



JARINGAN KOMPETISI

TEKNIK INFORMATIKA UNIVERSITAS SRIWIJAYA



Pelatihan Tanpa Supervisi

- Model pelatihan tanpa supervisi adalah model pelatihan dengan tidak ada target vektor yang dipakai sebagai tolok ukur kebenaran pelatihan.
- Salah satu model jaringan tanpa supervisi yang sering dipakai adalah **jaringan kompetisi**. Dalam model ini, neuron dipaksa untuk berkompetisi sehingga hanya satu diantaranya yang menjadi aktif (signal keluaran > 0). Prinsip seperti ini sering disebut “*winner takes all*”.



JARINGAN BERBASIS KOMPETISI

- Jaringan berbasis kompetisi menggunakan ide kompetisi untuk meningkatkan kontras dalam aktivasi neuron (winner take all).
- Hanya neuron yang aktivasinya diperbolehkan yang bisa tetap on.
- Jaringan ini dapat dikategorikan sebagai berikut:
 - **Jaringan kompetitif berbobot tetap:** a. Maxnet b. Mexican Hat c. Hamming Net
 - **Jaringan Kompetitif dengan bobot adaptif** a. Jaringan Kohonen Self Organizing Map (SOM) b. Jaringan Linear Vector Quantization (LVQ) c. Jaringan Counter Propagation (CP)



Jaringan kompetisi bobot tetap

- Dalam jaringan kompetisi bobot tetap, nilai titik-titiknya berubah selama proses iterasi berlangsung hingga akhirnya diperoleh satu titik yang menjadi pemenang (nilai > 0). Selama proses iterasi, semua bobotnya tidak berubah meskipun bobot ini mungkin dipakai untuk mengubah nilai titiknya
- Model jaringan kompetisi cocok dipakai dalam pengenalan pola. Pola masukan yang diberikan akan diubah hingga diperoleh keputusan tentang pola target mana yang paling menyerupai dengan masukan tersebut



Model Jaringan Kompetisi

- MaxNet

MaxNet merupakan model yang menggunakan dasar kompetisi. Selama proses, bobot dalam Maxnet dibuat tetap. Tidak ada proses pelatihan. MaxNet dapat dipakai sebagai bagian dari model jaringan lain untuk memperoleh neuron dengan masukan terbesar.

- Jaringan Hamming

Misalkan kita memiliki n buah vektor contoh. Jaringan Hamming dipakai untuk menentukan vektor contoh mana yang paling mirip dengan masukan yang diberikan. Vektor contoh akan menentukan bobot jaringan.

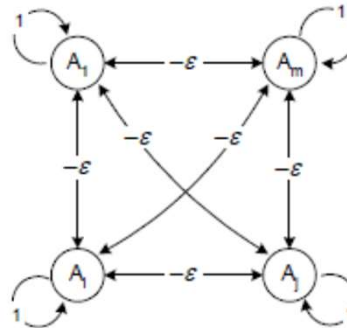
- Topi Meksiko (Mexican Hat)

Sub jaringan penguat lain yang ditemukan oleh Kohonen adalah jaringan topi meksiko (mexican hat). Nama ini diambil karena jika digambarkan, hasil iterasinya akan meruncing di tengah seperti sebuah topi orang meksiko.



MaxNet

- Arsitektur MaxNet seperti tampak pada gambar. Jaringan terdiri dari m titik yang semuanya saling berhubungan dengan bobot simetris. Bobot antar titik berbeda adalah $-\varepsilon$. Bobot suatu titik ke dirinya sendiri = 1.
- Bobot ini bernilai tetap selama prosesnya. Keluaran dari Maxnet adalah titik yang memiliki masukan terbesar.



- Fungsi aktivasi yang dipakai dalam MaxNet adalah :

$$f(x) = \begin{cases} x & \text{jika } x > 0 \\ 0 & \text{jika } x \leq 0 \end{cases}$$



MaxNet

Algoritma pemrosesan jaringan yang terdiri dari m unit input adalah sebagai berikut :

0. Inisialisasi ε dengan bilangan $0 < \varepsilon < \frac{1}{m}$

$$\text{Inisialisasi bobot } w_{ij} = w_{ji} = \begin{cases} 1 & \text{jika } i = j \\ -\varepsilon & \text{jika } i \neq j \end{cases}$$

1. Selama terdapat lebih dari 1 unit yang fungsi aktivasinya > 0 lakukan langkah 2

2. Modifikasi aktivasi titik a_j ($j = 1, 2, \dots, m$) dengan

$$a_j(\text{baru}) = f \left(a_j(\text{lama}) - \varepsilon \sum_{k \neq j} a_k(\text{lama}) \right)$$

Perhatikan bahwa dalam langkah (2), masukan fungsi aktivasi adalah jumlah semua masukan ke titik a_j).



Contoh :

Misalkan jaringan MaxNet seperti pada gambar memiliki bobot $\varepsilon = 0.2$ dengan nilai masukan mula-mula : $a_1 = 0.2$; $a_2 = 0.4$; $a_3 = 0.6$; $a_4 = 0.8$. Tentukan titik dengan masukan terbesar menggunakan iterasi MaxNet

Penyelesaian

Modifikasi nilai a_j dilakukan dengan aturan :

$$a_j(\text{baru}) = f\left(a_j(\text{lama}) - \varepsilon \sum_{k=j} a_k(\text{lama})\right)$$

Iterasi dilakukan terus hingga hanya satu a_j yang bernilai > 0



Iterasi 1

$$a_1(1) = f[0,2 - 0,2 \times (0,4 + 0,6 + 0,8)] = f[-0,16] = 0$$

$$a_2(1) = f[0,4 - 0,2 \times (0,2 + 0,6 + 0,8)] = f[0,08] = 0,08$$

$$a_3(1) = f[0,6 - 0,2 \times (0,2 + 0,4 + 0,8)] = f[0,32] = 0,32$$

$$a_4(1) = f[0,8 - 0,2 \times (0,2 + 0,4 + 0,6)] = f[0,56] = 0,56$$

} $K \neq j$

Iterasi 2

$$a_1(2) = f[0 - 0,2 \times (0,08 + 0,32 + 0,56)] = f[-0,192] = 0$$

$$a_2(2) = f[0,08 - 0,2 \times (0 + 0,32 + 0,56)] = f[-0,096] = 0$$

$$a_3(2) = f[0,32 - 0,2 \times (0 + 0,08 + 0,56)] = f[0,192] = 0,192$$

$$a_4(2) = f[0,56 - 0,2 \times (0 + 0,08 + 0,32)] = f[0,48] = 0,48$$



Iterasi 3

$$a_1(3) = f[0 - 0,2 \times (0 + 0,192 + 0,48)] = f[-0,1344] = 0$$

$$a_2(3) = f[0 - 0,2 \times (0 + 0,192 + 0,48)] = f[-0,1344] = 0$$

$$a_3(3) = f[0,192 - 0,2 \times (0 + 0 + 0,48)] = f[0,096] = 0,096$$

$$a_4(3) = f[0,48 - 0,2 \times (0 + 0 + 0,192)] = f[0,442] = 0,442$$

Iterasi 4

$$a_1(4) = f[0 - 0,2 \times (0 + 0,096 + 0,442)] = f[-0,1076] = 0$$

$$a_2(4) = f[0 - 0,2 \times (0 + 0,096 + 0,442)] = f[-0,1076] = 0$$

$$a_3(4) = f[0,096 - 0,2 \times (0 + 0 + 0,442)] = f[0,008] = 0,008$$

$$a_4(4) = f[0,442 - 0,2 \times (0 + 0 + 0,096)] = f[0,422] = 0,422$$

Iterasi 5

$$a_1(5) = f[0 - 0,2 \times (0 + 0,008 + 0,422)] = f[-0,086] = 0$$

$$a_2(5) = f[0 - 0,2 \times (0 + 0,008 + 0,422)] = f[-0,086] = 0$$

$$a_3(5) = f[0,008 - 0,2 \times (0 + 0 + 0,422)] = f[-0,0764] = 0$$

$$a_4(5) = f[0,422 - 0,2 \times (0 + 0 + 0,008)] = f[0,421] = 0,421$$



Hasil iterasi selengkapnya :

Iterasi	a_1	a_2	a_3	a_4
Mula-mula	0,2	0,4	0,6	0,8
1	0	0,08	0,32	0,56
2	0	0	0,192	0,48
3	0	0	0,096	0,442
4	0	0	0,008	0,442
5	0	0	0	0,421

Pada iterasi ke-5 hanya a_4 yang bernilai positif, maka iterasi dihentikan dengan masukan terbesar = a_4

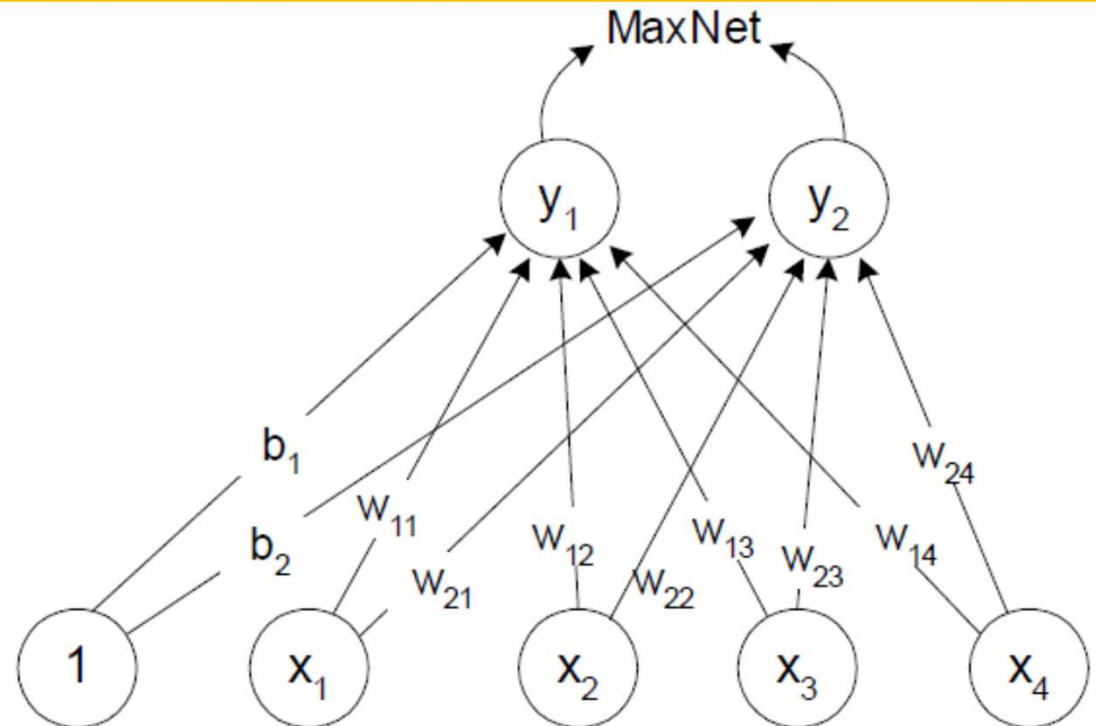


JARINGAN HAMMING

- Digunakan untuk menentukan vektor contoh mana yang paling mirip dengan masukan yang diberikan.
- Vektor contoh akan menentukan bobot jaringan
- Misal, x dan y adalah 2 buah vektor. Jarak Hamming antara x dan y didefinisikan sebagai banyaknya komponen berbeda diantara x dan y .
- Dalam prosesnya, jaringan Hamming menggunakan jarak Hamming sebagai ukuran kemiripan antara 2 buah vektor, dan MaxNet sebagai subnet untuk menentukan unit yang memiliki net masukan yang terbesar.



Arsitektur



Contoh arsitektur jaringan Hamming yang vektor masukannya terdiri dari 4 unit, dan memiliki 2 vektor contoh. Masing-masing node terhubung dengan garis koneksi yang memiliki bobot . Jaringan akan menentukan mana diantara vektor contoh yang paling mirip dengan vektor masukan yang diberikan.



- Misalkan : $e(1), e(2), \dots, e(m)$ adalah **m** buah vektor contoh, dengan masing-masing $e(j)$ terdiri dari n komponen. $e(j) = (e_1(j), e_2(j), \dots, e_n(j))$
- **n** adalah jumlah unit masukan
- Algoritma jaringan Hamming adalah sebagai berikut :

0. Inisialisasi bobot berdasarkan vektor contoh :

$$w_{ji} = \frac{e_i(j)}{2} \quad ; \quad i = 1, 2, \dots, n \quad ; \quad j = 1, 2, \dots, m$$

$$\text{Inisialisasi bias } b_j = \frac{n}{2} \quad ; \quad (j = 1, 2, \dots, m)$$

1. Untuk setiap vektor masukan x , lakukan langkah 2 - 4

$$2. \text{ Hitung } y_{\text{net}j} = b_j + \sum_i x_i w_{ji} \quad (j = 1, \dots, m)$$

3. Inisialisasi masukan MaxNet : $a_j(0) = y_{\text{net}j} \quad (j = 1, \dots, m)$

4. Gunakan prosedur MaxNet hingga diperoleh sebuah vektor yang bernilai positif (misal a_k). Maka vektor contoh yang paling baik adalah $e(k)$.



Contoh :

Diketahui 2 buah vektor contoh $e(1) = (1, -1, -1, -1)$ dan $e(2) = (-1, -1, -1, 1)$.

Gunakan jaringan Hamming untuk menentukan vektor contoh yang paling mirip dengan masing-masing dari 4 buah vektor berikut ini : $(1, 1, -1, -1)$, $(1, -1, -1, -1)$, $(-1, -1, -1, 1)$ dan $(-1, -1, 1, 1)$

Penyelesaian

Langkah pertama adalah menghitung bobot berdasarkan vektor contoh $e(1)$ dan $e(2)$.

$$w_{ji} = \frac{e_i(j)}{2} = \begin{pmatrix} 0.5 & -0.5 & -0.5 & -0.5 \\ -0.5 & -0.5 & -0.5 & 0.5 \end{pmatrix} \text{ dan } b_j = 4/2 = 2 \text{ (j = 1, 2)}$$

Berikutnya setiap vektor masukan dievaluasi kemiripannya dengan vektor pola contoh



Vektor $x = (1, 1, -1, -1)$:

$$y_{net_1} = 2 + 1 (0.5) + 1 (-0.5) - 1 (-0.5) - 1 (-0.5) = 3$$

$$y_{net_2} = 2 + 1 (-0.5) + 1 (-0.5) - 1 (-0.5) - 1 (0.5) = 1$$

Berikutnya digunakan jaringan Maxnet untuk menghitung unit yang menjadi pemenang. Misalkan diambil $\varepsilon = 0.2$

$$a_1(0) = y_{net_1} = 3 ; a_2(0) = y_{net_2} = 1.$$

Iterasinya menghasilkan :

$$a_1(1) = f(3 - 0.2 (1)) = f(2.8) = 2.8 ;$$

$$a_2(1) = f(1 - 0.2 (3)) = f(0.4) = 0.4$$

$$a_1(2) = f(2.8 - 0.2 (0.4)) = f(2.72) = 2.72 ;$$

$$a_2(2) = f(0.4 - 0.2 (2.8)) = f(-0.16) = 0$$



- Satu-satunya a_j yang bernilai positi adalah a_1 . Maka vektor contoh $e(1) = (1, -1, -1, -1)$ merupakan vektor yang paling cocok dengan masukan $x = (1, 1, -1, -1)$
- Perhitungan dengan jaringan MaxNet sebenarnya adalah mencari y_{net_j} yang memiliki nilai terbesar (dalam contoh ini adalah y_{net_1})
- Perhatikan bahwa perhitungan langkah (2) sama dengan menghitung jarak Hamming antara vektor masukan dengan vektor contoh.
- Jarak Hamming vektor masukan $x = (1, 1, -1, -1)$ dengan $e(1) = (1, -1, -1, -1)$ adalah 3 (**kedua vektor sama dalam 3 komponennya**, yaitu komponen ke-1, 3, dan 4).
- Jarak Hamming vektor masukan dengan $e(2) = (-1, -1, -1, 1)$ adalah 1 **karena hanya komponen ke-3 saja yang sama diantara keduanya**.



Vektor $x = (1, -1, -1, -1)$:

$$y_{net_1} = 2 + 1 (0.5) - 1 (-0.5) - 1 (-0.5) - 1 (-0.5) = 4$$

$$y_{net_2} = 2 + 1 (-0.5) - 1 (-0.5) - 1 (-0.5) - 1 (0.5) = 2$$

y_{net} terbesar adalah y_{net_1} , maka vektor $x = (1, -1, -1, -1)$ paling cocok dengan vektor contoh $e(1) = (1, -1, -1, -1)$

Bila digunakan jaringan Maxnet untuk menghitung unit yang menjadi pemenang. Misalkan diambil $\varepsilon = 0.2$

$$a_1(0) = y_{net_1} = 4 ; a_2(0) = y_{net_2} = 2.$$

Iterasinya menghasilkan :

$$a_1(1) = f(4 - 0.2 (2)) = f(3.6) = 3.6 ;$$

$$a_2(1) = f(2 - 0.2 (4)) = f(1.2) = 1.2$$

$$a_1(2) = f(3.6 - 0.2 (1.2)) = f(3.36) = 3.36 ;$$

$$a_2(2) = f(1.2 - 0.2 (3.6)) = f(0.48) = 0.48$$

$$a_1(3) = f(3.36 - 0.2 (0.48)) = f(3.264) = \mathbf{3.264}$$

$$a_2(3) = f(0.48 - 0.2 (3.36)) = f(-0.192) = \mathbf{0}$$



Vektor $x = (-1, -1, -1, 1)$:

$$y_{net_1} = 2 - 1(0.5) - 1(-0.5) - 1(-0.5) + 1(-0.5) = 2$$

$$y_{net_2} = 2 - 1(-0.5) - 1(-0.5) - 1(-0.5) + 1(0.5) = 4$$

y_{net} terbesar adalah y_{net_2} , maka vektor $x = (-1, -1, -1, 1)$ paling cocok dengan vektor contoh $e(2) = (-1, -1, -1, 1)$

Vektor $x = (-1, -1, 1, 1)$:

$$y_{net_1} = 2 - 1(0.5) - 1(-0.5) + 1(-0.5) + 1(-0.5) = 1$$

$$y_{net_2} = 2 - 1(-0.5) - 1(-0.5) + 1(-0.5) + 1(0.5) = 3$$

y_{net} terbesar adalah y_{net_2} , maka vektor $x = (-1, -1, 1, 1)$ paling cocok dengan vektor contoh $e(2) = (-1, -1, -1, 1)$

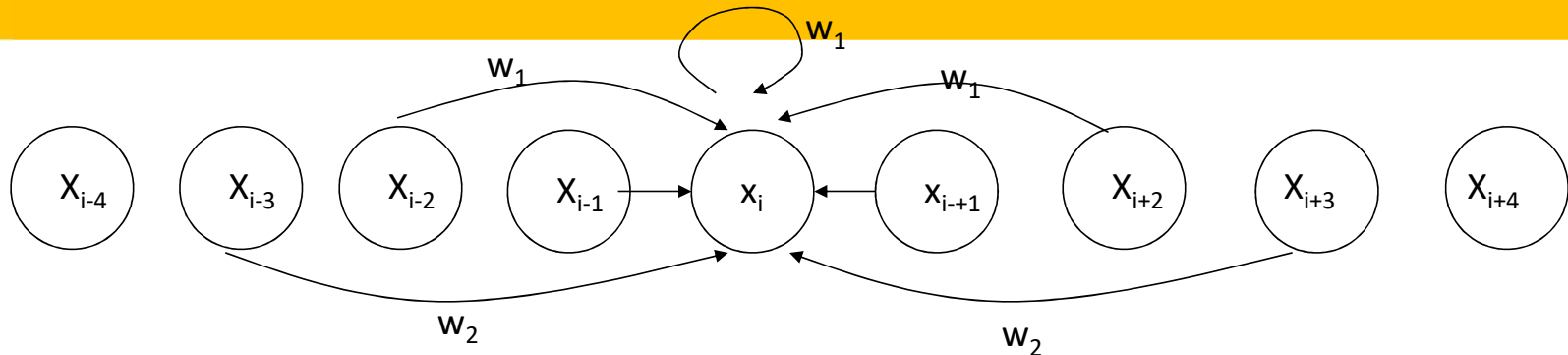


Soal-soal

- Diketahui 2 buah vector contoh: $e(1)=(1,-1,-1)$ dan $e(2)=(-1,-1,1)$.
Gunakan jaringan hamming untuk menentukan vector contoh yang paling mirip dengan vector berikut ini $X= (1,1,-1)$?



MEXICO HAT (topi meksiko)



- Ditemukan oleh Kohonen
- menggunakan 2 konstanta $R1$ dan $R2$ ($R1 < R2$) yang menunjukkan jari-jari titik yang memperkuat (bobot positif) dan memperlemah (bobot negatif) sinyal. Setiap neuron hanya dihubungkan dengan semua neuron berjarak $R1$ darinya (termasuk dengan dirinya sendiri) dengan bobot positif, dan dihubungkan juga dengan semua neuron berjarak R (dengan $R1 < R \leq R2$) darinya dengan bobot negatif.
- Gambar diatas menunjukkan arsitektur titik x_i dengan $R1 = 2$ dan $R2 = 3$. Titik x_i dihubungkan dengan semua titik yang jaraknya ≤ 2 (titik x_{i-2} , x_{i-1} , x_i , x_{i+1} , x_{i+2}) dengan bobot w_1 ($w_1 > 0$). Selain itu, x_i juga dihubungkan dengan titik yang jaraknya $= 3$ (titik x_{i-3} , x_{i+3}) dengan bobot w_2 ($w_2 < 0$).
- Dalam iterasinya, jaringan menyeleksi titik dg masukan maksimum beserta titik-titik disekitarnya
- Lama iterasi berhubungan dg jari-jari titik terkuat. Semakin lama iterasi, jumlah titik yg terpilih semakin sedikit.



Algoritma :

0. Inisialisasi R_1, R_2 dan t_{max} (jumlah iterasi maksimum)

$$\text{Inisialisasi bobot } w_k = \begin{cases} c_1 > 0 & \text{untuk } k = 0, \dots, R_1 \\ c_2 < 0 & \text{untuk } k = R_1 + 1, \dots, R_2 \end{cases}$$

Inisialisasi $x_i = 0$ ($i = 1, 2, \dots, n$)

$t = 0$

1. Selama $t < t_{max}$, lakukan langkah 2 - 5

2. Hitung net masukan x_i ($i = 1, 2, \dots, n$)

$$x_i = c_1 \sum_{k=-R_1}^{R_1} x_{i+k}(\text{lama}) + c_2 \sum_{k=-R_2}^{-R_1-1} x_{i+k}(\text{lama}) + c_2 \sum_{k=R_1+1}^{R_2} x_{i+k}(\text{lama})$$

3. $x_max = \max_i x_i$

4. Hitung fungsi aktivasi

$$x_i = \min(x_max, \max(0, x_i))$$

5. $t = t + 1$

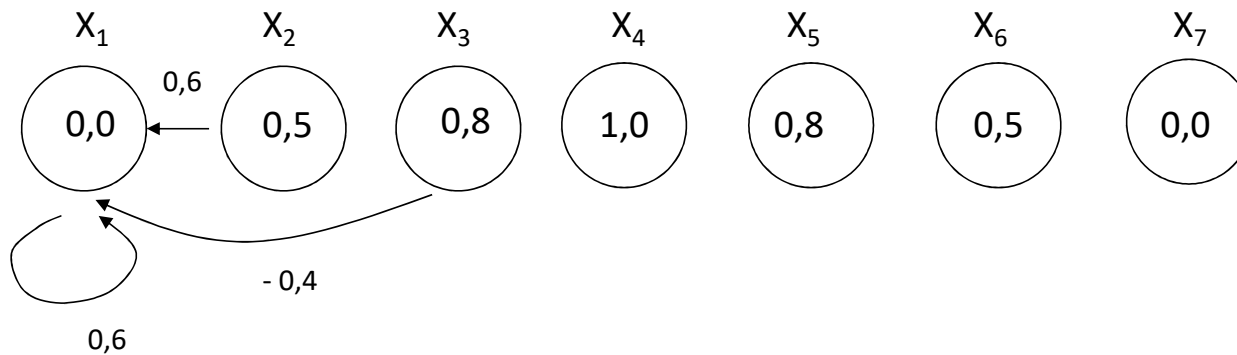


Contoh Soal

Gunakan algoritma topi meksiko pada vektor masukan x dg 7 unit : (0,0 0,5 0,8 1,0 0,8 0,5 0,0). Gunakan parameter $R1=1$, $R2=2$, $c1=0,6$ dan $c2= -0,4$

Penyelesaian

Modifikasi nilai x_i dalam iterasi pertama :





Iterasi berikutnya :

$$x1 = 0,6x1(lama) + 0,6 x2(lama) - 0,4 x3(lama)$$

$$x2 = 0,6x1(lama) + 0,6 x2(lama) + 0,6 x3(lama) - 0,4 x4(lama)$$

$$x3 = -0,4 x1(lama) + 0,6 x2(lama) + 0,6 x3(lama) + 0,6 x4(lama) - 0,4x5(lama)$$

$$x4 = -0,4 x2(lama) + 0,6 x3(lama) + 0,6 x4(lama) + 0,6 x5(lama) - 0,4x6(lama)$$

$$x5 = -0,4 x3(lama) + 0,6 x4(lama) + 0,6 x5(lama) + 0,6 x6(lama) - 0,4x7(lama)$$

$$x6 = -0,4 x4(lama) + 0,6 x5(lama) + 0,6 x6(lama) + 0,6 x7(lama)$$

$$x7 = -0,4 x5(lama) + 0,6 x6(lama) + 0,6 x7(lama)$$

Iterasi - 1 :

$$x1 = 0,6 (0,0) + 0,6 (0,5) - 0,4(0,8) = -0,2$$

$$x2 = 0,6 (0,0) + 0,6 (0,5) + 0,6 (0,8) - 0,4 (1,0) = 0,38$$

$$x3 = -0,4 (0,0) + 0,6 (0,5) + 0,6 (0,8) + 0,6 (1,0) - 0,4 (0,8) = 1,06$$

$$x4 = -0,4 (0,5) + 0,6 (0,8) + 0,6 (1,0) + 0,6 (0,8) - 0,4 (0,5) = 1,16$$

$$x5 = -0,4 (0,8) + 0,6 (1,0) + 0,6 (0,8) + 0,6 (0,5) - 0,4 (0,0) = 1,06$$

$$x6 = -0,4 (1,0) + 0,6 (0,8) + 0,6 (0,5) + 0,6 (0,0) = 0,38$$

$$x7 = -0,4 (0,8) + 0,6 (0,5) + 0,6 (0,0) = -0,2$$



$x_{\max} = 1.16$

Fungsi aktivasi menghasilkan :

$$x_1 = \min(1.16, \max(0, -0.2)) = 0$$

$$x_2 = \min(1.16, \max(0, 0.38)) = 0.38$$

$$x_3 = \min(1.16, \max(0, 1.06)) = 1.06$$

$$x_4 = \min(1.16, \max(0, 1.16)) = 1.16$$

$$x_5 = \min(1.16, \max(0, 1.06)) = 1.06$$

$$x_6 = \min(1.16, \max(0, 0.38)) = 0.38$$

$$x_7 = \min(1.16, \max(0, -0.2)) = 0$$

Didapat $x = (0, 0.38, 1.06, 1.16, 1.06, 0.38, 0)$



Iterasi - 2 :

$$x1 = 0,6 (0,0) + 0,6 (0,38) - 0,4 (1,06) = -0,196$$

$$x2 = 0,6 (0,0) + 0,6 (0,38) + 0,6 (1,06) - 0,4 (1,16) = 0,39$$

$$x3 = -0,4 (0,0) + 0,6 (0,38) + 0,6 (1,06) + 0,6 (1,16) - 0,4 (1,06) = 1,14$$

$$x4 = -0,4 (0,38) + 0,6 (1,06) + 0,6 (1,16) + 0,6 (1,06) - 0,4 (0,38) = 1,66$$

$$x5 = -0,4 (1,06) + 0,6 (1,16) + 0,6 (1,06) + 0,6 (0,38) - 0,4 (0) = 1,14$$

$$x6 = -0,4 (1,16) + 0,6 (1,06) + 0,6 (0,38) + 0,6 (0) = 0,39$$

$$x7 = -0,4 (1,06) + 0,6 (0,38) + 0,6 (0) = -0,196$$

$x_{\text{max}} = 1.66$ Fungsi aktivasi menghasilkan :

$$x1 = \min(1.66, \max(0, -0.196)) = 0$$

$$x2 = \min(1.66, \max(0, 0.39)) = 0.39$$

$$x3 = \min(1.66, \max(0, 1.14)) = 1.14$$

$$x4 = \min(1.66, \max(0, 1.66)) = 1.66$$

$$x5 = \min(1.66, \max(0, 1.14)) = 1.14$$

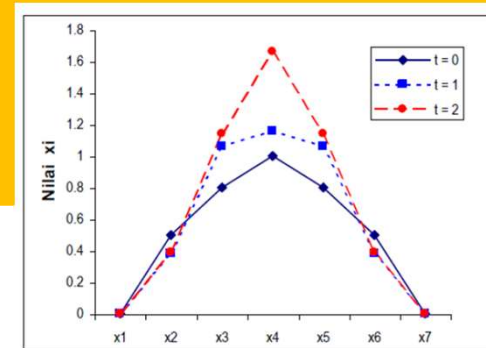
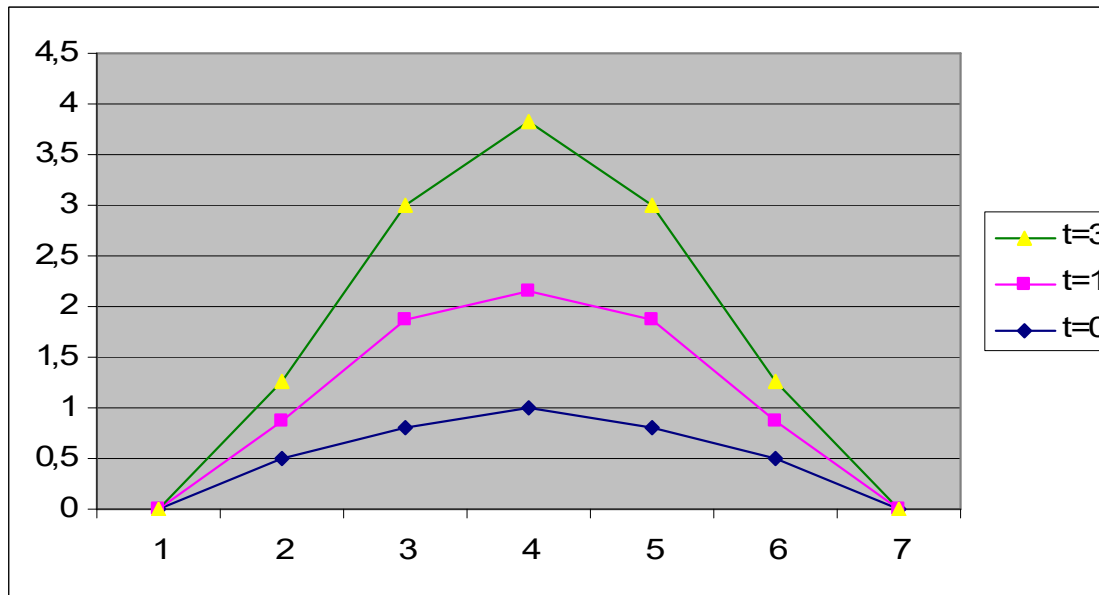
$$x6 = \min(1.66, \max(0, 0.39)) = 0.39$$

$$x7 = \min(1.66, \max(0, -0.196)) = 0$$

Didapat $x = (0, 0.39, 1.14, 1.66, 1.14, 0.39, 0)$



Iterasi bisa dilanjutkan.
Tampak pola yang terbentuk seperti topi
Yang teruncing adalah pemenangnya.





SOAL LATIHAN

1. Ulangi contoh pada algoritma topi meksiko dengan parameter $R_1=1$ dan $R_2=3$. Gambarkan grafiknya untuk $t=0$, $t=1$ dan $t=2$.

2. Gunakan algoritma topi meksiko pada vektor :

$$x = (1.4, 1.2, 0.7, 1.1, 2.3, 3.1, 2.4, 2.6)$$

Dengan parameter $R_1=2$, $R_2=3$, $c_1=0.5$, $c_2=-0.5$

3. Diketahui 4 buah vektor contoh pola

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

Gunakan jaringan Hamming untuk menentukan vektor contoh yang paling mirip dengan vektor masukan berikut :

a. # # #
 # . #
 # # #

b. # # .
 . # .
 . # #



Soal:

1. Diketahui 2 buah vector contoh $e(1)=(1,-1,-1)$ dan $e(2)=(-1,-1,1)$. Gunakan jaringan hamming untuk menentukan vector contoh yang paling mirip dengan vector berikut ini $X= (1,1,-1)$?