



به نام خدا



دانشگاه تهران
دانشکده مهندسی برق و کامپیوتر
شبکه های عصبی و یادگیری عمیق

تمرین 4

نام و نام خانوادگی	مهسا مسعود
شماره دانشجویی	810196635
تاریخ ارسال گزارش	1400/4/ 18

فهرست گزارش سوالات

3.....	SOM – 1 سوال
8.....	MaxNet – ۲ سوال
10	Mexican Hat – 3 سوال
13	Hamming Net – 4 سوال

سوال 1 – SOM

در این سوال به طبقه بندی داده های MNIST با استفاده از الگوریتم SOM می پردازیم.
در این قسمت ابتدا به preprocess داده ها پرداخته و تمامی مقادیر را normalize می کنیم و با تقسیم مقادیر به 255 که بیشترین مقدار RGB است اینکار انجام میشود. سپس reshape انجام می دهیم و داده ها را به فرم دلخواه در می آوریم.

(الف)

ابتدا وزن ها را به صورت یونیفرم وزن دهی میکنیم.

```
W = []  
for i in range(784):  
    W.append(rand(625))
```

همچنین برای learning rate نیز مقدار اولیه 0.6 را قرار می دهیم و آن را در هر اپیاک منهای epoch/5 می کنیم.

اگر $R=0$ باشد :

Accuracy on train data is : 42.6 %

Accuracy on test data is : 38.7 %

مقادیر نورون های فعال :

```
6,2  
3,5,5,6  
7,4,4,8,8  
1,1,1,1,1,1,1,4,1,1,1,1,1,1  
0,2,6,0,0,0,2,0,6,0,2,7,6,0,3,2,6,0,8  
3,7,2,2  
2,5,3,0,3,5  
6  
4,4,4  
2,2,4,2,6,2,7,6  
9,6,1,9,6,4,7  
4  
7,9  
6  
4,4,8,9,9,7,5,9,9,4,4,9,3,4,4,6,5,9,7,9,8,4,9,4,4,4,4,4,7,6  
5  
6,2,3,3,3,5,3,5,3,2,3,1  
1,1,8  
4,4  
4,6,0,5,5,5,9,2,8,8,5,5,5,0,3,0,5,5,0,0,7,0,5,6,5,0,4,3,3  
7,4,2,7,8,7,2,2  
2,0,2,2  
7,3,9,6,2,8,4,3,7,2,9,2,1,3
```

0s completed at 10:05 PM

Activate Windows
Go to Settings to activate Windows.

5
4,4,9,4,4,4,4,8,9,9,4,4,5,4,9,4,4,7,7,4,4,9,5
5
3,0,0
1,1,1,1,1,1,1,1,1,1,1
1,1,8,8
3,2,0,2,2,0,2,5,3,5,0,3,2,3,0,3,0,3,3,0,3,2,2,3,0,3,8,2,8,0,3,2
7
1,1
3,5,3,3
8
5,0,5,6,1
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
4,6,7,4
6,9,6,4,4,3,7,4,7,1,4,1,7,9,9,7,9,5,7,9,8,7,8,8,2,9,6,4,4,8,8,9,2,4,1,9,9,1,1,9,4,4,9,7,8,9,4,9,7,9,7,2,2,7,2,6,7,9,6,0,5,7,8,4,8,9,9,4,9,9,6,6,8,8,8,1,9,7,9,2,3,9,
8,0,8,4
9,9,7,7,8
9,1,3,6,6,6,1,7,0,3,8,2,8,8,0,5,5,6,6,6,8,7,2,1,6,2,6,6,7,6,3,8
6
5,5,2,6,2,6,6
5,8,1,7,9,1,1,1,5,9,1
2,4,5,7
4,0

Activate Windows
Go to Settings to activate Windows.

نتایج خروجی نورون هایی که مقادیر کلاس ها را به خود جذب کرده اند ، به صورت بالا در آمد و
برای تعیین کردن کلاس مورد نظر در هر نورون، برا اساس اینکه تعداد کدام کلاس در هر لیست از نورون
فرضی، از بقیه بیشتر است، آن کلاس (ماکسیمم رخداد جذب) به عنوان کلاس مورد نظر نورون برگزیده
شده است.

(ب)

R=2

Accuracy on train data is : 37.25 %

Accuracy on train data is : 36.4 %

مقادیر نورون های فعال :

[6, 3, 6]
[2, 2]
[2]
[2, 2, 2, 2]
[0, 0, 0, 0]
[2, 3, 3, 2]
[2, 5, 8, 2, 8, 8, 2, 8, 2, 2, 2, 2, 7]
[1, 2, 1, 9, 1, 1, 2, 1, 2, 2, 2, 2, 2, 8, 2, 2, 1, 1, 8, 8, 2, 1, 2, 2, 1, 1, 1, 1, 6, 7, 1, 1, 8, 2, 2]
[2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2]
[2, 1, 2, 2, 2, 2]
[5, 5, 9, 5, 9, 5, 5, 3, 5, 8, 5, 8, 5, 9, 2, 7, 3, 5, 5, 5, 5, 9, 9, 6, 2, 2, 5, 5, 9, 3, 9, 6, 3, 5, 7, 5, 7, 7, 2, 2, 2, 5, 3, 7, 2, 8, 5, 7, 3, 5, 2, 5]
[2, 3, 5, 5, 8, 2, 3, 7, 8, 3, 3, 8, 7, 8, 5, 8, 3, 5, 5, 3, 8, 2, 3, 3, 5, 3, 5, 3, 2, 3, 3, 3, 3, 7, 3, 2, 5, 3, 3]
[3, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3]
[2, 2]
[4, 9, 4, 7, 4, 4, 4, 5, 4, 5, 0, 4, 9, 8, 4, 7, 7, 4, 6, 4, 9, 9, 4, 4, 7, 2, 9, 4, 5, 4, 4, 4, 4, 9, 4, 4, 4, 9, 8, 7, 8, 4, 9, 0, 9, 0, 0, 5, 9, 4, 6, 4, 8, 4, 4
[4, 4, 4, 9, 6, 6, 8, 3, 0, 2, 8, 8, 9, 4, 2, 2, 9, 8, 6, 7, 3, 2, 4, 4, 3, 8, 4, 9, 6, 2, 4, 0, 9, 8, 9, 4, 8, 5, 9, 4, 2, 4, 5, 2, 8, 2, 4, 4, 2, 2, 4, 4, 4, 6, 4, 4, 2
[0, 6, 0, 0, 3, 6, 7, 6, 0, 2, 0, 8, 0, 8, 2, 2, 9, 0, 2, 4, 0, 0, 8, 0, 2, 8, 0, 0, 9, 3, 2, 1, 0, 8, 0, 0, 3, 8, 2, 8, 2, 0, 9, 2, 0, 8, 8, 8, 4, 7, 0, 2, 8, 8, 7, 6
[9, 0, 7, 2, 5, 7, 8, 7, 7, 0, 2, 2, 4, 2, 2, 6, 4, 7, 5, 7, 2, 0, 7, 4, 7, 2, 0, 9, 8, 7, 0, 2, 2, 2]
[6]
[0, 4, 0, 0, 0, 4, 4, 0, 0]
[0, 4, 9, 9, 3, 6, 7, 5, 7, 9, 0, 6, 6, 6, 5, 9, 4, 4, 9, 7, 6, 7, 0, 6, 7, 6, 0, 5, 6, 0, 6, 9, 0, 9, 0, 3, 6, 9, 9, 5, 7, 8, 4, 9, 7, 3, 7, 6, 0, 0, 9, 9, 0, 0, 7, 6
[7, 7, 7, 6, 3, 4, 4, 7, 4, 7, 5, 6, 7]
[3, 3, 7, 3, 1, 6, 6, 3, 1, 3, 5, 7, 7, 9, 5, 3, 4, 5, 2, 3, 3, 7, 3, 3, 7, 3, 7, 5, 6, 0, 5, 7, 6, 4, 5, 6, 7, 9, 5, 8, 5, 3, 3, 8, 3, 5, 6, 9, 6, 1]
[2, 2, 7, 8, 2, 7, 8, 6, 8, 8, 8, 3, 8, 1, 7, 2, 3, 7, 7, 8, 3, 1]
[0, 3]
[0]
[5]
[1, 9, 5, 1, 1, 1, 3, 9, 1, 7, 3, 1, 1, 1, 8, 7, 3, 8, 1, 1, 1, 8, 1, 5, 1, 7, 1, 1, 1, 1, 5, 1, 1, 1, 6, 1, 5, 3, 6, 5, 3, 3, 1, 2, 1, 5, 8, 5, 1, 1, 1, 1, 1, 7, 5, 1, 8, 1
[1, 6, 1, 1, 1, 1, 3, 7, 1, 1, 1, 1, 1, 1]
[1, 8]
...

Activate Windows
Go to Settings to activate Windows.

```
[1, 1, 1, 3]
[1, 1, 3, 3, 1, 3, 3, 3, 6, 1, 1, 3, 3, 1, 3, 3, 3, 3, 3]
[7, 4, 9, 5, 9, 9, 4, 3, 3, 4]
[9, 7]
[7]
[4, 5, 4, 4, 4, 9, 4, 9, 9, 6, 9]
[4, 7, 9, 4, 6, 4, 9, 7]
[9, 3]
```

تعداد نورون های on که کلاسی را به خود جذب کرده اند به مراتب کاهش یافته است.

شعاع همگرایی بزرگتر شده است و عمل آپدیت کردن نورون ها به صورت خطی انجام می شود که 2 نورون سمت راست و چپ هر نورون با خودش (جمعا 5 تا) آپدیت میشوند که از آنجاییکه وزن های ابتدایی رندوم هستند ، در این مساله در بعضی مواقع به accuracy بالاتری در این حالت دست می یابیم و در بعضی موارد مثل مثال فعلی، accuracy تغییر زیادی نمی کند اما تعداد نورون ها به مراتب کاهش میابد که کار طبقه بندی را به شدت ساده تر می کند.

(ج)

R=1

در این حالت ، 4 نورون سمت راست، چپ، بالا و پایین هر نورون آپدیت می شود که جمعا 5 حالت می شود و توزیع متعادل تر و smooth تری نسبت به حالت قبلی برای آپدیت دارد.

Accuracy on train data is : 45.25 %

Accuracy on train data is : 34.8 %

این مورد باعث می شود که روی داده های آموزش درصد بالاتری بگیریم اما در داده های تست، احتمالا به دلیل اینکه در نهایت باز هم، در هر ایپاک، 5 تا نورون آپدیت می شوند، accuracy مثل حالت قبل است.

مقادیر نورون ها ی فعال:

```

[6, 6, 4, 4, 6]
[3]
[8, 7]
[5, 0, 2, 2, 0, 3, 3, 2, 3, 3, 6, 6, 3]
[2]
[0]
[5]
[5, 6, 8, 9, 9, 7, 4, 9, 6, 1, 7, 6, 3, 9, 2, 7, 4, 9, 2, 8, 4, 9, 9, 6, 9, 7, 3, 6, 9, 2, 6, 9, 4, 4, 6, 4, 7, 2, 2, 9, 6, 7, 2, 9]
[4, 8]
[5, 8, 1, 3, 1, 1, 3, 8, 5, 3]
[2]
[7, 8, 8, 4, 4, 2]
[6]
[6, 4, 0, 6, 0]
[1, 6, 2, 8, 6, 2]
[7, 8]
[9, 9, 4, 7, 4, 4, 2, 9, 2, 9, 4, 2, 4, 5, 2, 8, 4, 4, 4, 4, 8, 4, 8, 3, 2, 2, 4, 7, 4, 2, 8, 8, 9, 2, 4, 4, 4, 4, 2, 4, 2, 8, 8, 4, 2, 8]
[7, 7, 4]
[2, 7, 3, 7]
[5, 5, 2, 6, 6, 3, 3, 8]
[3, 4]
[3]
[9, 7, 9]
[7]

```

Activate \

```

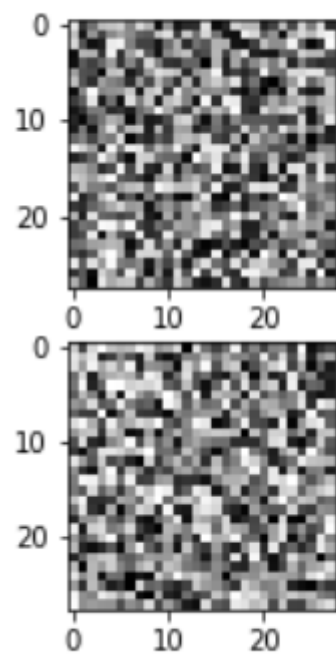
[1, 1, 4]
[2]
[3, 5, 8, 8]
[1]
[5]
[2, 3, 8, 8, 3, 3, 8, 0]
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
[7, 9, 9]
[4]
[2, 8, 2, 6, 0, 0]
[7, 7, 7, 2, 7, 7, 9, 5, 3, 8, 8, 7, 2, 7, 8, 6]
[4, 4]
[9]
[5, 3]
[4, 4, 4, 4, 5, 9, 8, 7, 8, 8, 4, 7, 7, 4, 7, 7, 8, 4, 7, 9, 9, 7, 7, 7, 9, 4, 8, 8, 9, 7, 5, 4]
[5, 2]
[3, 5, 3, 5, 6, 3, 5, 3, 6, 6, 3, 6, 3, 3, 3, 5, 5, 9, 3, 3, 4, 3, 6, 9, 5]
[7, 7]
[4]
[9, 9, 9, 9, 4, 9, 2, 8, 5]
[9, 4, 9, 4]
[9, 9, 4]
[6, 2, 6]
[3, 3, 1, 3, 3, 1, 5, 1, 3, 5, 1, 3, 4, 3, 3, 3, 3, 8, 3, 8, 3, 3, 3, 3, 3, 5]

```

در نهایت دو نمونه کلاس 6 و 7 در این حالت نشان داده شده اند که به دلیل دقت به نسبت پایین (به دلیل زمان بر بودن هر ایپاک و برای 35 ایپاک ترین شده که به مقدار بیشتری احتیاج داشتیم ، به سختی قابل تشخیص هستند.

همچنین اگر مقادیر وزن ها را در ابتدا صورت نرمال بین 0 و 1 مقدار دهی میکردیم، امکان accuracy بالاتر وجود داشت.

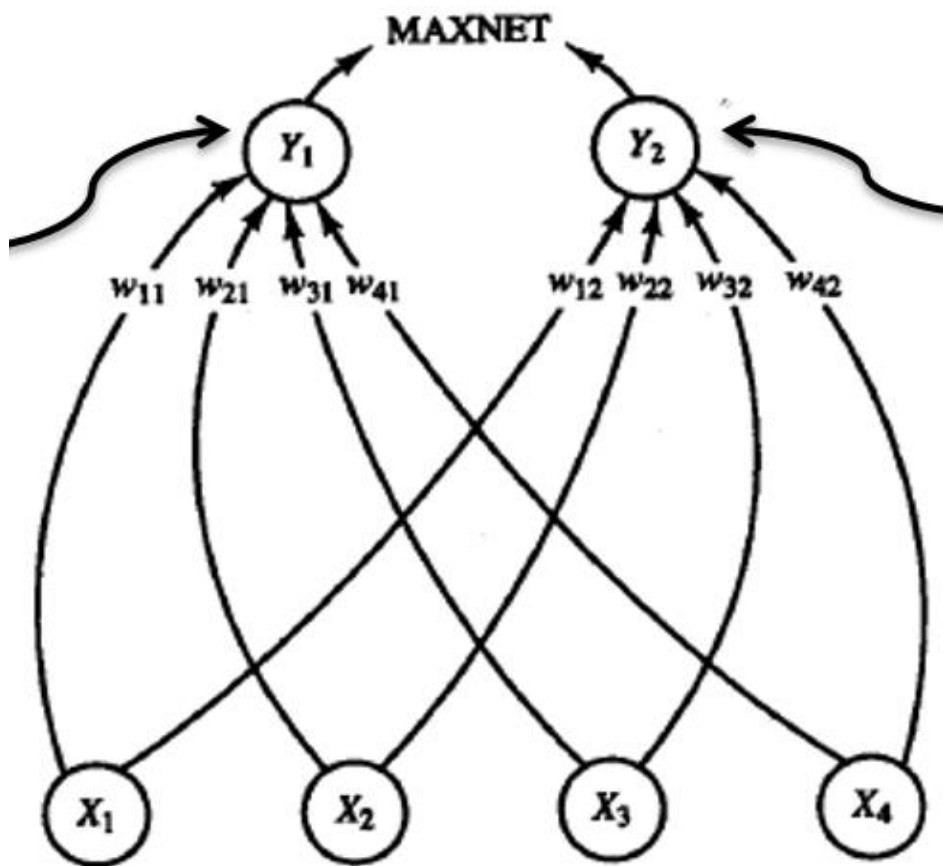
<matplotlib.image.AxesImage at 0x7f0e98d75790>



شکل 1-1- تصاویر خروجی کلاس 6 و 7

سوال ۲ – Max Net

در این سوال با پیاده سازی الگوریتم Max Net، مقدار بیشینه آرایه را به صورت زیر به دست می آوریم. این الگوریتم به این صورت است که در هر مرحله، خانه ای که مقدار صفر به خود میگیرد، کمترین مقدار را دارد. یعنی در اولین iteration آن خانه ای که صفر است، مینیمم آرایه است سپس دومین کمترین مقدار در دفعات بعدی صفر می شود و این روند تا انتها ادامه دارد تا فقط یک خانه غیر صفر باقی بماند که Maximum شبکه است.



شکل 1-2- MaxNet network

در این سوال اپسیلون برابر با 0.13 است.


```
input_arr = np.array([1.2,1.1,0.5,1.5,1.13,0.8])
epsilon = 0.13
```

خروجی :

Maximum value is in 3 th cell of array, with value of 1.5

برای به دست آوردن قدر مطلق هر بردار، می دانیم که یا بزرگترین عدد مثبت در این بردار، بیشترین مقدار را دارد یا کوچکترین عدد منفی. پس یک بار الگوریتم را روی خود بردار داده شده اجرا میکنیم و خروجی را ذخیره می کنیم و یک بار هم ابتدا همه اعداد بردار را در -1 ضرب کرده و بعد دوباره این الگوریتم را روی بردار قرینه شده اجرا کرده و خروجی را ذخیره می کنیم.

در نهایت بین این دو مقدار ذخیره شده max میگیریم:

```
inv_input_arr = -1 * np.array([1.2,1.1,0.5,1.5,1.13,0.8])
max_val_2 = MaxNet(inv_input_arr, 2)

print("Maximum absolute value is in ", str(max_val[1]), "th cell of array, with value of ", str(max(max_val_2[0], max_val[0])), "
```

Maximum absolute value is in 3 th cell of array, with value of 1.5 .

سوال 3 – Mexican Hat

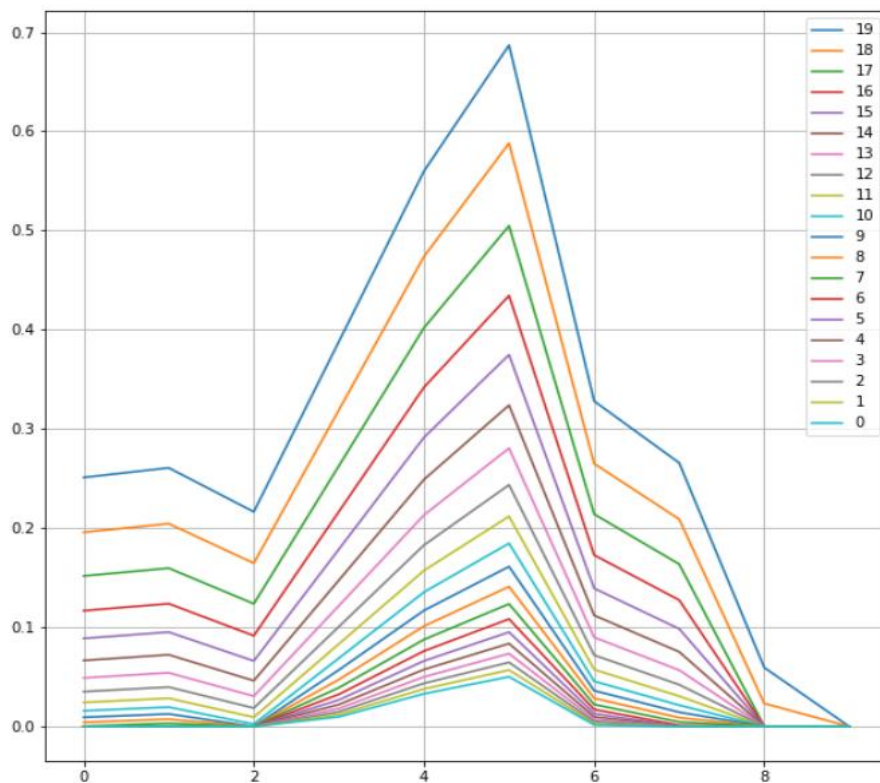
در این بخش با استفاده از این شبکه ها می خواهیم soft maximization را محاسبه کنیم. برای این کار باید دو مقدار R_1 و R_2 را ثابت قرار دهیم و با استفاده از رابطه زیر، مرحله update کردن را انجام دهیم.

$$x_i = c_1 \sum_{k=-R_1}^{R_1} x_{old_{i+k}} + c_2 \sum_{k=-R_2}^{-R_1-1} x_{old_{i+k}} + c_2 \sum_{k=R_1+1}^{R_2} x_{old_{i+k}}$$

برای دو حالت مختلف شعاع، این عملیات روی آرایه ورودی انجام شده است. نتایج با $tmax=20$ به صورت زیر است:

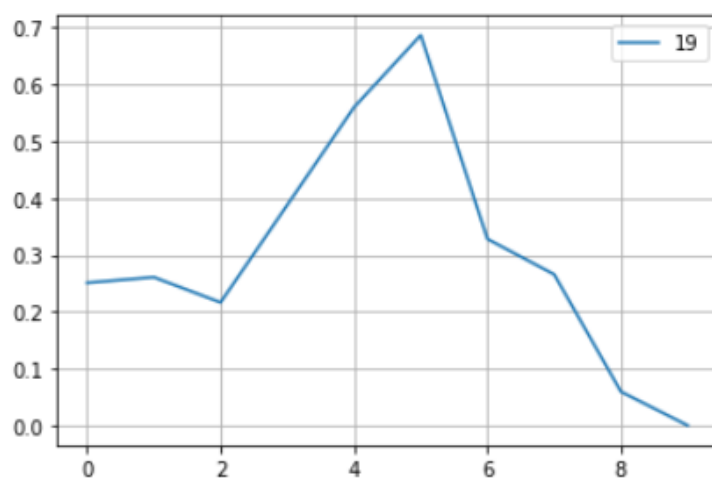
$$R_1 = 0, R_2 = \text{infinity}$$

در این قسمت با قرار دادن $c_2 = -0.01$ و $c_1 = 0.89$ ، به نتیجه زیر می رسیم. در این حالت از آن جایی که شعاع همکاری صفر و شعاع رقابت بی نهایت است، فقط node های مرکزی در نظر گرفته میشوند:



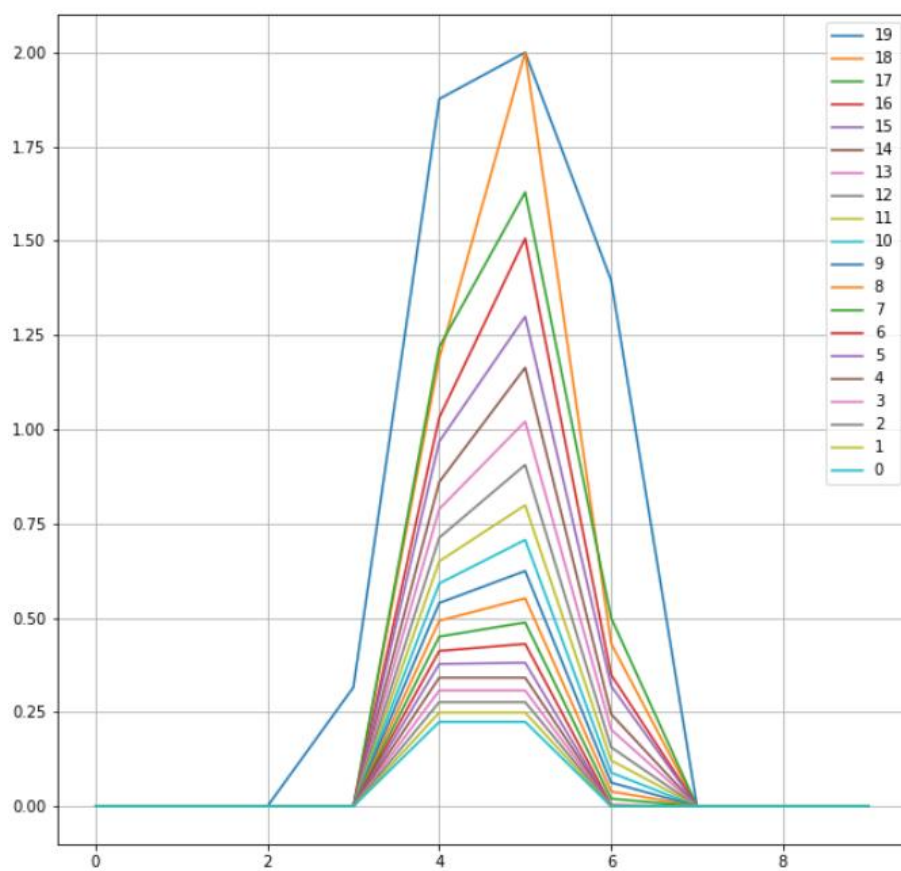
شکل 3-1 - نمایش خروجی mexican hat برای خصوصیات ذکر شده

سیگنال مرحله آخر :



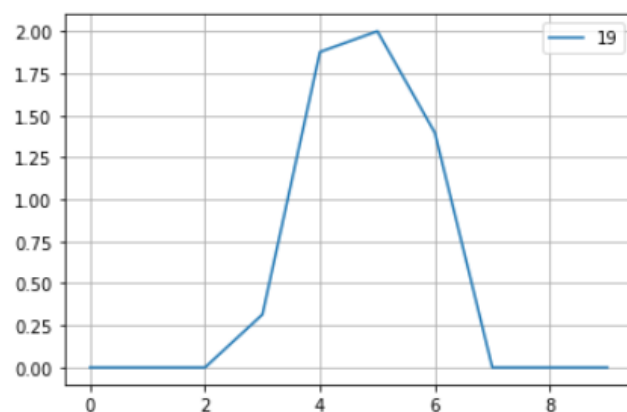
شکل 3-2- نمایش خروجی mexican hat برای آخرین iteration

در حالت دوم ، $R1 = 1$ و $R2 = 5$ و $C1 = 0.45$ و $C2 = -0.5$ است.



شکل 3-3 - نمایش خروجی mexican hat برای خصوصیات ذکر شده

سیگنال مرحله آخر:



شکل 3-4- نمایش خروجی mexican hat برای آخرین iteration

سوال 4 – Hamming Net

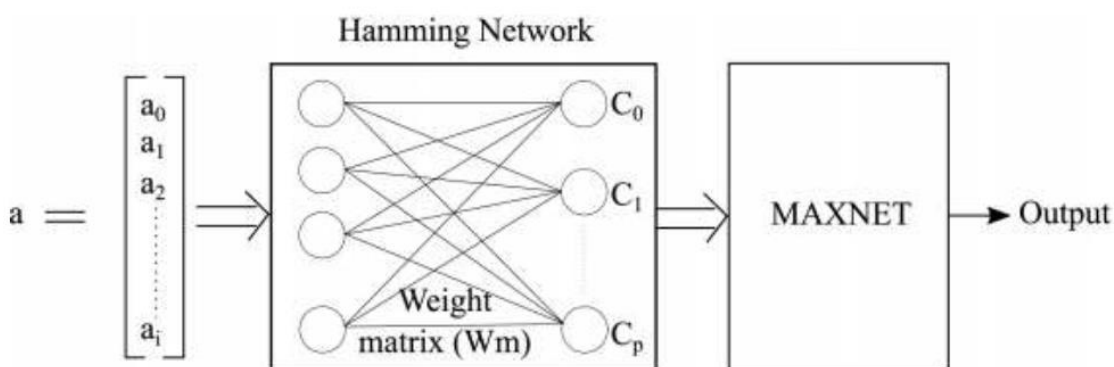
(الف)

در این شبکه، شباهت بین بردار های پایه (که ماتریس W را باهم تشکیل می دهند) با بردار های ورودی سنجیده می شود و شبیه ترین بردار پایه به بردار ورودی، خروجی این Network خواهد بود. برای این کار ابتدا باید ماتریس W (وزن ها) و b (بایاس) را تشکیل دهیم.

از رابطه زیر داریم:

$$b = \frac{n}{2}, W = \frac{1}{2} [e_1, e_2, e_3]$$

خروجی به دست آمده از ضرب بردار ورودی در W و جمع آن با بردار بایاس را با کمک گرفتن از یک MaxNet، به پاسخ نهایی تبدیل می کنیم. به عبارتی، بین شباهت بردار ورودی و تمامی بردار های پایه Max می گیریم و شبیه ترین را انتخاب می کنیم.



شکل 4-1- نمایش شماتیک Hamming Network

(ب)

با کمک گرفتن از روابط قسمت قبل ماتریس وزن ها را به شکل زیر تشکیل می دهیم:

$$\begin{bmatrix} 0.5 & -0.5 & 0.5 & -0.5 & 0.5 & -0.5 \\ -0.5 & 0.5 & -0.5 & 0.5 & -0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & -0.5 & -0.5 & -0.5 \end{bmatrix}$$

سپس با استفاده از MaxNet طراحی شده در سوال اول و ضرب ورودی در W و جمع حاصل با b ، خروجی به شکل زیر به دست می آید.

e1 is the closest vector to v1
value of similarity: 3.0

e2 is the closest vector to v2
value of similarity: 2.0

e1 is the closest vector to v3
value of similarity: 3.0

e2 is the closest vector to v4
value of similarity: 2.0

e3 is the closest vector to v5
value of similarity: 5.0

این موارد با دقت و شباهت سنجی چشمی نیز درست است.
