

فهرست گزارش سوالات

2.....	SOM – 1 سوال
5.....	MaxNet – ۲ سوال
8.....	Mexican Hat – 3 سوال
12.....	Hamming Net – 4 سوال

سوال 1 – SOM

** ماتریس W اولیه برای تمامی قسمت ها بصورت رندوم بین صفر و یک مشخص شده است. در ادامه نتایج حاصل بر روی داده ی تست آورده می شود.

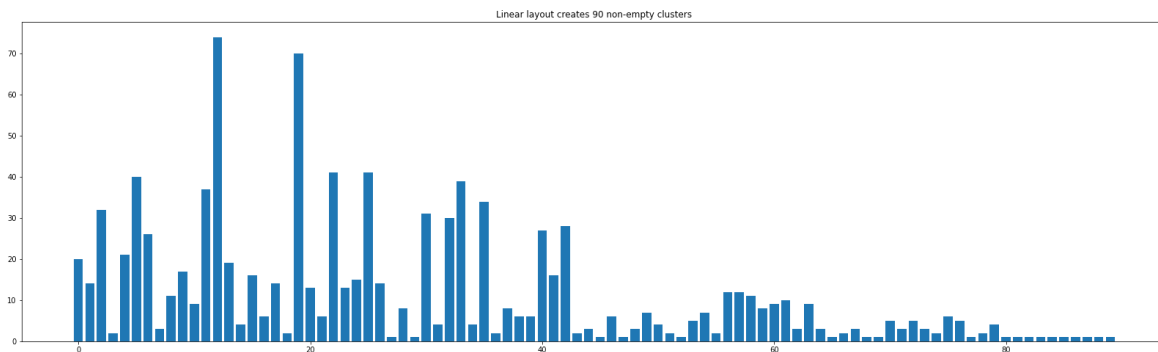
الف (در این بخش آرایش نورون ها خطی فرض شده و $R=0$ در نظر گرفتیم :

Max_epoch=20

Alpha0=0.01

بعد از اجرای ترینینگ و بدست آوردن ماتریس W نهایی ، نتیجه ی ارزیابی آن را بر روی داده ی تست مشخص می کنیم:

در این حالت ۹۰ نورون حداقل دارای یک عضو می باشند.



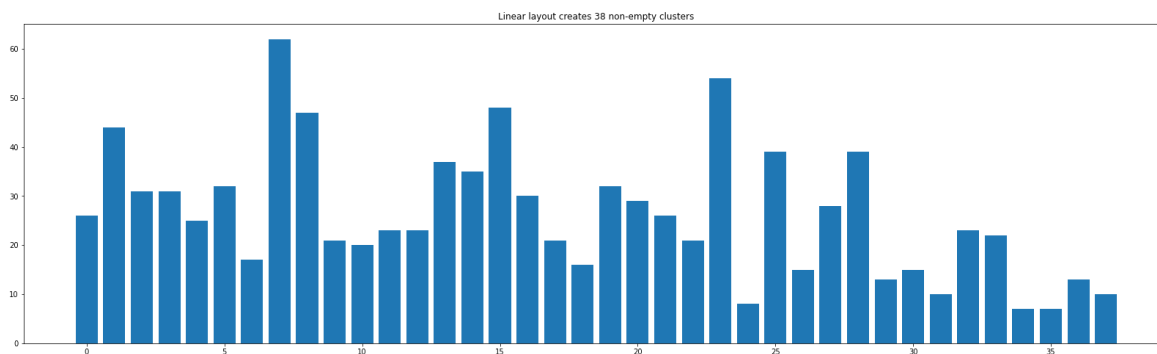
شکل ۱-۱) تعداد قرار گرفته در هر نورون برنده برای آرایش خطی با $R=0$

با بررسی این نمودار مشخص می شود که ۳۰ تا ازین نورون ها بیش از ۱۰ عضو دارند که می توان آن ها را به عنوان مرکز کلاستر ها برگزید (با تغییر این ترشهولد می توان تعداد کلاستر ها را به حد دلخواه رساند).

ب (در این بخش آرایش نورون ها خطی فرض شده و $R=2$ در نظر گرفتیم) یعنی در هر مرحله هر نود و نودهای چپ و راست آن در صورت وجود آپدیت وزنی می شوند).

نتیجه ی آن بر روی داده ی تست به شرح زیر می باشد.

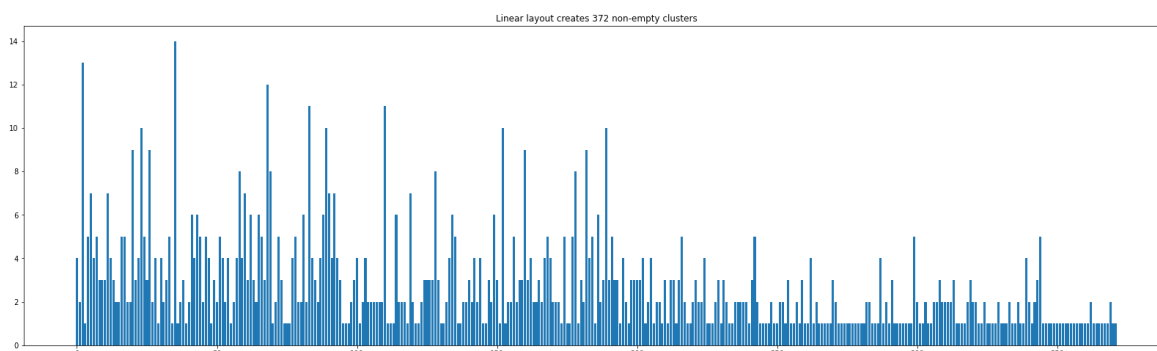
در این حالت ۳۸ نورون دارای عضو می باشند که نمودار پخش آن برای نود های برنده در شکل ۲-۱ آورده شده است.



شکل ۲-۱) تعداد قرار گرفته در هر نورون برنده برای $R=2$

شکل ۲-۱) نشان می دهد که در این حالت نسبت به حالت قبلی تعداد بیشتری نورون دارای اعضای بیشتر از ده بوده و تمرکز اعضا در این دسته ها بیشتر شده است.

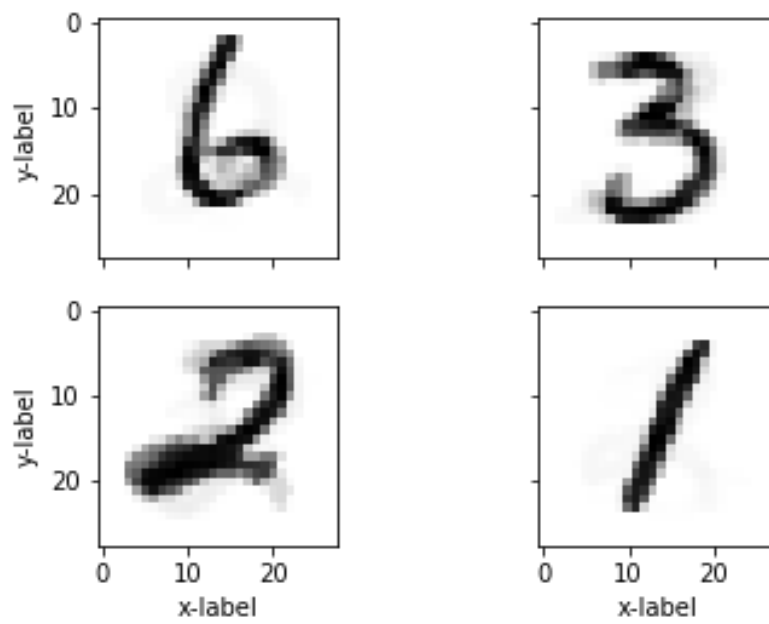
ج) در این حالت آرایش نورون ها را به صورت مربعی با $R=1$ در نظر می گیریم به این صورت که هر نورون یک مربع ۳ در ۳ همسایه داشته باشد (در مرزها این همسایگی کمی متفاوت خواهد بود) نتیجه حاصل از W بدست آمده از این شبکه بر روی داده ی تست بصورت زیر هست :



شکل ۳-۱) تعداد قرار گرفته در هر نورون برنده در آرایش مربعی با $R=1$

در این حالت انتظار داریم مطابق تئوری و توضیحات داده شده در رابطه با کاهش فاصله اقلیدسی در قسمت ترین شبکه وزن W به نوعی در نقاط بر تراکم مرکز این دسته ها را نشان دهد (پترن مرتبط با آن را استخراج می نماید).

در شکل ۴-۱) وزن مرتبط با چهار عدد از نورون هایی که برنده به حساب آمده و تعداد زیادی از داده ی تست را به خود اختصاص می دهند آورده شده است.



شکل ۴-۱ (چهار عدد مرکز دسته از داده های پرتکرار تر

شکل ۴-۱) کاملاً مطابق انتظار ما پترن مختص با چهار عدد $\{6, 3, 2, 1\}$ را مشخص کرده است.

سوال ۲ – MaxNet

در بخش اول این سوال بردار x را که یک بردار با اعضای تماماً مثبت هست به شبکه MaxNet می دهیم.

$X_arr = \begin{bmatrix} 1.20 & 1.10 & 0.50 & 1.50 & 1.13 & 0.80 \end{bmatrix}$

($\epsilon = 0.13$) ماتریس W بصورت زیر می باشد.

```
array([[ 1.  , -0.13, -0.13, -0.13, -0.13, -0.13],
       [-0.13,  1.  , -0.13, -0.13, -0.13, -0.13],
       [-0.13, -0.13,  1.  , -0.13, -0.13, -0.13],
       [-0.13, -0.13, -0.13,  1.  , -0.13, -0.13],
       [-0.13, -0.13, -0.13, -0.13,  1.  , -0.13],
       [-0.13, -0.13, -0.13, -0.13, -0.13,  1.  ]])
```

نتایج در مراحل مختلف را گزارش می کنیم.

1 st iteration:

New vector [0.55 0.43 0.00 0.89 0.47 0.09]

2 st iteration:

New vector [0.30 0.17 0.00 0.68 0.21 0.00]

3 st iteration:

New vector [0.16 0.02 0.00 0.60 0.06 0.00]

4 st iteration:

New vector [0.07 0.00 0.00 0.56 0.00 0.00]

5 st iteration:

New vector [0.00 0.00 0.00 0.55 0.00 0.00]

6 st iteration:

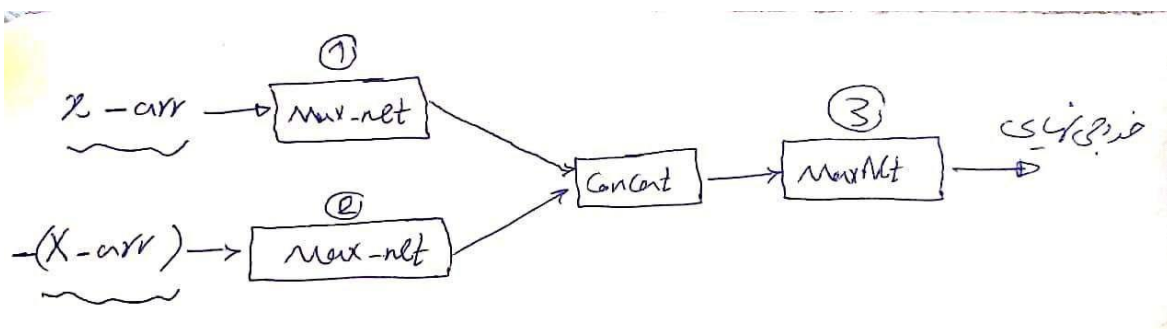
New vector [0.00 0.00 0.00 0.55 0.00 0.00]

با توجه به نتایج بدست آمده بعد از ۶ مرحله متوجه می شویم که ۴ امین درایه بزرگترین مقدار را داشته است. پس بزرگترین مقدار این آرایه 1.5 می باشد.

در بخش دوم بردار x ترکیبی از اعداد مثبت و منفی است.

$x_arr = [1.2, 1.1, 0.5, -1.5, 1.13, -0.8]$

برای اینکه بتوانیم الگوریتم Maxnet را بر روی آن اجرا نماییم به ترتیب زیر عمل می کنیم.
در این حالت در واقع از ۳ واحد MaxNet که آرایش آن در شکل ۱-۲ آورده شده است استفاده می شود. بدین ترتیب که ورودی Maxnet اول، بردار x و ورودی دومی گزینه ی بردار x می باشد. در نهایت خروجی این دو Maxnet با هم مقایسه می شود (توسط یک Maxnet دیگر).



Scanned with CamScanner

شکل ۱-۲ (آرایش به کار رفته در این سوال)

خروجی Maxnet اول :

```

1 st iteration:
New vector [1.14 1.03 0.35 0.00 1.06 0.00]
2 st iteration:
New vector [0.83 0.70 0.00 0.00 0.74 0.00]
3 st iteration:
New vector [0.64 0.49 0.00 0.00 0.54 0.00]
4 st iteration:
New vector [0.50 0.34 0.00 0.00 0.39 0.00]
5 st iteration:
New vector [0.41 0.23 0.00 0.00 0.28 0.00]
6 st iteration:
New vector [0.34 0.14 0.00 0.00 0.20 0.00]
7 st iteration:
New vector [0.30 0.07 0.00 0.00 0.14 0.00]
8 st iteration:
New vector [0.27 0.01 0.00 0.00 0.09 0.00]
9 st iteration:
New vector [0.26 0.00 0.00 0.00 0.05 0.00]
10 st iteration:
New vector [0.26 0.00 0.00 0.00 0.02 0.00]
11 st iteration:
New vector [0.25 0.00 0.00 0.00 0.00 0.00]

```

خروجی MaxNet دوم :

```

1 st iteration:
New vector [0.00 0.00 0.00 1.91 0.00 1.12]
2 st iteration:
New vector [0.00 0.00 0.00 1.76 0.00 0.87]
3 st iteration:
New vector [0.00 0.00 0.00 1.65 0.00 0.64]
4 st iteration:
New vector [0.00 0.00 0.00 1.57 0.00 0.42]
5 st iteration:
New vector [0.00 0.00 0.00 1.51 0.00 0.22]
6 st iteration:
New vector [0.00 0.00 0.00 1.48 0.00 0.02]
7 st iteration:
New vector [0.00 0.00 0.00 1.48 0.00 0.00]

```

در نهایت خروجی این دو Maxnet با هم ترکیب شده و به Maxnet آخری می روند.

Maximum absolute Value of x_arr: [1.2, 1.1, 0.5, -1.5, 1.13, -0.8] is = 1.5

سوال 3 – Mexican Hat

در این الگوریتم بررسی می کنیم که همکاری و رقابت نودها چگونه در شکل گیری جواب آن ها تاثیر دارد به این صورت که R1 و R2 شعاع همکاری و رقابت را نشان می دهند. در ادامه نتایج را برای ۳ تکرار از این شبکه می آوریم. هم چنین مقادیر $C1=0.2$ و $C2=-0.1$ در نظر گرفته شده است.

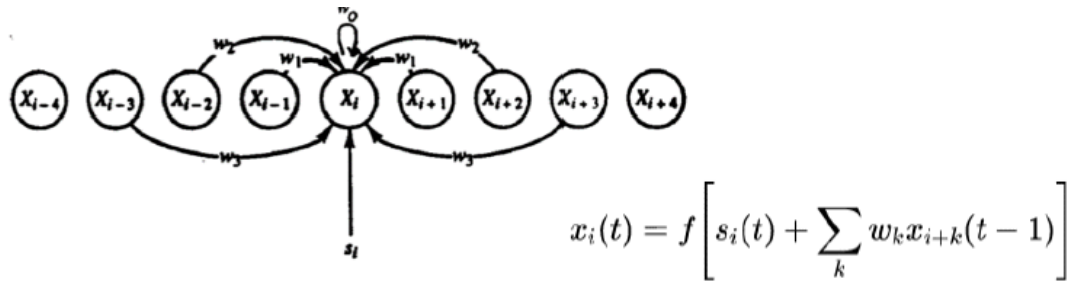


Figure 4.2 Mexican Hat interconnections for unit X_i .

شکل ۳-۱) ساختار شبکه Mexican Hat

الگوریتم در ادامه آورده شده است.

Algorithm

The algorithm given here is similar to that presented by Kohonen [1989a]. The nomenclature we use is as follows:

R_2	Radius of region of interconnections; X_i is connected to units X_{i+k} and X_{i-k} for $k = 1, \dots, R_2$.
R_1	Radius of region with positive reinforcement; $R_1 < R_2$.
w_k	Weight on interconnections between X_i and units X_{i+k} and X_{i-k} : w_k is positive for $0 \leq k \leq R_1$, w_k is negative for $R_1 < k \leq R_2$.
\mathbf{x}	Vector of activations.
$\mathbf{x_old}$	Vector of activations at previous time step.
t_max	Total number of iterations of contrast enhancement.
s	External signal.

Step 0. Initialize parameters t_max , R_1 , R_2 as desired.
Initialize weights:

$$w_k = C_1 \text{ for } k = 0, \dots, R_1 \text{ } (C_1 > 0)$$

$$w_k = C_2 \text{ for } k = R_1 + 1, \dots, R_2 \text{ } (C_2 < 0).$$

Initialize x_old to 0.

Step 1. Present external signal s :

$$x = s.$$

Save activations in array x_old (for $i = 1, \dots, n$):

$$x_old_i = x_i.$$

Set iteration counter: $t = 1$.

Step 2. While t is less than t_max , do Steps 3–7.

Step 3. Compute net input ($i = 1, \dots, n$):

$$x_i = C_1 \sum_{k=-R_1}^{R_1} x_old_{i+k} + C_2 \sum_{k=-R_2}^{-R_1-1} x_old_{i+k} + C_2 \sum_{k=R_1+1}^{R_2} x_old_{i+k}.$$

Step 4. Apply activation function (ramp function from 0 to x_max , slope 1):

$$x_i = \min(x_max, \max(0, x_i)) \text{ } (i = 1, \dots, n).$$

Step 5. Save current activations in x_old :

$$x_old_i = x_i \text{ } (i = 1, \dots, n).$$

Step 6. Increment iteration counter:

$$t = t + 1.$$

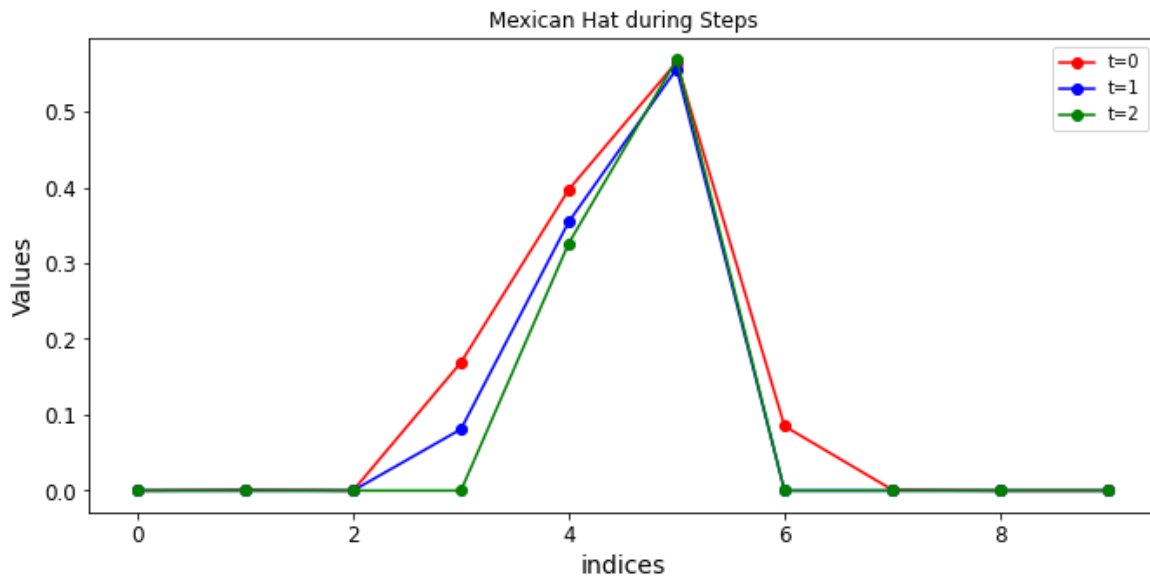
Step 7. Test stopping condition:

If $t < t_max$, continue; otherwise, stop.

با پیاده سازی الگوریتم فوق نتایج را برای دو حالت خواسته شده می آوریم :

حالت اول) در این قسمت $R1=0$ و $R2=\infty$ در نظر می گیریم :

مطابق تئوری در این حالت هر نود با تمامی نودهای دیگر رقابت می کند(چیزی شبیه به MaxNet ساده داریم).

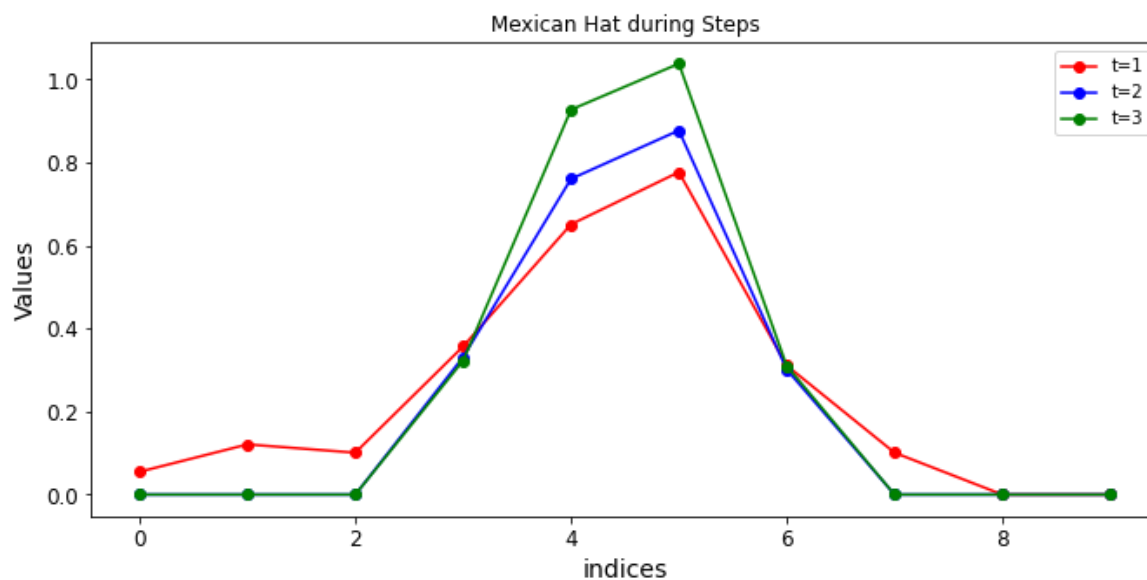


شکل ۲-۳) مقادیر هر نود بر حسب هر ایتريشن

نشان می دهد که بعد از سه تکرار چگونه مقادیر بیشتر در میانه ی بردار برنده شده اند.

حالت دوم) در این قسمت $R1=1$ و $R2=5$ در نظر می گیریم :

در این حالت هر نود با نودهای کناری خود همکاری کرده و با باقی نودها رقابت دارد در این حالت بعد از سه تکرار داریم.



شکل ۳-۳) مقادیر هر نود بر حسب هر ایتريشن

مقایسه ی نتیجه ی این حالت با حالت قبل به خوبی نشان می دهد که چگونه همکاری هر نود به نودهای کناری باعث شده که نودهای با اندیس ۳ و ۶ بعد از ۳ تکرار مقدار مثبتی داشته باشند.

سوال 4 – Hamming Net

الف) در شبکه Hamming Net برای دو بردار x و y با بعد n داریم :

$$x \cdot y = a - d \quad (1)$$

a = Number of equal bits

d = Number of not equal bits

با توجه به تعریف a و d مشخص است که :

$$d = n - a \quad (2)$$

در نتیجه از روابط ۱ و ۲ نتیجه می شود:

$$a = x \left(\frac{y}{2} \right) + \left(\frac{n}{2} \right)$$

پس در این شبکه وزنهای خروجی از هر $x(i)$ را بصورت $\left(\frac{y}{2} \right)$ قرار داده و بایاس را $b = \left(\frac{n}{2} \right)$ می گذاریم.

شکل ۱-۳ ساختار این شبکه را نشان می دهد با این تفاوت که در لایه ی میانی ۳ بردار پایه (y) خواهیم داشت و در لایه ی آخر نیز یک Maxnet قرار داده می شود

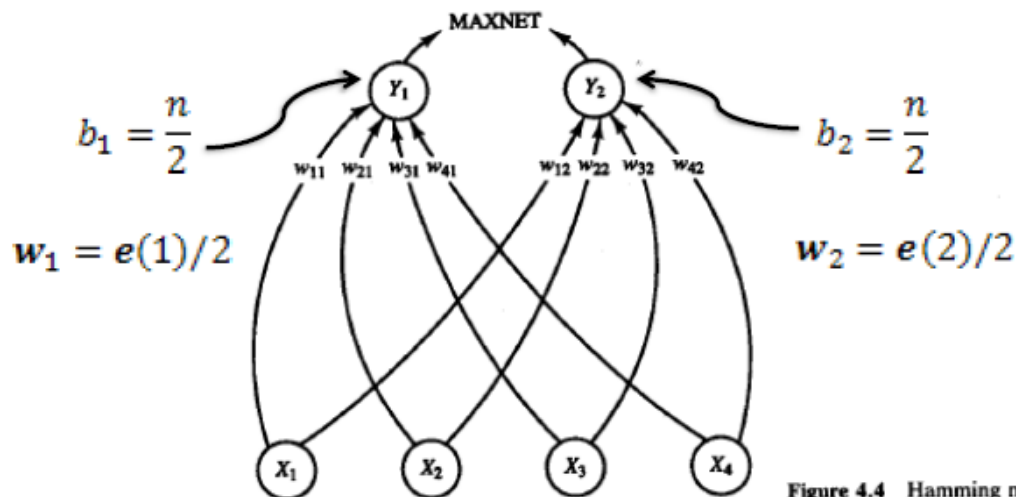


Figure 4.4 Hamming net.

شکل ۱-۴ ساختار یک شبکه Hamming Net با دو بردار پایه y

حال الگوریتم تست این شبکه به صورت زیر می باشد:

در هر کدام از نودهای میانی net حاصل از جمع نودهای متصل به آن (مطابق step 2) با وزن های مشخص شده محاسبه می شود.

The application procedure for the Hamming net is:

Step 1. For each vector \mathbf{x} , do Steps 2–4.

Step 2. Compute the net input to each unit Y_j :

$$y_in_j = b_j + \sum_i x_i w_{ij}, (j = 1, \dots, m).$$

Step 3. Initialize activations for MAXNET:

$$y_j(0) = y_in_j, (j = 1, \dots, m).$$

Step 4. MAXNET iterates to find the best match exemplar

به کمک الگوریتم فوق خروجی شبکه Maxnet خروجی را می دهد که بیشترین شباهت به یکی از بردار های پایه را دارد و این بردار را به عنوان خروجی شبکه نمایش می دهیم. هم چنین فاصله ی Hamming و Similarity distance نیز بدست می آید.

ب) در این قسمت با توجه به الگوریتم توضیح داده شده در قسمت الف ، ۵ بردار ورودی را به شبکه اعمال کرده و نتایج را توسط سه بردار پایه معرفی شده بررسی می نماییم.

1 st input result are:

Maximum absolute Value of Similarities: [3. 2. 3.] is = 3.0

The input is : [1. 1. 1. 1. 1. 1.]

the output e0 is the nearest vector: [1 -1 1 -1 1 -1]

2 st input result are:

Maximum absolute Value of Similarities: [2. 3. 2.] is = 3.0

The input is : [-1. 1. -1. -1. 1. 1.]

the output e1 is the nearest vector: [-1 1 -1 1 -1 -1]

3 st input result are:

Maximum absolute Value of Similarities: [3. 2. 1.] is = 3.0

The input is : [-1. -1. 1. 1. 1. 1.]

the output e0 is the nearest vector: [1 -1 1 -1 1 -1]

4 st input result are:

Maximum absolute Value of Similarities: [2. 3. 2.] is = 3.0

The input is : [-1. -1. 1. 1. -1. 1.]

the output e1 is the nearest vector: [-1 1 -1 1 -1 -1]

5 st input result are:

Maximum absolute Value of Similarities: [3. 4. 5.] is = 5.0

The input is : [-1. 1. 1. -1. -1. -1.]

the output e2 is the nearest vector: [1 1 1 -1 -1 -1]

در نتایج فوق ورودی و بردار خروجی مرتبط با آن بیان شده است. علاوه بر آن Similarities نشان می دهد که چقدر از درایه های هر بردار با بردار های پایه شباهت دارد. در نهایت اعمال Maxnet بیشترین شباهت و نزدیک ترین بردار را به ما می دهد.