots \\ w_{i,R} \end{bmatrix}$$

the weight matrix:

$$W = \begin{bmatrix} {}_1W^T \\ {}_2W^T \\ \vdots \\ {}_SW^T \end{bmatrix}$$

Elemen vektor pada baris ke-i :

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \vdots & \vdots & \ddots & \vdots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix}$$

composed of the elements

$${}_iW = \begin{bmatrix} w_{i,1} \\ w_{i,2} \\ \vdots \\ w_{i,R} \end{bmatrix}$$

the weight matrix:

$$W = \begin{bmatrix} {}_1W^T \\ {}_2W^T \\ \vdots \\ {}_SW^T \end{bmatrix}$$

Matriks bobot dapat ditulis dalam bentuk yang dipartisi sbb. :

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \vdots & \vdots & \ddots & \vdots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix}$$

composed of the elements

$${}_iW = \begin{bmatrix} w_{i,1} \\ w_{i,2} \\ \vdots \\ w_{i,R} \end{bmatrix}$$

the weight matrix:

$$W = \begin{bmatrix} {}_1W^T \\ {}_2W^T \\ \vdots \\ {}_SW^T \end{bmatrix}$$

4.2 Aturan belajar Perseptron

Perseptron dilatih dengan menggunakan satu set contoh pasangan masukan-keluaran yang diinginkan, yang dapat dinyatakan sebagai berikut :

$$p_1 t_1, p_2 t_1, \dots, p_Q t_Q$$

Digunakan variabel a (keluaran jaringan), t (nilai target), dan e (galat / *error*)

Tiga kemungkinan kasus dalam pembelajaran perseptron ini :

Kasus 1. Jika keluaran neuron benar ($a = t$, dan $e = t - a = 0$), maka bobot tidak diubah. (Jika $e = 0$, maka $\Delta \mathbf{w} = 0$)

Kasus 2. Jika keluaran neuron bernilai 0, namun seharusnya bernilai 1 ($a = 0$ dan $t = 1$, dan $e = t - a = 1$), maka vektor masukan \mathbf{p} ditambahkan pada bobot \mathbf{w} . Ini akan membuat vektor bobot menjadi lebih dekat ke input, dan meningkatkan kemungkinan diklasifikasikannya input sebagai 1 pada proses selanjutnya. (Jika $e = 1$, maka $\Delta \mathbf{w} = \mathbf{p}^T$)

Kasus 3. Jika keluaran neuron 1, namun seharusnya 0 ($a = 1$ dan $t = 0$, dan $e = t - a = -1$), vektor input \mathbf{p} digunakan untuk mengurangi vektor bobot \mathbf{w} , sehingga meningkatkan peluang terjadinya keluaran 0 pada proses berikutnya. (Jika $e = -1$, maka $\Delta \mathbf{w} = -\mathbf{p}^T$)

Ketiga kasus di atas dapat diekspresikan sbb. :

$$\Delta \mathbf{w} = (t - a)\mathbf{p}^T = e\mathbf{p}^T$$

Perubahan vektor bias sama dengan perubahan vektor bobot, dengan masukan bernilai 1. Sehingga :

$$\Delta \mathbf{b} = (t - a)(1) = e$$

Untuk lapisan yang mengandung banyak neuron, berlaku :

$$\Delta \mathbf{W} = (\mathbf{t} - \mathbf{a})(\mathbf{p})^T = \mathbf{e}(\mathbf{p})^T$$

$$\Delta \mathbf{b} = (\mathbf{t} - \mathbf{a}) = \mathbf{E}$$

Dengan demikian, *Aturan Belajar Perseptron* adalah sbb. :

$$\mathbf{W}^{new} = \mathbf{W}^{old} + \mathbf{e}\mathbf{p}^T$$

$$\mathbf{b}^{new} = \mathbf{b}^{old} + \mathbf{e}$$

dengan $e = t - a$.