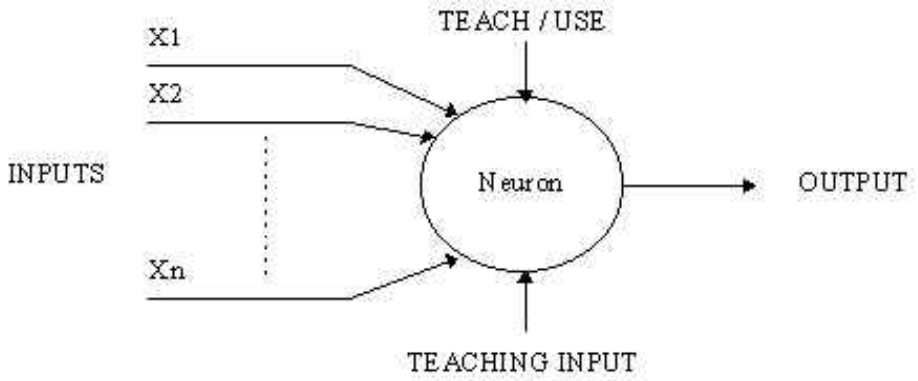
1

CONTOH ILUSTRATIF

1 Sebuah Neuron Sederhana

- Suatu artificial neuron adalah suatu divais dengan banyak input dan satu output.
- Neuron mempunyai dua model operasi; model *training*/pelatihan dan model *using*.
- Dalam model pelatihan, neuron dapat dilatih untuk menembak (atau tidak), untuk pola input tertentu.
- Dalam model *using*, ketika satu pola input diajarkan ia dideteksi di input, output yang berhubungan akan menjadi output saat itu. Jika pola input tidak menjadi anggota di dalam daftar pola input yang diajarkan, *firing rule* digunakan untuk menentukan apakah *fire*/menembak atau tidak.



1

Firing Rule

- *Firing rule* adalah satu konsep penting di dalam jaringan neural dan menghitung untuk tingkat fleksibilitas mereka. Sebuah *firing rule* menentukan bagaimana mengkalkulasi apakah satu neuron harus *fire* untuk pola input apapun. Ia berhubungan dengan seluruh pola input, tidak hanya dengan satu node yang dilatih.
- Sebuah *firing rule* sederhana dapat diterapkan dengan menggunakan teknik *Hamming dista*nce. Aturan tersebut adalah sebagai berikut:

  Ambil satu koleksi pola pelatihan untuk satu node, beberapa diantaranya menyebabkan ia untuk *fire* (*1-taught set of patterns*) hal lainnya mencegahnya dari melakukan *fire* (*0-taught set*). Kemudian pola yang tidak dalam koleksi menyebabkan node untuk *fire* jika pada perbandingan, mereka memiliki lebih banyak elemen input secara umum 'paling dekat' dengan pola dalam 1 dibandingkan dengan 'paling dekat' pola di 0. Jika ada sebuah hasil kalkulasi yang sama, maka pola dalam keadaan tak tergambarkan.

1

Contoh

- Sebagai contoh, suatu neuron 3-input mengajarkan untuk output 1 ketika input (X1,X2 dan X3) adalah 111 atau 101 dan untuk output 0 ketika input adalah 000 atau 001. Kemudian, sebelum menerapkan *firing rule*, tabel kebenaran adalah;

X1:		0	0	0	0	1	1	1	1
X2:		0	0	1	1	0	0	1	1
X3:		0	1	0	1	0	1	0	1
OUT:		0	0	0/1	0/1	0/1	1	0/1	1

- Sebagai contoh dari cara *firing rule* diterapkan, ambil pola 010. Ia berbeda dengan 000 di1 elemen, 001 di 2 elemen, 101 di 3 elemen dan 111 di 2 elemen. Oleh karena itu, pola 'paling dekat' adalah 000 yang merupakan anggota dalam *0-taught set*. Dengan demikian firing rule memerlukan bahwa neuron tidak harus *fire* ketika input adalah 001.

1

Contoh

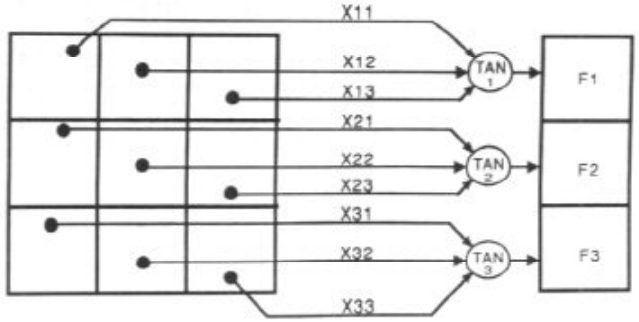
- Di sisi lainnya, 011 adalah sama jauhnya dari dua pola yang diajar yang mempunyai output berbeda dan dengan demikian output tetap tak tergambarkan (0/1). Sehingga tabel kebenaran berikut diperoleh;

X1:		0	0	0	0	1	1	1	1
X2:		0	0	1	1	0	0	1	1
X3:		0	1	0	1	0	1	0	1
OUT:		0	0	0	0/1	0/1	1	1	1

- Perbedaan antara kedua tabel kebenaran disebut generalisation neuron. Oleh karena itu *firing rule* memberikan neuron suatu makna kesamaan/*similarity* dan memungkinkannya untuk memberikan reaksi ‘yang sesuai)' ke pola yang tidak terlihat selama pelatihan.

1 Suatu Contoh – Pengenalan Pola

- Satu aplikasi penting dari jaringan neural adalah pengenalan pola. Pengenalan pola mungkin dapat diterapkan dengan menggunakan jaringan *neural feed-forward* (Gambar 2) yang telah dilatih. Selama pelatihan, jaringan dilatih untuk menghubungkan output dengan pola input. Ketika jaringan digunakan, ia mengidentifikasikan pola input dan mencoba menghasilkan keluaran pola output yang berhubungan. Kekuatan jaringan neural datang ketika satu pola yang tidak mempunyai output berhubungan dengannya, yang diberikan sebagai satu input. Dalam hal ini, jaringan memberikan output yang menyesuaikan dengan satu pola input yang diajarkan yang **paling mendekati** pola yang diberikan.



Gambar 2. Jaringan neural feed-forward

1 Contoh Pengenal Pola

Sebagai contoh:

- Jaringan dari gambar 1 dilatih untuk mengenali pola T dan H. Pola yang berhubungan adalah semua yang hitam dan semua yang putih berturut-turut sebagaimana yang diperlihatkan di bawah.

- Jika kita merepresentasikan bujur sangkar hitam dengan 0 dan bujur sangkar putih dengan 1 maka tabel kebenaran untuk 3 neuron setelah digeneralisasi adalah;

X11:		0	0	0	0	1	1	1	1
X12:		0	0	1	1	0	0	1	1
X13:		0	1	0	1	0	1	0	1
OUT:		0	0	1	1	0	0	1	1

Top neuron

1 Contoh Pengenal Pola

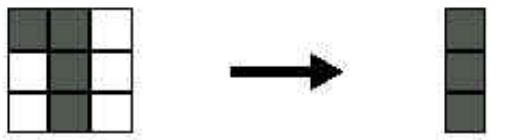
X21:		0	0	0	0	1	1	1	1
X22:		0	0	1	1	0	0	1	1
X23:		0	1	0	1	0	1	0	1
OUT:		1	0/1	1	0/1	0/1	0	0/1	0

Middle neuron

X21:		0	0	0	0	1	1	1	1
X22:		0	0	1	1	0	0	1	1
X23:		0	1	0	1	0	1	0	1
OUT:		1	0	1	1	0	0	1	0

Bottom neuron

- Dari tabel tsb dapat dilihat keterhubungan berikut dapat di-ekstrak:

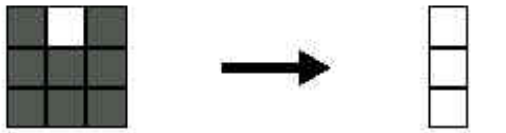


INPUT OUTPUT

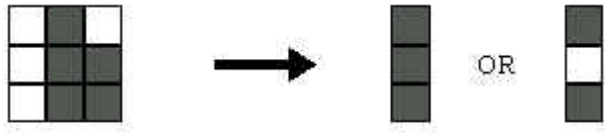
1

Contoh Pengenal Pola

- Dalam hal ini, adalah jelas nyata bahwa output harus semua hitam karena pola input adalah hampir sama halnya pola 'T'.

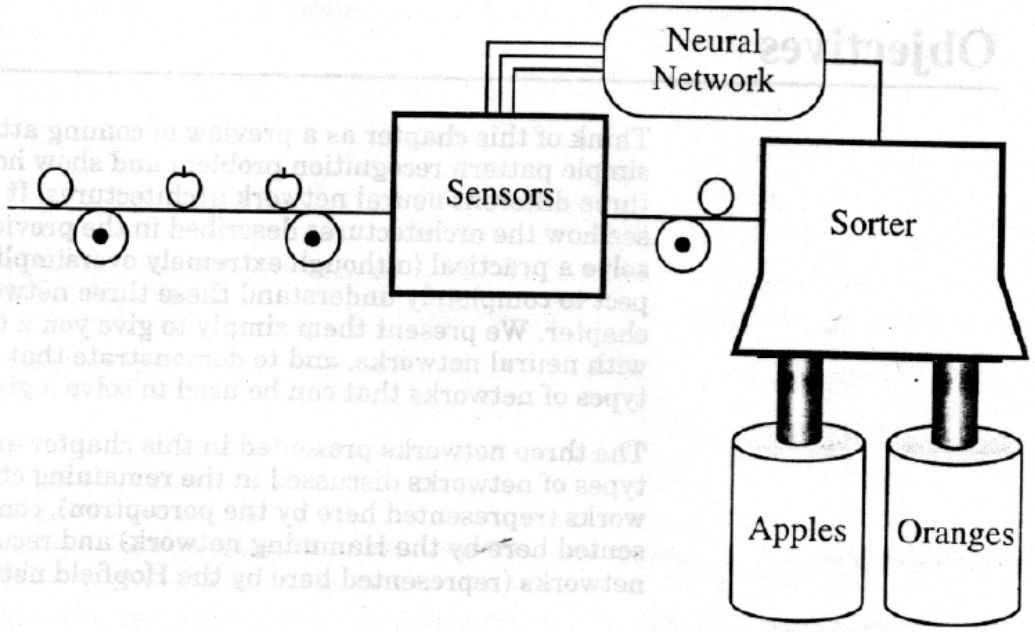


- Di sini juga, adalah jelas nyata bahwa output harus semua putih karena pola input adalah hampir sama halnya pola 'H'.



- Di sini, baris puncak adalah 2 error dari T dan 3 dari H. Sehingga output puncak adalah hitam. Baris pertengahan adalah 1 error untuk keduanya, yaitu T dan H sehingga output adalah acak. Baris paling bawah adalah 1 error dari T dan 2 dari H. Oleh karena itu output adalah hitam. Total output dari jaringan adalah masih mirip pada bentuk T.

1 CONTOH ILUSTRATIF



- Pada gambar di atas, setiap buah yang memasuki sensor akan direpresentasikan dalam bentuk vektor tiga dimensi :

$$\mathbf{p} = \begin{bmatrix} bentuk \\ tekstur \\ berat \end{bmatrix}$$

1 CONTOH ILUSTRATIF

- Dengan menggunakan ketentuan berikut ini :

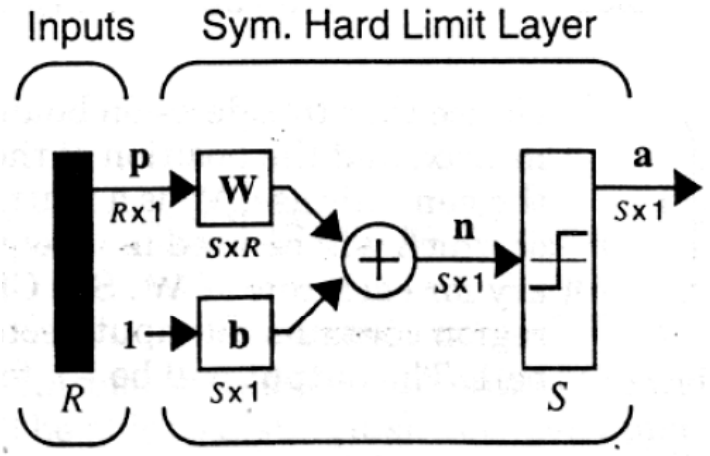
Elemen	Nilai	Arti
bentuk	1 dan –1	bulat dan lonjong
tekstur	1 dan –1	halus dan kasar
berat	1 dan –1	> 1 pound dan < 1 pound

- prototip jeruk akan direpresentasikan sebagai $\mathbf{p}_1 = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}$

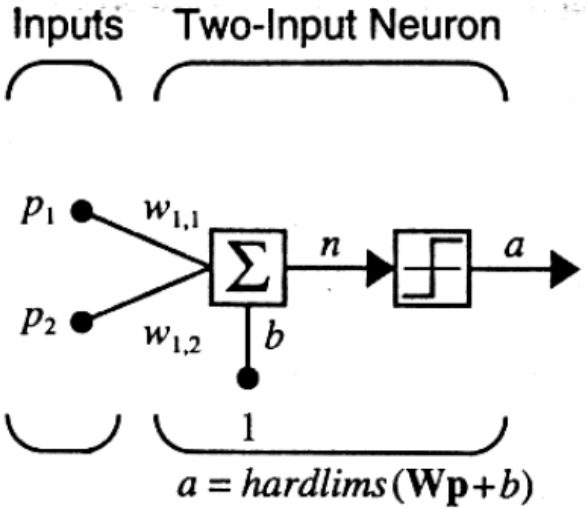
- prototip apel akan direpresentasikan sebagai $\mathbf{p}_2 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$

1

Perseptron



- **Kasus dua input**

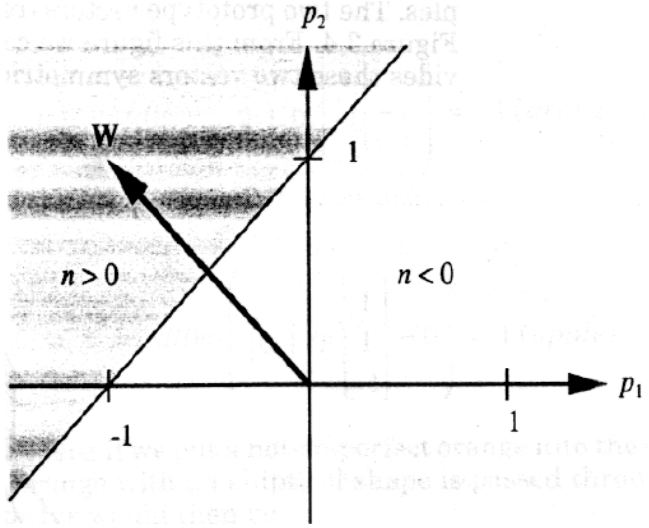


1

Perseptron

- Jika bobot $w^{1,1} = -1$ dan bobot $w^{1,2} = 1$, maka keluaran neuron adalah :

  $a = hardlims\ (n) = hardlims\ ([-1\ 1])\mathbf{p} + \mathbf{b})$
- Jika perkalian menghasilkan nilai $\geq -b$ *maka output bernilai 1.*
- Jika perkalian menghasilkan nilai $< -b$ *maka output bernilai –1.*
- Dengan asumsi bahwa $b = -1$, *dapat dibuat batas keputusan seperti pada gambar berikut ini*

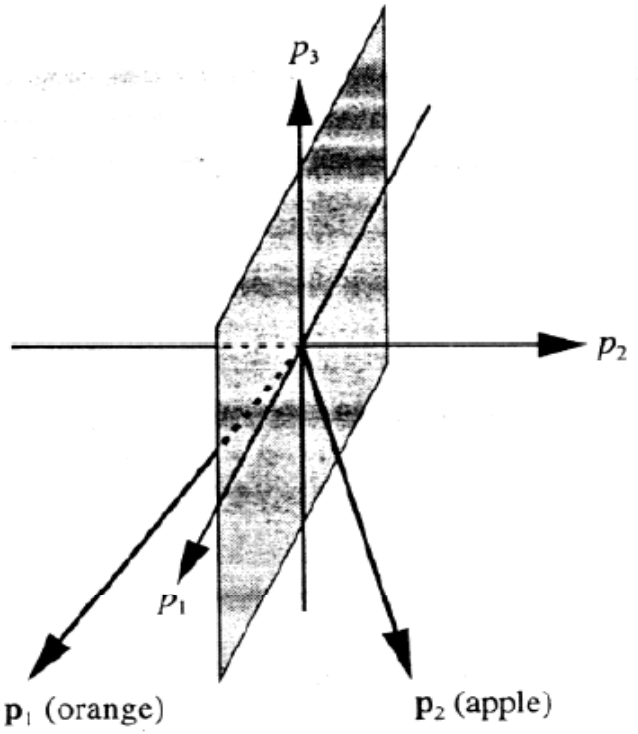


1 Contoh Pengenalan Pola

- Untuk masalah pengenalan apel dan jeruk di atas, karena hanya terdapat dua kategori, maka dapat digunakan perseptron satu neuron. Dengan vektor input tiga dimensi, persamaan perseptron menjadi :

$$a = hardlims\left(\begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} + b \right)$$

- Bidang batas untuk masalah ini adalah sbb. :



1 Contoh Pengenalan Pola

- Dengan matriks bobot **W = [0 1 0]** dan bias ***b* = *0*,** maka klasifikasi apel dan jeruk dapat berlangsung sempurna, karena :

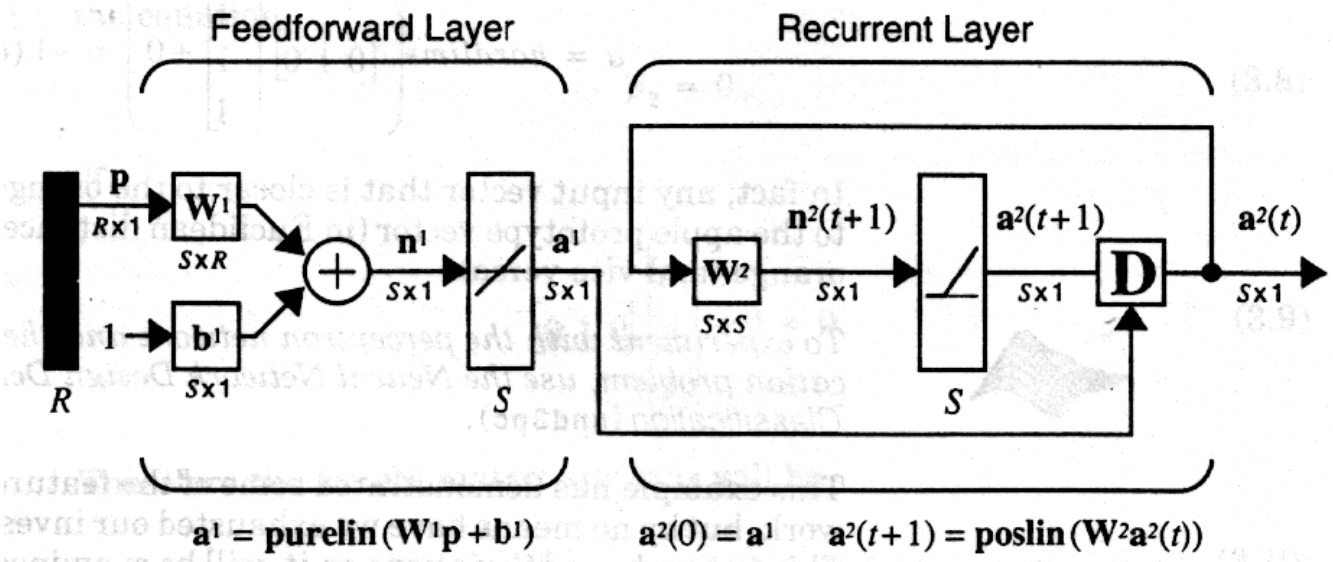
Orange:

$$a = hardlims\left(\begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} + 0 \right) = -1\,(orange)\,,$$

Apple:

$$a = hardlims\left(\begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} + 0 \right) = 1\,(apple)\,.$$

1 Jaringan Hamming



- ***Feedforward layer***

Matriks bobot :

$$\mathbf{W}^1 = \begin{bmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 \\ 1 & 1 & -1 \end{bmatrix}$$

Jika vektor bias :

$$\mathbf{b}^1 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

Jaringan Hamming

maka keluaran *feedforward layer :*

$$\mathbf{a}^1 = \mathbf{W}^1\mathbf{p} + \mathbf{b}^1 = \begin{bmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \end{bmatrix}\mathbf{p} + \begin{bmatrix} 3 \\ 3 \end{bmatrix} = \begin{bmatrix} \mathbf{p}_1^T\mathbf{p} + 3 \\ \mathbf{p}_2^T\mathbf{p} + 3 \end{bmatrix}$$

- ***Recurrent layer***

  Persamaan kompetisi : **a2 (0) = a1 (kondisi awal) dan a2(t + 1) = poslin(W2a2(t))** (superscript mengindikasikan nomor lapisan, bukan pangkat !)

  Matriks bobot : $\mathbf{W}^2 = \begin{bmatrix} 1 & -\varepsilon \\ -\varepsilon & 1 \end{bmatrix}$

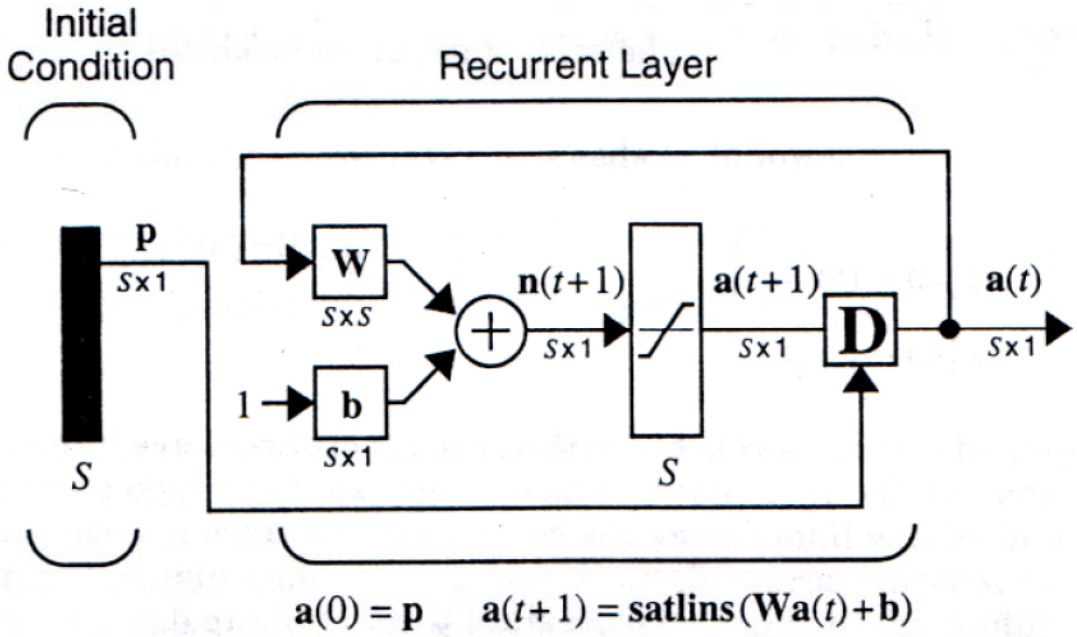
- dengan ε adalah sebuah nilai yang lebih kecil dari 1/(S – 1), dan S adalah jumlah neuron pada lapisan *recurrent*

1 Jaringan Hopfield

- Iterasi lapisan *recurrent berlangsung sbb. :*

$$\mathbf{a}^2(t+1) = \mathbf{poslin}\left(\begin{bmatrix} 1 & -\varepsilon \\ -\varepsilon & 1 \end{bmatrix}\mathbf{a}^2(t)\right) = \mathbf{poslin}\left(\begin{bmatrix} a_1^2(t) - \varepsilon a_2^2(t) \\ a_2^2(t) - \varepsilon a_1^2(t) \end{bmatrix}\right)$$

- Jaringan Hopfield



1

Jaringan Hopfield

- Jika matriks bobot dan bias : $\mathbf{W} = \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 1.2 & 0 \\ 0 & 0 & 0.2 \end{bmatrix}; \mathbf{b} = \begin{bmatrix} 0.9 \\ 0 \\ -0.9 \end{bmatrix}$ maka

  operasi jaringan Hopfield berturut-turut adalah :

  - $a\,1(t+1) = satlins(0.2\,a1\,(t) + 0.9)$
  - $a\,2(t+1) = satlins(1.2\,a2\,(t))$
  - $a\,3(t+1) = satlins(0.2\,a3\,(t) - 0.9)$

  Digunakan contoh jeruk (***p1)*** dan apel ***(p2)*** di atas. Faktor pembeda kedua objek terdapat pada elemen ke dua vektor ciri, sementara elemen-1 dan elemen-3 masing-masing bernilai 1 dan –1.

1 Jaringan Hopfield

- Jika elemen ke-2 diberi nilai 1, maka hasil iterasi adalah sbb. :

$$\mathbf{a}(0) = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}; \mathbf{a}(1) = \begin{bmatrix} 0.7 \\ -1 \\ -1 \end{bmatrix}; \mathbf{a}(2) = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}; \mathbf{a}(3) = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}$$

- yang dengan jelas menunjuk kategori **jeruk.**