



به نام خدا



دانشگاه تهران  
دانشکده مهندسی برق و کامپیوتر  
درس شبکه‌های عصبی و یادگیری عمیق

تمرین سری ۴

سیاوش شمس	نام و نام خانوادگی
810197644	شماره دانشجویی
۰۱/۰۳/۲۵	تاریخ ارسال گزارش

## فهرست گزارش سوالات

سوال ۱ - SOM ..... ب

الف) ..... ب

ب) ..... ب

پ) ..... ج

ت) ..... خ

سوال ۲ - MaxNet ..... ۸

سوال ۳ - Mexican Hat ..... ۱۰

الف) ..... ۱۰

ب) ..... ۱۱

سوال ۴ - Hamming Net ..... ۱۲

الف) ..... ۱۲

ب) ..... ۱۲

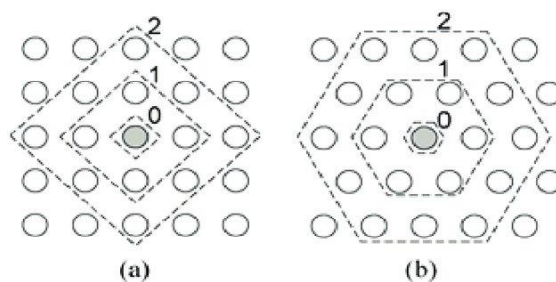
پ) ..... ۱۳

## SOM - ۱ سوال

(الف)

با توجه به معماری مربعی شبکه و ابعاد تصویر ماتریس وزن ها دارای ابعاد  $(15 \times 15 \times 784)$  می باشند.

نوع چینش نورون ها در مرحله آپدیت کردن وزن ها در همسایگی مشخص شده به کمک شعاع تاثیر گذار است. مثلا در شکل ۱-۱ دو نوع مختلف همسایگی با توجه به نحوه چینش نورون ها نشان داده شده است.



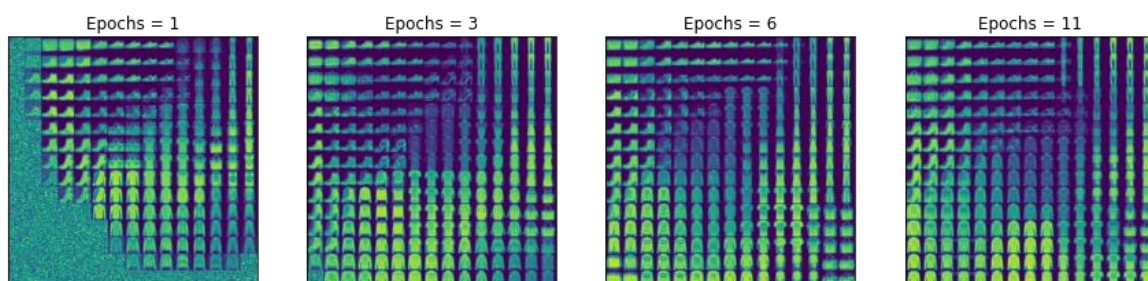
شکل ۱-۱- تفاوت همسایگی به دلیل نحوه چینش نورون ها

(ب)

بعد از آزمایش با مقادیر مختلف برای ضریب یادگیری و ضریب میرایی آن مقدار اولیه ضریب یادگیری را برابر 0.4 و مقدار ضریب میرایی را برابر 0.1 در نظر می گیریم و ضریب یادگیری را به صورت زیر آپدیت می کنیم

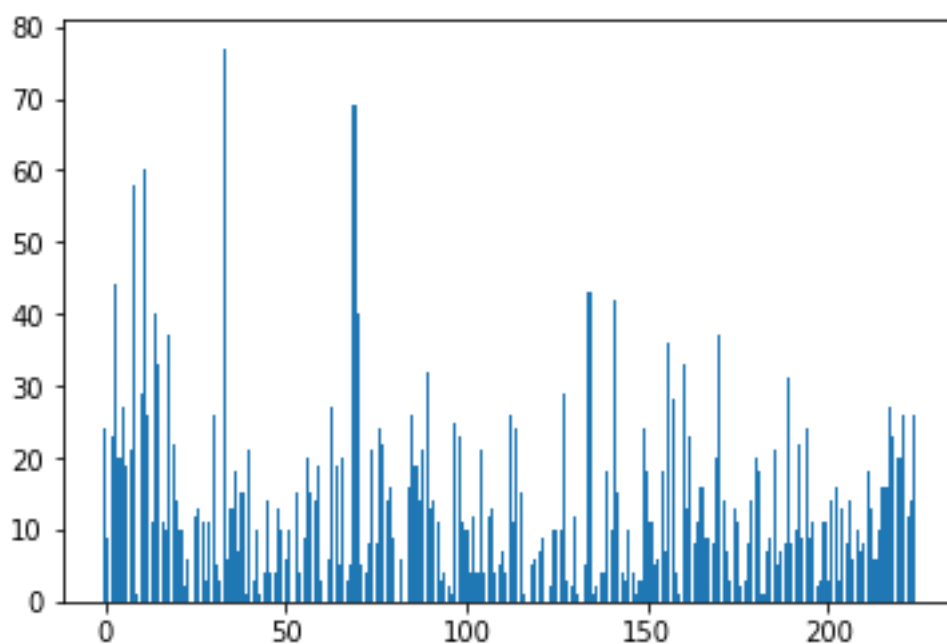
$$\alpha = \alpha \times e^{-decay \cdot epoch}$$

$$\alpha_0 = 0.4, \quad decay = 0.1$$

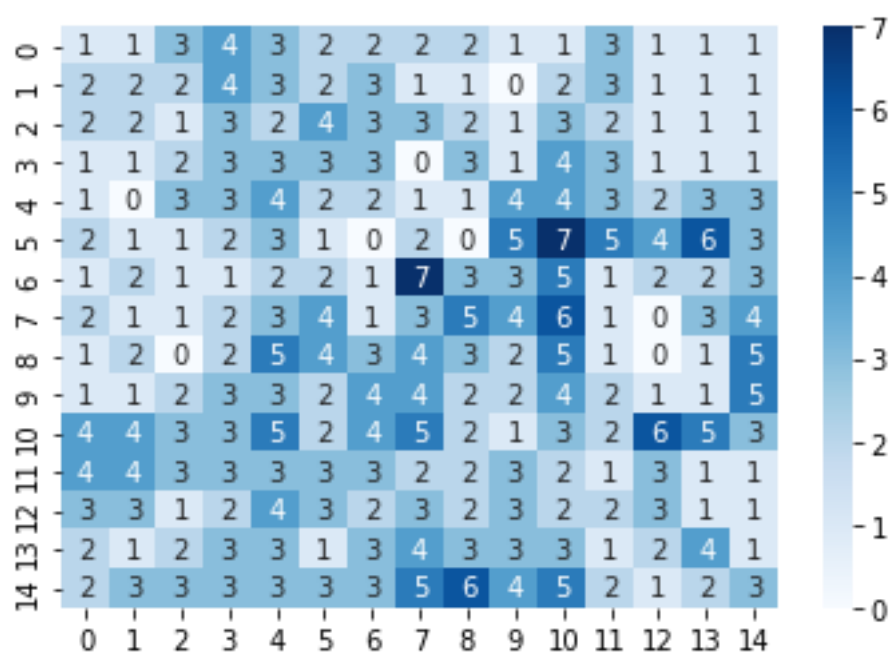


شکل ۱-۲- نتایج خروجی وزن ها به ازای ایپاک های مختلف

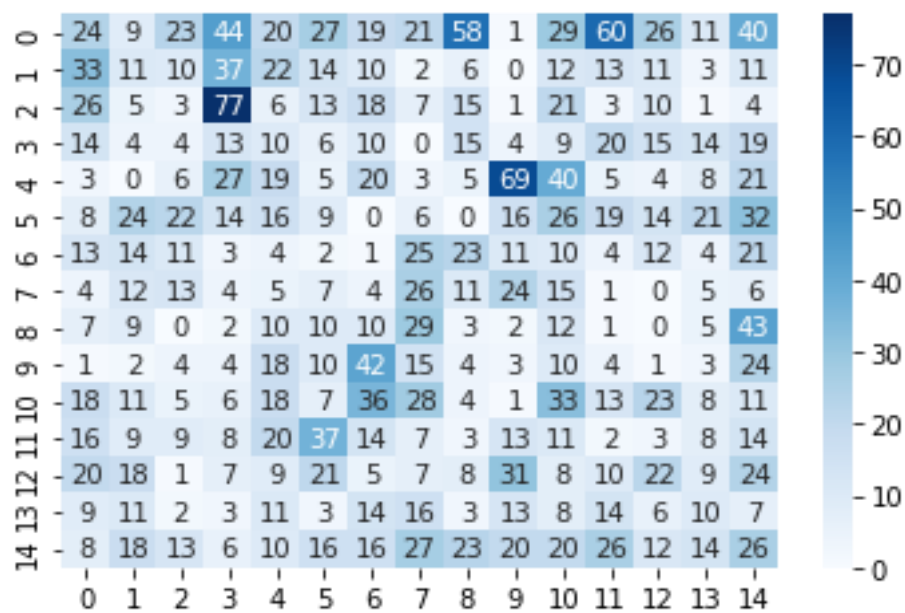
\* تصویر بزرگتر در فایل Q1.ipynb نشان داده شده است



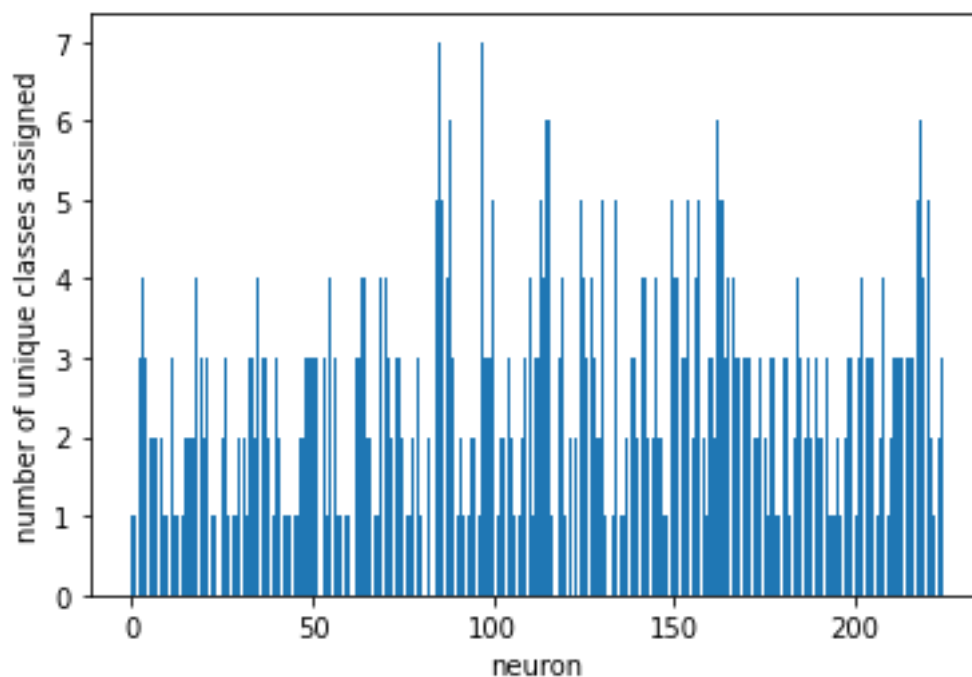
شکل ۳-۱- هیستوگرام تعداد ورودی های مپ شده به هر نورون



شکل ۴-۱- تعداد کلاسه های مختلف های مپ شده به هر نورون

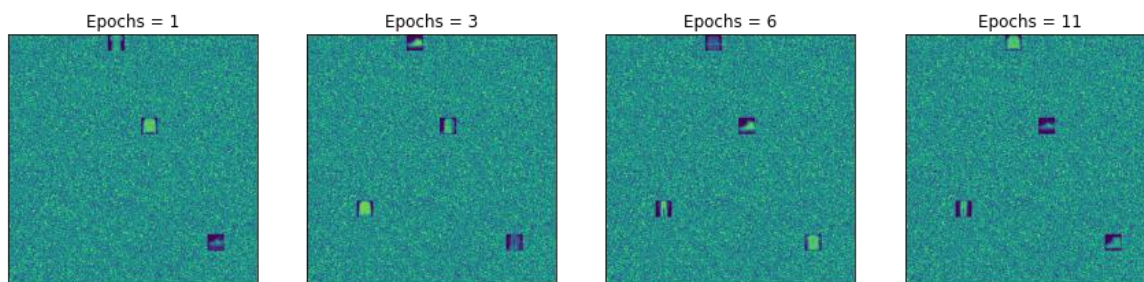


شکل ۵-۱- تعداد ورودی های مپ شده به هر نورون



شکل ۶-۱- هیستوگرام تعداد کلاس مختلف مپ شده به هر نورون

average classes assigned to each neuron: 2.3377777777777777



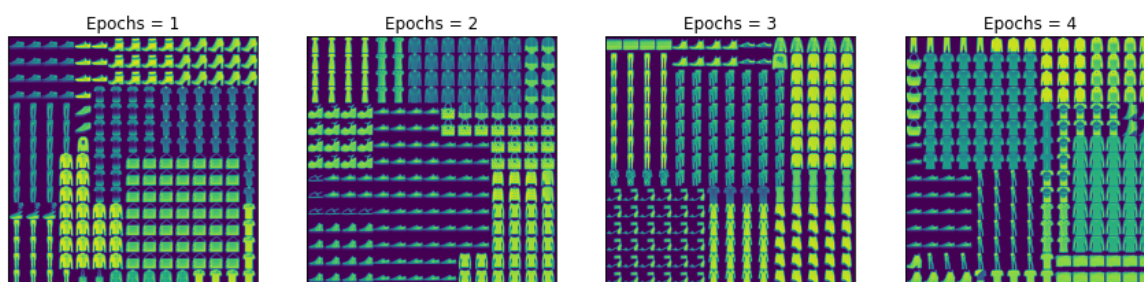
شکل ۷-۱- نتایج خروجی وزن ها به ازای ایپاک های مختلف هنگام  $R=0$

همانطور که می بینیم ماتریس وزن ها برای حالت  $R=0$  به خوبی آپدیت نشده است.

(پ)

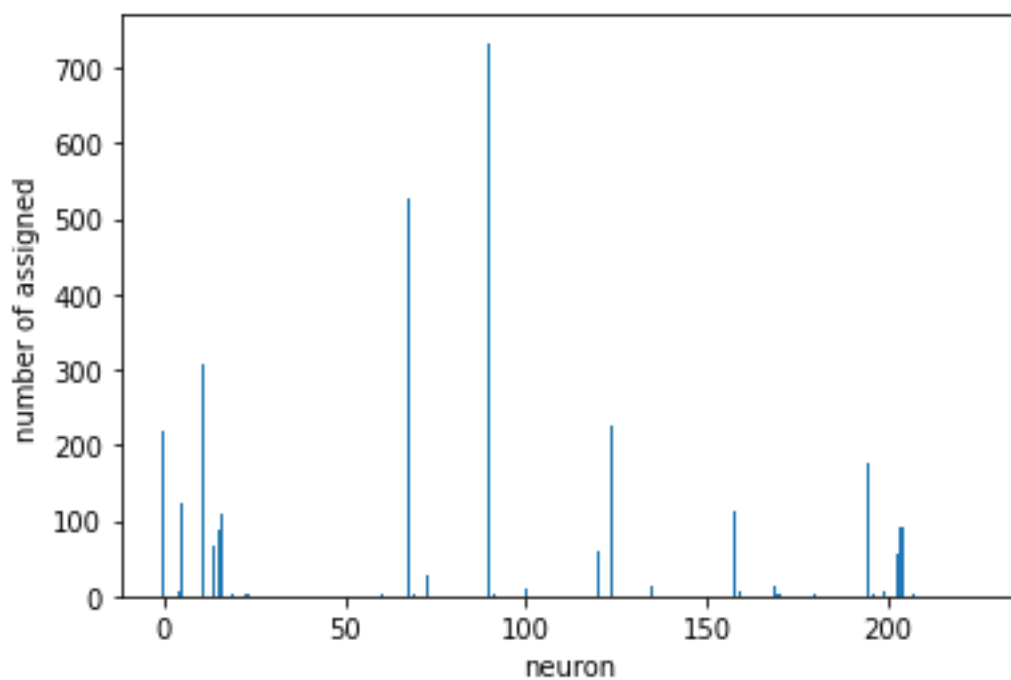
مقادیر پارامتر ها را همانند قسمت ب در نظر گرفتیم و تغییری ندادیم (initial learning rate=0.4,

decay=0.2)



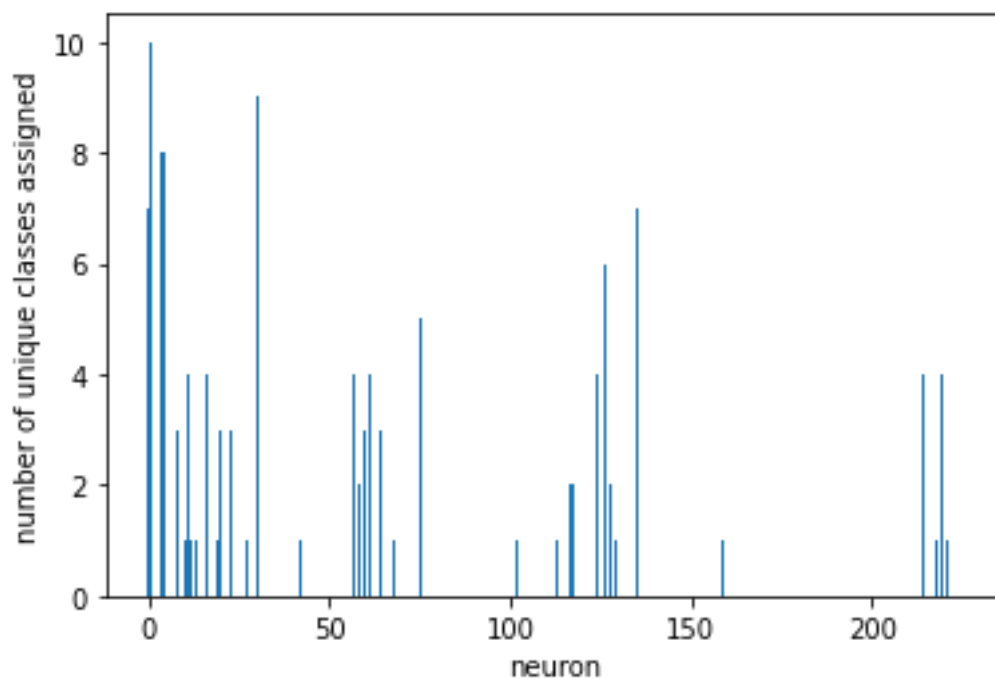
شکل ۸-۱- نتایج خروجی وزن ها به ازای ایپاک های مختلف

\* تصویر بزرگتر در فایل Q1.ipynb نشان داده شده است



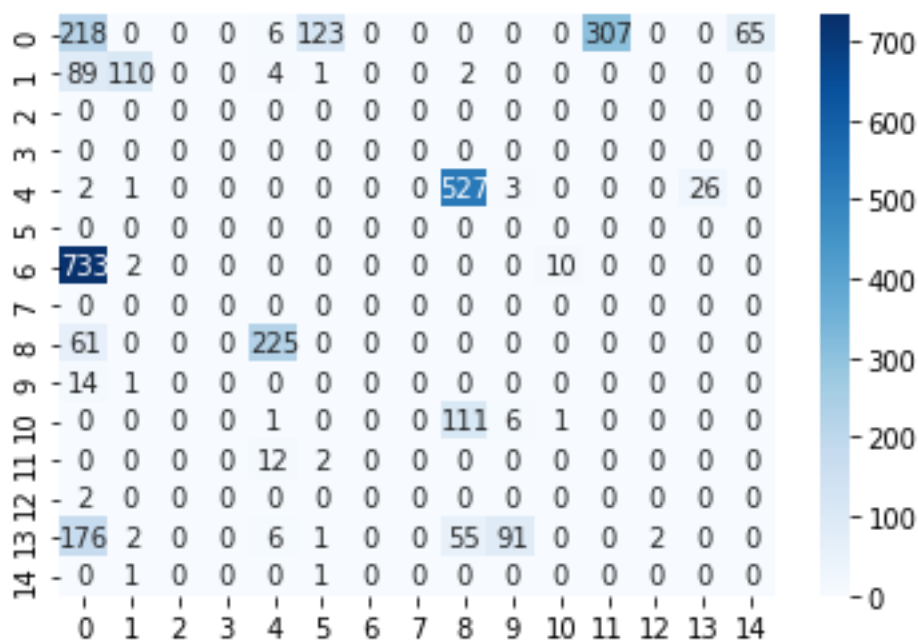
شکل ۹-۱- هیستوگرام تعداد ورودی های مپ شده به هر نورون

average classes assigned to each neuron: 0.5066666666666667



شکل ۱۰-۱- هیستوگرام تعداد کلاسهای مختلف مپ شده به هر نورون

به ۲۴ نورون بیشتر از ۱ کلاس مپ شده است



شکل ۱۱-۱- تعداد ورودی های مپ شده به هر نورون

#### ت)

همانطور که دیدیم در حالتی که شعاع در ابتدا ۳ باشد و هر ایپاک ۱ عدد کاهش پیدا کند در تعداد ایپاک کمتر به نتیجه بهتری نسبت به شعاع ثابت رسیدیم، همچنین با توجه به هیستوگرام های رسم شده در حالت استفاده از شعاع کاهشی به طور میانگین تعداد کلاس کمتری به نورون ها اختصاص داده می شود که نشان دهنده مطلوب بودن طبقه بندی می باشد. دلیل این بهبود این است که در ابتدا همسایگی بزرگی را آپدیت می کنیم تا کلاس طبقه بندی شوند سپس با کاهش شعاع بهترین نورون را برای نمایندگی آن کلاس انتخاب می کنیم پس در این حالت کلاس ها بیشتر به مرکز دسته های مربوط به خودشان نگاشت می شوند.



## سوال ۲ – MaxNet

پس از اعمال الگوریتم MaxNet روی بردار داده شده نتایج به روزرسانی به صورت شکل ۲-۱ خواهد

بود

بیشترین عدد بردار بعد از ۹ ایپاک به دست می آید.

```
epoch 1 : [1.2, 1.1, 1, 0.9, 0.95, 1.15]
epoch 2 : [0.43499999999999994, 0.31999999999999995, 0.20499999999999996, 0.08999999999999997, 0.14749999999999985, 0.37749999999999984]
epoch 3 : [0.264, 0.13174999999999998, -0.0004999999999999727, -0.13275, -0.06662500000000001, 0.19787499999999998]
epoch 4 : [0.21455625000000003, 0.06246875, -0.08904374999999998, -0.08904374999999998, -0.08904374999999998, 0.13851249999999999]
epoch 5 : [0.18440906250000005, 0.009508437500000022, -0.06233062499999998, -0.06233062499999998, -0.06233062499999998, 0.09695874999999999]
epoch 6 : [0.16843898437500007, -0.03269673437499997, -0.04363143749999999, -0.04363143749999999, -0.04363143749999999, 0.06787112499999999]
epoch 7 : [0.15825831562500001, -0.035446516406249996, -0.035446516406249996, -0.035446516406249996, -0.035446516406249996, 0.04260527734374989]
epoch 8 : [0.15186752402343762, -0.030129538945312496, -0.030129538945312496, -0.030129538945312496, -0.030129538945312496, 0.018866529999999875]
epoch 9 : [0.14903754452343765, -0.025610108103515623, -0.025610108103515623, -0.025610108103515623, -0.025610108103515623, -0.003913598603515766]
maximum found in unit : 1
```

شکل ۲-۱- خروجی MaxNet در ایپاک های مختلف و واحد برنده

هنگامی که بزرگترین عدد بردار منهای مقدار  $\beta$  عددی مثبت باشد می توان  $\beta$  را از کل اعداد بردار کم کرد و الگوریتم MaxNet را روی بردار جدید پیاده کرد. در این حالت اعدادی که از  $\beta$  کمتر هستند بعد از عبور از تابع فعالساز صفر می شوند، در نتیجه الگوریتم MaxNet روی اعداد بزرگتر از  $\beta$  اعمال می شود

$$\max(a) - \beta > 0$$

```
epoch 1 : [0.29999999999999993, 0.20000000000000007, 0.09999999999999998, 0.0, 0.04999999999999993, 0.24999999999999999]
epoch 2 : [0.20999999999999996, 0.09500000000000011, -0.01999999999999999, -0.13499999999999995, -0.07750000000000004, 0.15249999999999999]
epoch 3 : [0.17287499999999997, 0.04062500000000014, -0.06862499999999999, -0.06862499999999999, -0.06862499999999999, 0.10674999999999999]
epoch 4 : [0.15076874999999995, -0.0013187499999998409, -0.048037500000000004, -0.048037500000000004, -0.048037500000000004, 0.07472499999999997]
epoch 5 : [0.13955999999999996, -0.033824062499999974, -0.033824062499999974, -0.033824062499999974, -0.033824062499999974, 0.052109687499999984]
epoch 6 : [0.13174354687499998, -0.028750453124999978, -0.028750453124999978, -0.028750453124999978, -0.028750453124999978, 0.031175687499999989]
epoch 7 : [0.12706719375, -0.024437885156249978, -0.024437885156249978, -0.024437885156249978, -0.024437885156249978, 0.011414155468749892]
epoch 8 : [0.12535507042968752, -0.020772202382812483, -0.020772202382812483, -0.020772202382812483, -0.020772202382812483, -0.007645923593750107]
maximum found in unit : 1
```

شکل ۲-۲- خروجی MaxNet در ایپاک های مختلف و واحد برنده هنگام کم کردن مقدار ۰.۹ از کل ورودی ها

همانطور که در شکل ۲-۲ می بینیم پس از کم کردن عدد 0.9 از بردار داده شده الگوریتم زودتر (در ۸ ایپاک) به جواب می رسد.

برای پیاده سازی مرتب کردن اعداد از کوچک به نزولی به کمک MaxNet بردار ورودی را به MaxNet می دهیم و MaxNet بزرگترین عضو را مشخص کرده و آن را از لیست حذف می کند، در تابع MaxNet

خود تابع تا وقتی که اندازه بردار ورودی برابر ۱ می باشد به صورت بازگشتی صدا زده می شود، نتیجه نهایی بردار مرتب شده نزولی خواهد بود،

برای پیاده سازی مرتب کردن اعداد از کوچک به بزرگ بردار ورودی ابتدا معکوس شده و سپس عملیات مانند فوق روی آن انجام می شود و در انتها پس از معکوس گیری بردار حاصل اعداد از کوچک به بزرگ مرتب می شوند. شرایط استفاده از این الگوریتم ها مثبت بودن اعداد بردار می باشد.

```
epoch 8 : [ 0.21065461 -0.05087792  0.07765361]
epoch 9 : [ 0.19900656 -0.04324623  0.04605542]
epoch 10 : [ 0.19209825 -0.0367593  0.01620444]
epoch 11 : [ 0.18966759 -0.0312454 -0.0126103 ]
maximum found in unit : 1
input : [0.9, 0.95]
epoch 1 : [0.9 0.95]
epoch 2 : [0.7575 0.815 ]
epoch 3 : [0.63525 0.701375]
epoch 4 : [0.53004375 0.6060875 ]
epoch 5 : [0.43913063 0.52658094]
epoch 6 : [0.36014348 0.46071134]
epoch 7 : [0.29103678 0.40668982]
epoch 8 : [0.23003331 0.3630343 ]
epoch 9 : [0.17557816 0.32852931]
epoch 10 : [0.12629877 0.30219258]
epoch 11 : [0.08096988 0.28324777]
epoch 12 : [0.03848272 0.27110229]
epoch 13 : [-0.00218263 0.26532988]
maximum found in unit : 2
descending sorted vector [1.2, 1.15, 1.1, 1, 0.95]
```

شکل ۳-۲- خروجی MaxNet در ایپاک های مختلف و خروجی مرتب شده به صورت نزولی

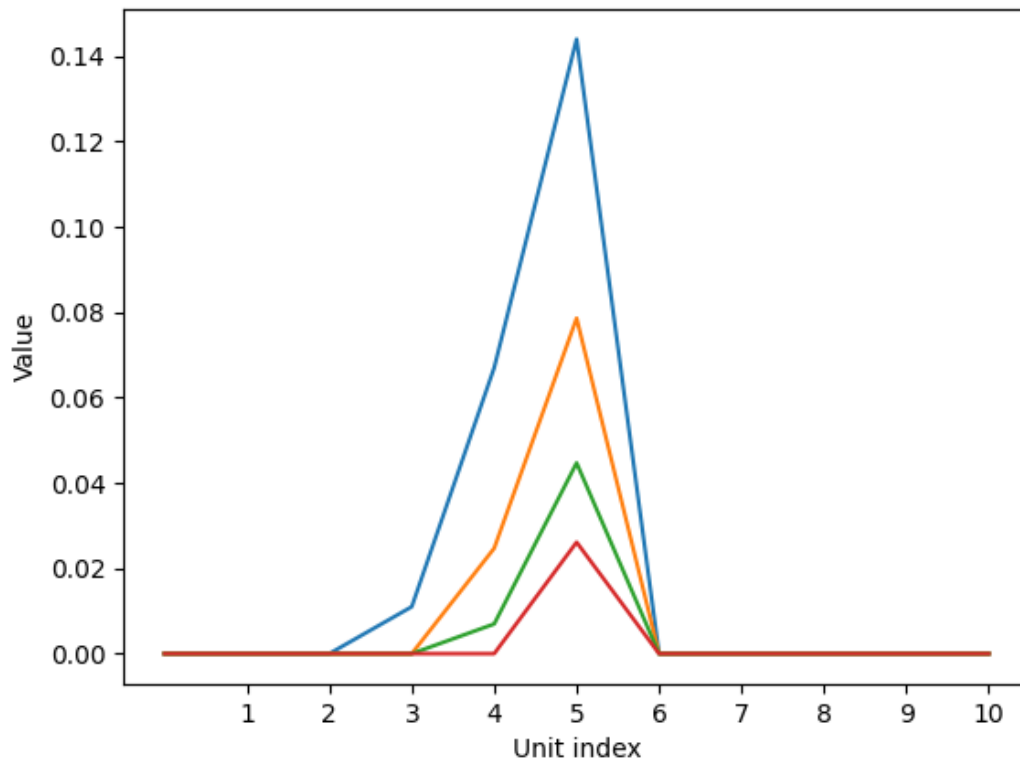
```
epoch 8 : [-0.02759521 0.17392145 0.06878232]
epoch 9 : [-0.03640557 0.1636041 0.04269411]
epoch 10 : [-0.03094473 0.15719998 0.01815349]
epoch 11 : [-0.02630302 0.15447696 -0.00542651]
maximum found in unit : 2
input : [0.8333333333333334, 0.8695652173913044]
epoch 1 : [0.83333333 0.86956522]
epoch 2 : [0.70289855 0.74456522]
epoch 3 : [0.59121377 0.63913043]
epoch 4 : [0.4953442 0.55044837]
epoch 5 : [0.41277695 0.47614674]
epoch 6 : [0.34135494 0.4142302 ]
epoch 7 : [0.27922041 0.36302696]
epoch 8 : [0.22476636 0.3211439 ]
epoch 9 : [0.17659478 0.28742894]
epoch 10 : [0.13348044 0.26093972]
epoch 11 : [0.09433948 0.24091766]
epoch 12 : [0.05820183 0.22676674]
epoch 13 : [0.02418682 0.21803646]
epoch 14 : [-0.00851865 0.21440844]
maximum found in unit : 2
ascending sorted vector ['0.90', '0.95', '1.00', '1.10', '1.15']
```

شکل ۴-۲- خروجی MaxNet در ایپاک های مختلف و خروجی مرتب شده به صورت صعودی

### سوال ۳- Mexican Hat

(الف)

در این قسمت R1 را برابر 0 و R2 را برابر 20 در نظر می گیریم تا تمام واحدهای همسایه هر واحد را به عنوان رقیب آن حساب کند، لذا نتیجه همانند خروجی MaxNet بیشترین واحد پیدا خواهد شد.



شکل ۳-۱- مقدار هر واحد پس از اجرای الگوریتم Mexican Hat ۴ بار

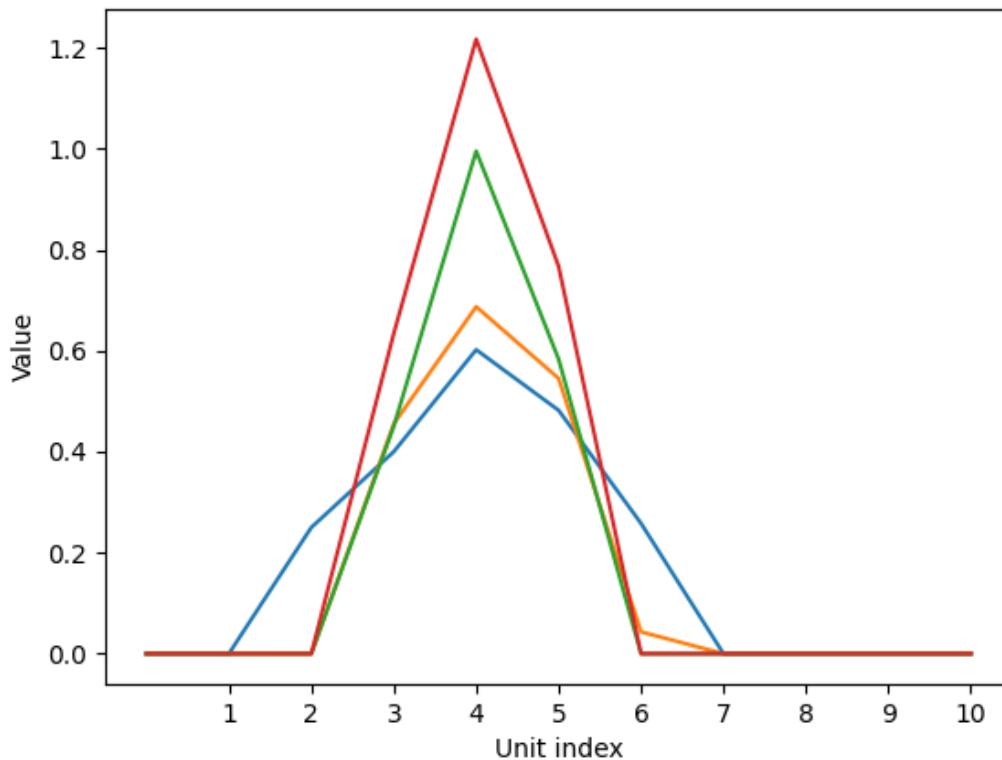
همانطور که در شکل ۳-۱ می بینیم واحد شماره ۵ در ۴ ایتريشن مختلف به عنوان بیشترین واحد مشخص شده است

```
Epoch 1 : [0. 0. 0. 0.011 0.067 0.144 0. 0. 0. 0. 0. ]
Epoch 2 : [0. 0. 0. 0. 0.0247 0.0786 0. 0. 0. 0. 0. ]
Epoch 3 : [0. 0. 0. 0. 0.00696 0.04469 0. 0. 0. 0. 0. ]
Epoch 4 : [0. 0. 0. 0. 0. 0.026118 0. 0. 0. 0. 0. ]
```

شکل ۳-۲- سیگنال خروجی در هر اپاک از اجرای الگوریتم Mexican Hat

(ب)

در این قسمت  $R1$  را برابر 1 و  $R2$  را برابر 3 در نظر می گیریم تا یک همسایگی که واحد های آن بیشترین مقادیر را دارند پیدا شود.



شکل ۳-۳- مقدار هر واحد پس از اجرای الگوریتم Mexican Hat با شعاع  $R1=1, R2=3$

همانطور که در شکل ۳-۳ می بینیم در این حالت همسایگی که واحد های آن بیشترین مقدار را دارند به مرکزیت واحد چهارم پیدا شده است.

Epoch 1 :	[0.	0.	0.25	0.4	0.602	0.482	0.258	0.	0.	0.	0.]
Epoch 2 :	[0.	0.	0.	0.4552	0.6872	0.5452	0.0432	0.	0.	0.	0.]
Epoch 3 :	[0.	0.	0.	0.45008	0.99528	0.58328	0.	0.	0.	0.	0.]
Epoch 4 :	[0.	0.	0.	0.633904	1.217184	0.767104	0.	0.	0.	0.	0.]

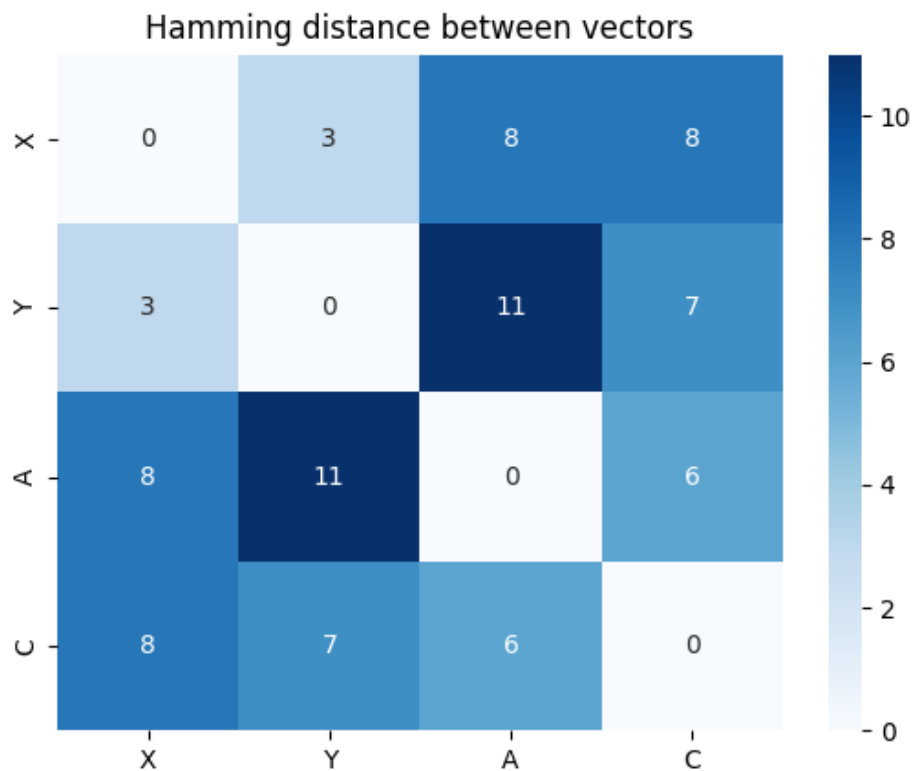
شکل ۳-۴- سیگنال خروجی در هر ایپاک از اجرای الگوریتم Mexican Hat با شعاع  $R1=1, R2=3$

پس با مقایسه قسمت الف و ب نتیجه می گیریم که به کمک تعیین شعاع های  $R1$  و  $R2$  می توانیم مرکز همسایگی که واحد های آن بیشترین مقدار را دارند پیدا کنیم.

## سوال ۴- Hamming Net

(الف)

ماتریس شکل ۴-۱ شامل فاصله Hamming هر حرف از حروف دیگر می باشد



شکل ۴-۱- فاصله hamming هر حرف ورودی از سایر حروف

(ب)

$$w = \frac{y}{2}$$

$$w = \frac{n}{2} = 6$$

هم چنین مقادیر  $C1$  و  $C2$  را به صورت زیر در نظر می گیریم:

$$C1 = 0.6$$

$$C2 = -0.1$$

جدول ۴-۱- ماتریس وزن ها

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	-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
0.5	-0.5	0.5	-0.5
-0.5	0.5	-0.5	0.5
0.5	-0.5	0.5	0.5

پ)

```
Output of net [ 8. 11.  2.  6.]
epoch 1 : [ 8. 11.  2.  6.]
epoch 2 : [ 5.15  8.6 -1.75  2.85]
epoch 3 : [ 3.4325  7.4 -2.49  0.7875]
epoch 4 : [ 2.204375  6.767 -1.743 -0.837375]
epoch 5 : [ 1.189325  6.43634375 -1.34570625 -1.34570625]
epoch 6 : [ 0.22387344  6.257945 -1.14385031 -1.14385031]
epoch 7 : [-0.71481831  6.22436398 -0.97227277 -0.97227277]
maximum found in unit : 2
```

شکل ۴-۲- خروجی شبکه hamming و سپس اعمال الگوریتم MaxNet روی آن و نتیجه واحد برنده

بردار خروجی شبکه را به MaxNet می دهیم و واحد شماره دوم که بردار حرف Y باشد به عنوان برنده انتخاب می شود. به این معنی است که طبق انتظار ما حرف T بیشترین شباهت را به Y دارد.