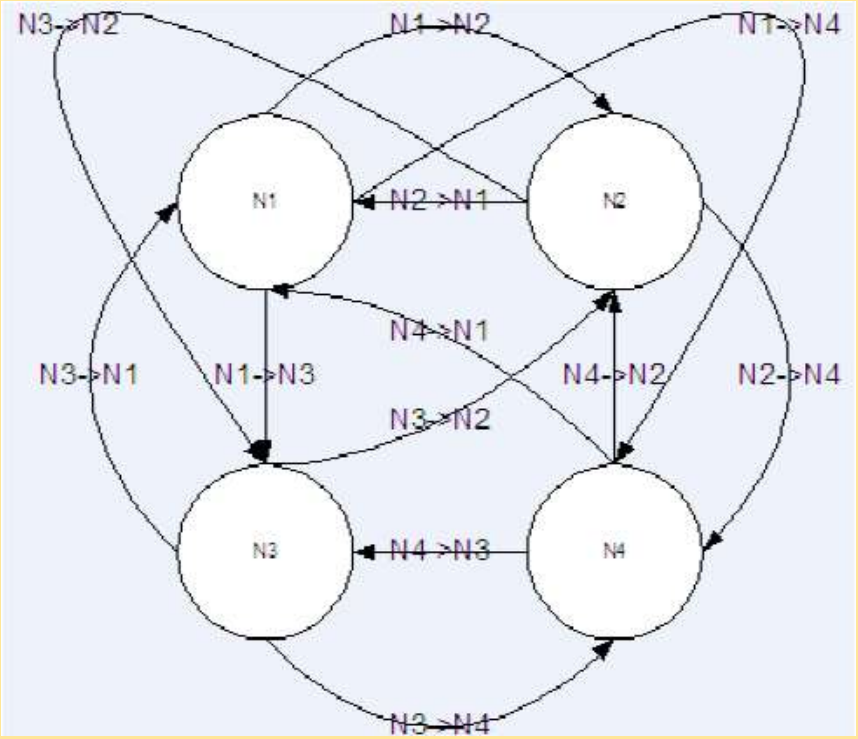
Jaringan Hopfield Diskrit

Jaringan Hopfield

- JST yang fully connected : setiap unit terhubung dengan unit yang lainnya.
- Memiliki bobot-bobot yang simetris : $w_{ij} = w_{ji}$ untuk $i \neq j$
- Tidak memiliki hubungan dengan dirinya sendiri, $w_{ij} = 0$ untuk $i = j$
- Fungsi aktivasi menggunakan fungsi energi Lyapunov : fungsi yang terbatas dan menurun untuk mendapat kestabilan pada aktivasinya.

Jaringan Hopfield



Fungsi aktivasi

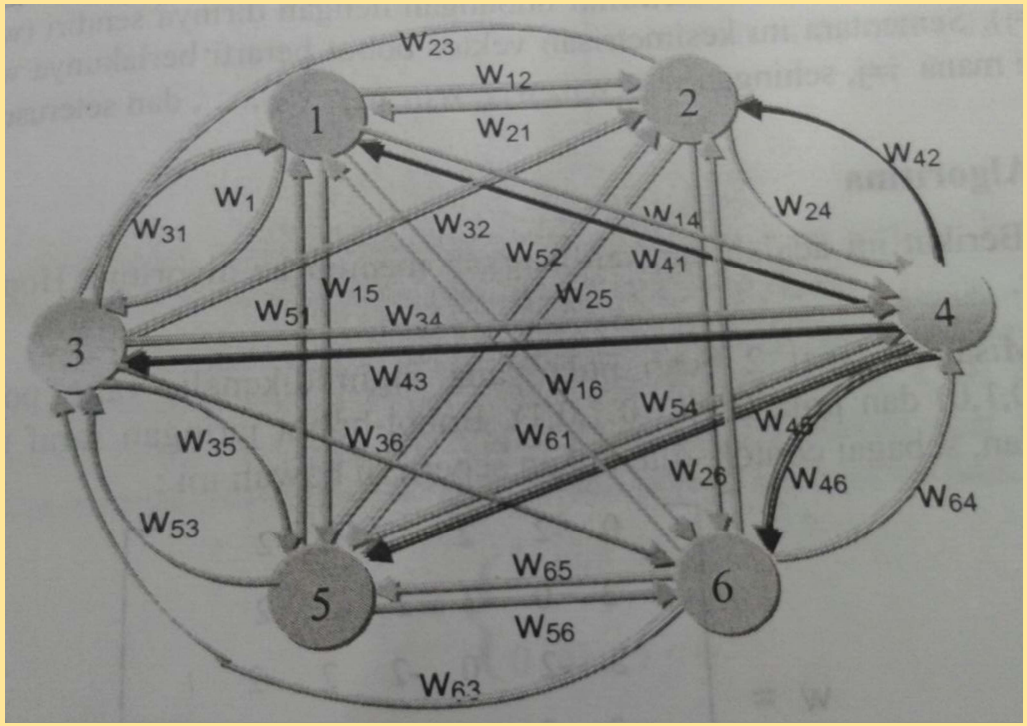
$$F(t) = \begin{cases} 1 \; jika \; t \geq \theta \\ 0 \; jika \; t < \theta \end{cases} \text{ atau } F(t) = \begin{cases} 1 \; jika \; t \geq \theta \\ -1 \; jika \; t < \theta \end{cases}$$

Dimana :

t : aktivasi node

θ : nilai ambang

Jaringan hopfield 6 neuron



$$\begin{bmatrix} 0 & W_{12} & W_{13} & W_{14} & W_{15} & W_{16} \\ W_{21} & 0 & W_{23} & W_{24} & W_{25} & W_{26} \\ W_{31} & W_{32} & 0 & W_{34} & W_{35} & W_{36} \\ W_{41} & W_{42} & W_{43} & 0 & W_{45} & W_{46} \\ W_{51} & W_{52} & W_{53} & W_{54} & 0 & W_{56} \\ W_{61} & W_{62} & W_{63} & W_{64} & W_{65} & 0 \end{bmatrix}$$

Kasus 1

- Ada 2 buah pola yg ingin dikenali:

  pola A (1,0,1,0,1,0) C (1,0,1,0,0,0)

  pola B (0,1,0,1,0,1) D (0,0,0,1,0,1)

  Pola A dan B diperlakukan sebagai vektor. Dot product antara A dan B diperoleh dengan cara mengalikan komponen kedua vektor tersebut dengan vektor bobot W.

- Bobot-bobotnya sbb:

$$W = \begin{bmatrix} 0 & -2 & 2 & -2 & 2 & -2 \\ -2 & 0 & -2 & 2 & -2 & 2 \\ 2 & -2 & 0 & -2 & 2 & -2 \\ -2 & 2 & -2 & 0 & -2 & 2 \\ 2 & -2 & 2 & -2 & 0 & -2 \\ -2 & 2 & -2 & 2 & -2 & 0 \end{bmatrix}$$

Algoritma

- Aktivasi node pertama pola A

$$(1\ 0\ 1\ 0\ 1\ 0).\begin{bmatrix}0\\-2\\2\\-2\\2\\-2\end{bmatrix} = 0 + 0 + 2 + 0 + 2 + 0 = 4$$

- Aktivasi node kedua pola A

$$(1\ 0\ 1\ 0\ 1\ 0).\begin{bmatrix}-2\\0\\-2\\2\\-2\\2\end{bmatrix} = (-2) + 0 + (-2) + 0 + (-2) + 0 = -6$$

- Node 3-6 hasilnya 4,-6,4,-6
- cara yg sama lakukan utk pola B yg hasilnya -6,4,-6,4,-6,4

Output

- Hitung output dengan belrdasarkan fungsi aktivasi Lyapunov di mana Ɵ biasanya sama dengan 0.

- Output untuk pola A dan B setelah dihitung fungsi aktivasinya :
  - A = (1,0,1,0,1,0) → sama dengan pola input
  - B = (0,1,0,1,0,1) → sama dengan pola input

- Pola A dan B memiliki pola yang sama dengan pola input sehingga A dan B dikatakan sukses dipanggil kembali dengan menggunakan bobot tersebut.

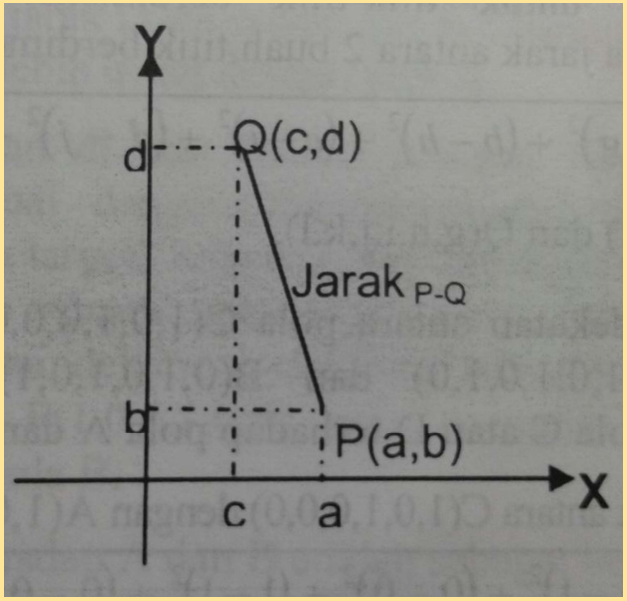
- Pola A dan B akan tersimpan sebagai target output

pengujian

- Mengenali pola C (1,0,1,0,0,0) yang dianggap sebagai citra pola A yg mengalami distorsi
- Aktivasi node 1-6 menghasilkan (2,-4,2,-4, 4,-4), maka output (1,0,1,0,1,0)
- Mengenali pola D (0,0,0,1,0,1) dianggap citra pola B yg mengalami distorsi
- Bagaimana dg pola D?

- Kesimpulan : Jaringan Hopfield akan menghasilkan output yang sesuai dengan kedekatan Antara pola input dan **pola output (target).**
- Secara matematika kedekatan dua buah titik dimensi 2 (P(a,b) dan Q(c,d) dapat dihitung sebagai berikut :

$$Jarak\ _{PQ} = \sqrt{(a-c)^2 + (b-d)^2}$$

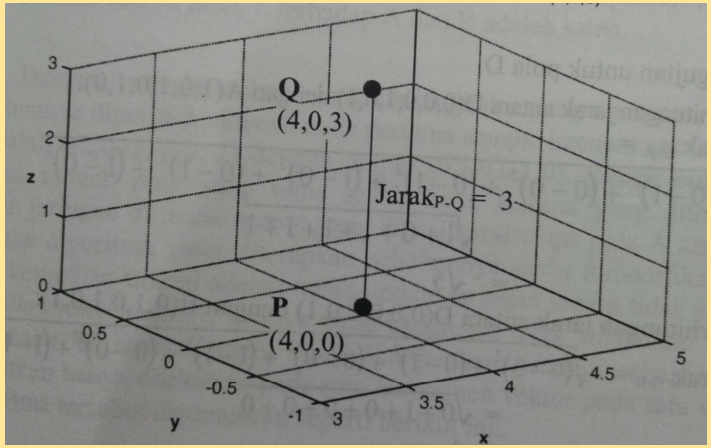


Jarak 2 buah titik

- Kedekatan 2 buah titik dimensi 3 (P(a,b,c) dan Q(d,e,f) dapat dihitung sebagai berikut :

$$Jarak\ _{PQ} = \sqrt{(a-d)^2 + (b-e)^2 + (c-f)^2}$$

- Sehingga jarak 2 buah titik dimensi 4, 5, 6 dan seterusnya dapat juga dihitung
- Kedekatan 2 buah titik ditandai dengan semakin kecilnya jarak Antara 2 buah titik tersebut



Algoritma dg Asynchronous update

- Mengenali pola E (1,0,1,1,0,1)
- Aktivasi node 1-6 diperoleh (-2,0,-2,-2,0,-2) dg output (0,1,0,0,1,0) -> **bukan A atau B**
- maka algoritma yang diterapkan sebelumnya perlu dimofdifikasi.
- Ide yang kemudian timbul adalah meng-update jaringan secara tidak sinkron (solusi dg **Asynchronous update**),
- artinya pengupdatean tidak dilakukan secara bersamaan ke semua output yang diumpankan kembali sebagai input melainkan hanya dilakukan pada satu komponen vektor pada satu waktu

algoritma

1. Inisialisasi matriks bobot W
2. Masukan vector input (invec), lalu inisialisasi vector output (outvec) yaitu outvec = invec
3. Mulai dg counter i=1

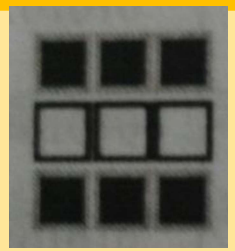
   Selama invec ≠ outvec lakukan langakh 4-7,jika I sampai maks maka reset mjd 1
4. Hitung nilai ke-i = dotproduct(invec, kolom ke-I dari W)
5. Hitung outvec ke-i = f(nilai ke-i), f adalah fungsi ambang
6. Update invec dg outvec
7. i=i+1

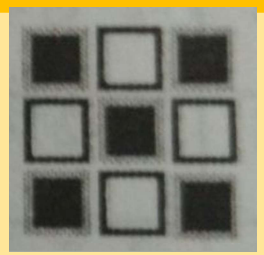
Aplikasi pd vektor E

Langkah	i	Vektor Input	Kolom Vektor Bobot	Nilai Aktivasi	Vektor Output	Catatan
0		1 0 1 1 0 1			1 0 1 1 0 1	Inisialisasi
1	1	1 0 1 1 0 1	0 -2 2 -2 2 -2	-2	0 0 1 1 0 1	Krn nilai aktivasi -2, Kolom ke-1 outvec diubah sesuai output Aktivasi (1 jad 0)
2	2	0 0 1 1 0 1	-2 0 -2 2 -2 2	2	0 1 1 1 0 1	F aktivasi = 2 , nilai 0 jadi 1 pada kolom 2 outvek
3	3	0 1 1 1 0 1	2 -2 0 -2 2 -2	-6	0 1 0 1 0 1	
4	4	0 1 0 1 0 1	-2 2 -2 0 -2 2	4	0 1 0 1 0 1	stabil
5	5	0 1 0 1 0 1	2 -2 2 -2 0 -2	-6	0 1 0 1 0 1	stabil
6	6	0 1 0 1 0 1	-2 2 -2 2 -2 0	4	0 1 0 1 0 1	stabil
7	1	0 1 0 1 0 1	0 -2 2 -2 2 -2	-6	0 1 0 1 0 1	stabil
8	2	0 1 0 1 0 1	-2 0 -2 2 -2 2	4	0 1 0 1 0 1	stabil
9	3	0 1 0 1 0 1	2 -2 0 -2 2 -2	-6	0 1 0 1 0 1	stabil
10	4	0 1 0 1 0 1	-2 2 -2 0 -2 2	4	0 1 0 1 0 1	stabil
11	5	0 1 0 1 0 1	2 -2 2 -2 0 -2	-6	0 1 0 1 0 1	stabil
12	6	0 1 0 1 0 1	-2 2 -2 2 -2 0	4	0 1 0 1 0 1	stabil

Vektor E dikenali sebagai pola B

Kasus 2 : Pengenalan pola “=“ dan “x”





- Pola “=“ : (1,1,1,-1,-1,-1,1,1,1)
- Pola “x” : (1,-1,1,-1,1,-1,1,-1,1)
- Bobot diset matrik (-3,3)
- Pola input “=“ nilai aktivasinya (3,3,3,-9,-6,-9,12,6,15), dg output (1,1,1,-1,-1,-1 ,1,1,1)
- Pola “x” nilai aktivasinya (9,-9,9,-9,6,-9,6,-6,9), dg output (1,-1,1,-1,1,-1,1,-1,1)
- Berarti jaringan telah sukses memanggil kembali pola-pola tsb

Vektor Bobot (-3,3)

Pola "=" : (1,1,1,-1,-1,-1,1,1,1)
Pola "x" : (1,-1,1,-1,1,-1,1,-1,1)

0	0	3	-3	0	-3	3	0	3
0	0	0	0	0	0	0	3	0
3	0	0	-3	0	-3	3	0	3
-3	0	-3	0	0	3	-3	0	-3
0	0	0	0	0	0	0	-3	0
-3	0	-3	3	0	0	-3	0	-3
3	0	3	-3	0	-3	0	0	3
0	3	0	0	-3	0	0	0	0
3	0	3	-3	0	-3	3	0	0

Spurious stable state

- Jaringan Hopfield diskrit dapat digunakan untuk menentukan apakah suatu vektor input “tidak dikenali” oleh jaringan.
- Disebut “dikenali” apabila output aktivasi yang dihasilkan jaringan sama dengan salah satu vektor yang disimpan oleh jaringan.
- Sebaliknya, jika vektor input “tidak dikenali” dan jaringan berkonvergensi menghasilkan sebuah vektor yang yang tidak merupakan salah satu pola yang disimpan dalam jaringan maka keadaan seperti ini disebut keadaan stabil palsu (*spurious stable state*).

Contoh :

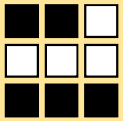
- Bagaimana jika dimasukan vektor input (-1,-1,-1,1,-1,1,-1,-1,-1)?

Catatan :

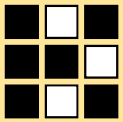
- Jaringan Hopfield dikatakan sampai kepada nilai maksimum jika sebuah pola tertentu stabil dipanggil ulang.
- Batas iterasi biasanya cukup satu kali siklus setelah pola tertentu dipanggil secara stabil.

tugas

- Apa output jaringan jika dimasuki input berikut?

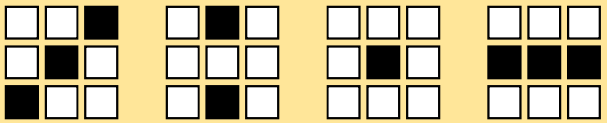


Pola '=' yg hilang 1 pixel (***incomplete data***)



Pola 'x' yg mendapat tambahan 1 pixel (***noisy data***)

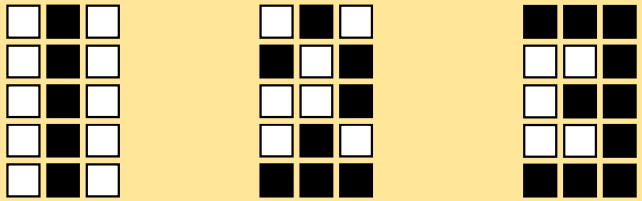
- Modifikasi matriks bobot dengan (-5,5), bagaimana bila dimasuki input bipolar pola-pola berikut:



- bandingkan hasilnya dg matriksbobot (-3,3)

project

- Buatlah Program yang mengimplementasikan jaringan hopfield diskrit yang menyimpan cukup salah satu pola numerik berikut:



- Dengan menyimpan satu pola, berapa banyak pola numerik yg dapat dipanggil kembali secara tepat?
- Dengan menyimpan dua pola, berapa banyak pola numerik yg dapat dipanggil kembali secara tepat?