به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

شبکه های عصبی و یادگیری عمیق

تمرین سری چهارم

نام و نام خانوادگی	محمدحسین بدیعی
شماره دانشجویی	810199106
تاریخ ارسال گزارش	12 بهمن

فهرست گزارش سوالات

سوال 4 – **maxnet&hamming net** 3

سوال 5 – **SOM** 10

سوال 4 – maxnet&hamming net

۱- آرایه ی زیر را در نظر بگیرید:

$$[۰.۲۴, ۰.۳, ۰.۴۵, ۰.۵۷, ۰.۸, ۰.۶۹, ۰.۴۲, ۰.۲۶, ۰.۱۴]$$

الف) درخصوص نحوه ی عملکرد شبکه ی ***Mexican Hat*** توضیح مختصری ارائه دهید؛ سپس با استفاده از این شبکه ماکزیمم مقدار آرایه را برای $R_۱ = ۱$ و $R_۲ = ۴$ پیدا کنید. در هر تکرار، نمودار اندیس اعضای آرایه و سیگنال خروجی را رسم کنید. تابع فعال ساز را نیز بصورت زیر در نظر بگیرید:

$$f(x) = \begin{cases} ۰ & x < ۲ \\ x & ۰ \le x < ۲ \\ ۲ & x \ge ۲ \end{cases}$$

پاسخ بخش 4-1-الف)

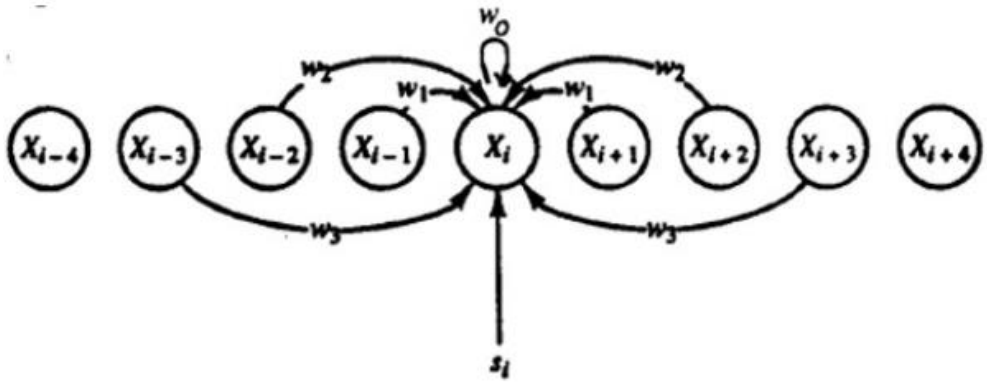
توضیح عملکرد شبکه Mexican Hat:

شبکه‌ی Mexican hat یکی از نمونه‌های شبکه‌های مبتنی بر رویکرد رقابتی است. در این شبکه تعداد $m > 1$ نودِ به‌ترتیب‌قرارگرفته وجود دارد. هدف این است که نودی را بیابیم که یک بیشینه‌ی نرم بین نودها ایجاد کند. به عنوان مثال نودهای به ترتیب شده زیر را در نظر بگیرید. در این شبکه دو شعاع تعریف می‌شود به صورت R_1 و R_2 به نحوی که در همسایگی شعاع R_1 پیرامون هر نود، نودها با یکدیگر همکاری کرده و از شعاع R_1+1 تا R_2 پیرامون هر نود، نودها به عنوان نودهای رقیب تلقی می‌شوند. لذا در این شبکه سه دسته نورون داریم:

- نورون‌های همکار (در همسایگی تا شعاع R_1 در دو طرف نورون)
- نورون های‌رقیب (از شعاع R_1+1 تا R_2 در دوطرف نورون)
- نورون‌های بی‌تفاوت (از شعاع R_2 الی اخر در دوطرف نورون)

حال در این الگوریتم هر نورون با وزن‌دهی به خودش و همسایگانش سعی می‌نماید با سایر نورون‌ها رقابت می‌نماید تا بیشینه‌ی نرم مشخص شود. وزن خود نورون و نورون‌های همکار مثبت و وزن نورون‌های رقیب منفی لحاظ می‌شود و خارج از شعاع رقابت نیز وزن صفر یا به عبارتی در محاسبات مربوط به آپدیت هر نورون مشارکتی نمی‌کند.

ابتدا نورون‌های به ترتیب قرار گرفته در ذیل را مشاهده فرمایید.



شکل 1اتصالات داخلی بین نورون‌های شبکه‌ی **Mexican hat** برای نورون x_i

نحوه اپدیت شدن هر نورون به صورت زیر است.

$$x_i(t) = f[s_i(t) + \sum_k w_k x_{i+k}(t-1)]$$

شبه کد مربوط به آموزش شبکه را مشاهده می‌کنیم.

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

$w_k = C_1$ for $k = 0, \ldots, R_1$ $(C_1 > 0)$

$w_k = C_2$ for $k = R_1 + 1, \ldots, R_2$ $(C_2 < 0)$.

Initialize **x_old** to **0**.

Step 1. Present external signal **s**:

$\mathbf{x} = \mathbf{s}$.

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

$x\_old_i = x_i$.

Set iteration counter: $t = 1$.

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

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

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

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

$x_i = \min(x\_max, \max(0, x_i))$ $(i = 1, \ldots, n)$.

Step 5. Save current activations in **x_old**:

$x\_old_i = x_i$ $(i = 1, \ldots, n)$.

Step 6. Increment iteration counter:

$t = t + 1$.

Step 7. Test stopping condition:
If $t < t\_max$, continue; otherwise, stop.

شکل 2 شبه کد آموزش شبکه **Mexican hat**

گام 0: همانطور که مشاهده می‌فرمایید در گام 0 پارامترهای R_1 و R_2 و t_{max} را initial می‌نماییم. همچنین برای نورون‌های همکار و خود نورون وزن C_1 که مقداری مثبت است را در نظر گرفته و برای نورون‌های رقیب برای هر نورون وزن C_2 که مقداری منفی است را در نظر می‌گیریم.

گام 1: سیگنال خروجی را به عنوان ورودی قرار داده و در ورودی قدیم ذخیره می‌نماییم. سپس iteration را بر روی مقدار 1 تنظیم می‌کنیم.

گام2: تا زمانی که زمان فعلی کمتر از t_{max} است مراحل زیر را اجرا کرده:

- 3 و 4. مطابق با فورمول $x_i(t) = f[s_i(t) + \sum_k w_k x\_old_{i+k}(t-1)]$ ، x ها را آپدیت می‌نماییم.
- X فعلی را به در x_{old} ذخیره می‌کنیم.
- یکی به مقدار iteration اضافه می‌نماییم.
- شرط توقف را بررسی می‌کنیم. یعنی زمان اجرای الگوریتم از ماکسیمم زمان تعیین شده فراتر رود.

حال طبق خواسته‌ی سوال این شبکه را با معماری فوق در کلاسی با نام MexicanHat پیاده نمودیم.

در نهایت خروجی‌های شبکه را به صورت زیر استخراج نمودیم. لذا قادیر پارامترها را به صورت زیر تنظیم می‌نماییم.

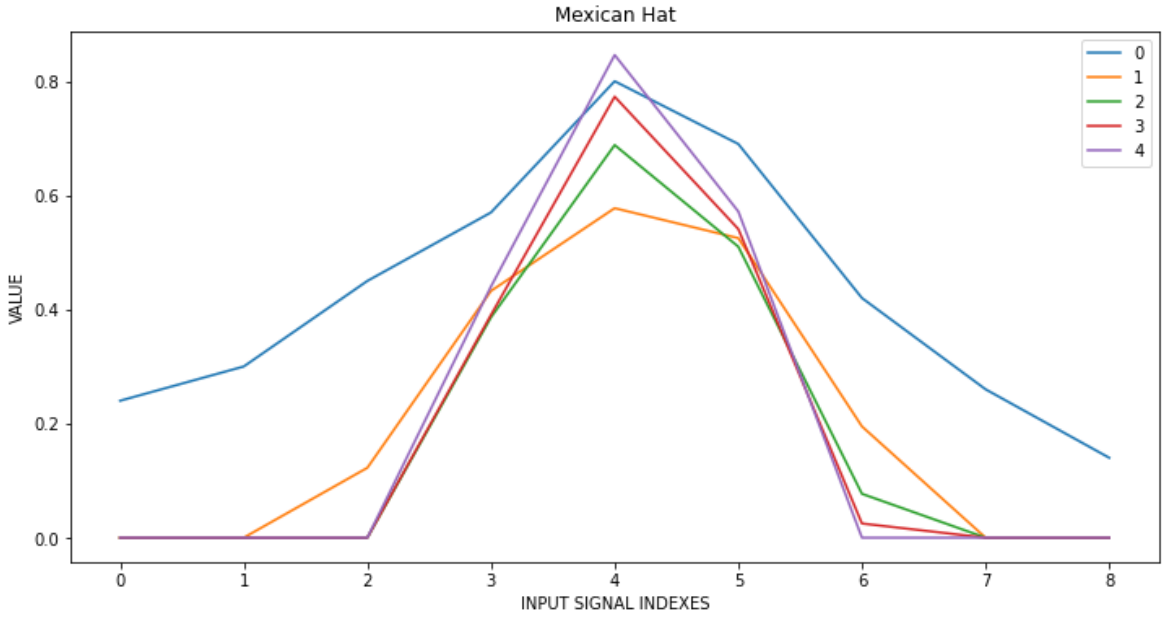
$$\begin{cases} C_1 = 0.5 \\ C_2 = -0.25 \\ R_1 = 1 \\ R_2 = 4 \\ t_{max} = 5 \end{cases}$$

در مقادیر آپدیت شده‌ی آرایه در هر گام را بررسی می‌کنیم. نتایج استخراج شده از کد به صورت زیر است.

STEPS	input 0	input 1	input 2	input 3	input 4	input 5	input 6	input 7	input 8
STEP0	0.24	0.3	0.45	0.57	0.8	0.69	0.42	0.26	0.14
STEP1	0	0	0.1225	0.4325	0.5775	0.525	0.195	0	0
STEP2	0	0	0	0.3862	0.6881	0.51	0.0769	0	0
STEP3	0	0	0	0.3905	0.773	0.5409	0.0248	0	0
STEP4	0	0	0	0.4403	0.846	0.5718	0	0	0

شکل 3 مقادیر آپدیت شده آرایه در هر گام برای **Mexican Hat**

سپس در هر تکرار نمودار مقادیر آرایه رسم نمودیم که به صورت زیر می‌باشد:



شکل 4 نمودار آپدیت مقادیر آرایه در هر تکرار برای **Mexican Hat**

پاسخ بخش 4-1-ب)

در این شبکه در زمانیکه شعاع همکاری برای صفر و شعاع رقابت به اندازه تمامی همسایگی (یا از نظر مقداری بی‌نهایت) باشد و نیز $C_1=1$ و $C_2=-\varepsilon$ باشد، عملا شبکه به یک شبکه‌ی MaxNet تبدیل می‌شود چراکه هیچ همکاری‌ای وجود نداشته و هر نورون با دیگر نورون‌های موجود در شبکه رقیب می‌شود. و در آپدیت شبکه برای هر نورون، دقیقا مقدار نورون با $\varepsilon-$ برابر مجموع سایر نورون‌ها جمع می‌شود که دقیقا همان رفتار شبکه‌ی MaxNet در آموزش سیگنال است. لذا در چنین شبکه‌ای اعداد کوچکتر سریعتر از سایر اعداد صفر شده و بدین صورت اعداد با ترتیب کوچکی sort می‌شوند و در نهایت عدد ماکسیمم به عنوان آخرین عدد باقی می‌ماند چون که سایر عناصر صفر می‌شوند و هیچ عنصر دیگری نیست که در اپسیلون ضرب شود و از مقدار ماکسیمم آپدیت شده نهایی بتواند مقداری را کم کند.

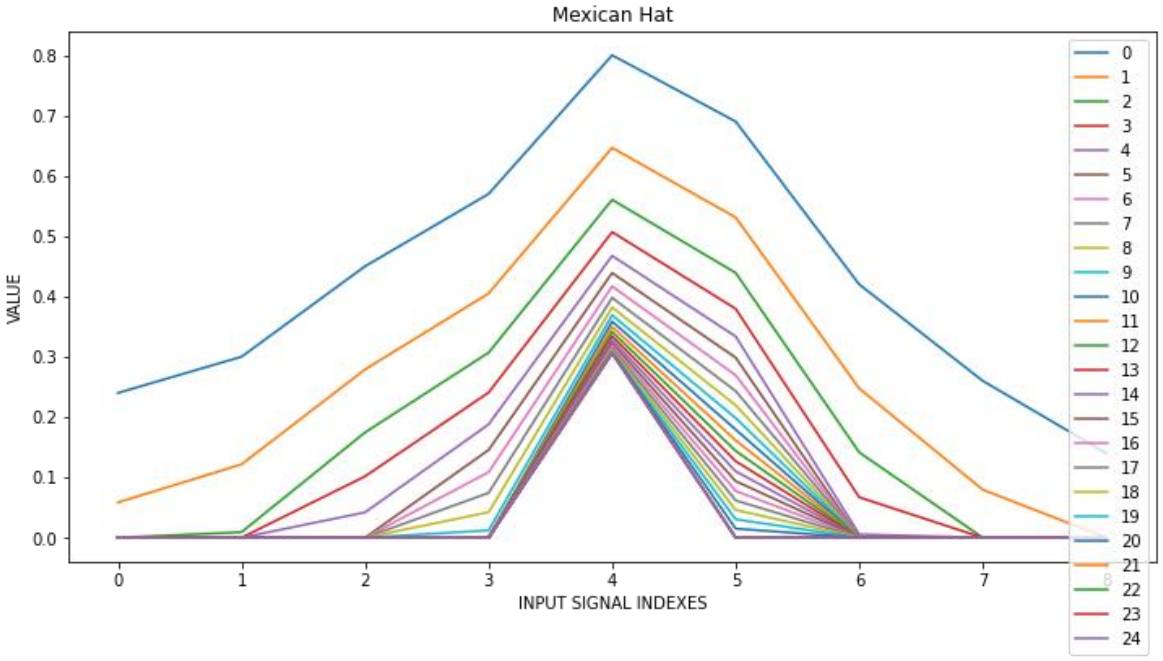
$$\begin{cases} C_1 = 1 \\ C_2 = 0.05 \\ R_1 = 0 \\ R_2 = \infty \\ t_{max} = 25 \end{cases}$$

حال مقادیر آرایه در هر آپدیت را برای این بخش نیز بررسی می‌کنیم که به شرح زیر است.

STEPS	input 0	input 1	input 2	input 3	input 4	input 5	input 6	input 7	input 8
STEP0	0.24	0.3	0.45	0.57	0.8	0.69	0.42	0.26	0.14
STEP1	0.0585	0.1215	0.279	0.405	0.6465	0.531	0.2475	0.0795	0
STEP2	0	0.0091	0.1745	0.3068	0.5604	0.4391	0.1414	0	0
STEP3	0	0	0.1017	0.2406	0.5068	0.3795	0.0669	0	0
STEP4	0	0	0.042	0.1878	0.4674	0.3337	0.0055	0	0
STEP5	0	0	0	0.1454	0.439	0.2986	0	0	0
STEP6	0	0	0	0.1085	0.4168	0.2693	0	0	0
STEP7	0	0	0	0.0742	0.3979	0.2431	0	0	0
STEP8	0	0	0	0.0422	0.382	0.2195	0	0	0
STEP9	0	0	0	0.0121	0.3689	0.1983	0	0	0
STEP10	0	0	0	0	0.3584	0.1792	0	0	0
STEP11	0	0	0	0	0.3494	0.1613	0	0	0
STEP12	0	0	0	0	0.3414	0.1438	0	0	0
STEP13	0	0	0	0	0.3342	0.1268	0	0	0
STEP14	0	0	0	0	0.3278	0.11	0	0	0
STEP15	0	0	0	0	0.3223	0.0937	0	0	0
STEP16	0	0	0	0	0.3177	0.0775	0	0	0
STEP17	0	0	0	0	0.3138	0.0617	0	0	0
STEP18	0	0	0	0	0.3107	0.046	0	0	0
STEP19	0	0	0	0	0.3084	0.0304	0	0	0
STEP20	0	0	0	0	0.3069	0.015	0	0	0
STEP21	0	0	0	0	0.3061	0	0	0	0
STEP22	0	0	0	0	0.3061	0	0	0	0
STEP23	0	0	0	0	0.3061	0	0	0	0
STEP24	0	0	0	0	0.3061	0	0	0	0

شکل 5 مقادیر آرایه به صورت عددی در هر بار تکرار

حال به صورت نمودار نیز مقادیر آپدیت شده را رسم می‌کنیم که به شرح زیر است.



شکل 6 نمودار آپدیت مقادیر آرایه در هر تکرار برای **Mexican Hat** (بخش ب قسمت 1 که به صورت **MaxNet** شبکه را طراحی کردیم)

مشاهده می‌فرمایید که عنصر پنجم یا x[4] که برابر با 0.8 است به عنوان بیشترین مقدار در ارایه انتخاب می‌شود که کاملا درست ارزیابی شده است.

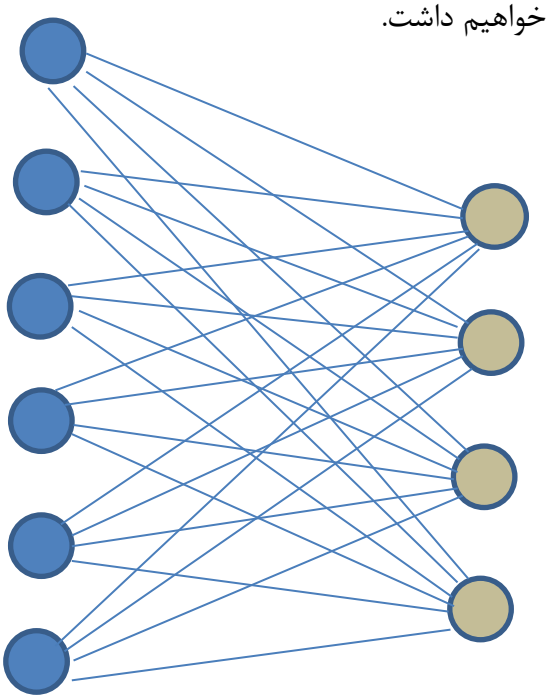
پاسخ بخش 4-2-الف)

معماری شبکه را با توجه به ساختار همینگ‌نت طراحی می‌کنیم. بدین صورت است که چون چهار بردار پایه یا به عبارتی چهار وکتور رفرنس داریم، لذا چهار نورون خروجی برای شبکه لحاظ می‌کنیم. مطابق با ساختار همینگ‌نت می‌دانیم که وزن‌های هر نورون خروجی برابر با نصف مقدار وکتور رفرنس است، فلذا وزن‌ها را بدین صورت متناظر با هر بردار رفرنس، مقدار دهی می‌کنیم به گونه‌ای که وزن‌های نورون مربوطه نصف مقدار بردار مرجعی باشد که آن نورون خروجی نمایانگرِ میزان شباهت ورودی به بردار مرجع مذکور است.

لذا هر نورون خروجی مطابق با فورمول مندرج در ذیل، میزان شباهت بردار ورودی تا بردار رفرنس متناظر با آن نورون خروجی را ارائه می‌کند.

$$a_i = x_i\left(\frac{e_i}{2}\right) + \left(\frac{6}{2}\right) = x_i\left(\frac{e_i}{2}\right) + 3 \quad for\ i = 1,2,3,4$$

در واقع $\left(\frac{e_i}{2}\right)$ همان وزن‌های مربوط به نورون iام است. همچنین توجه داریم که هرچه a_i به عنوان خروجی iام مقدار بیشتری نسبت به سایر خروجی‌ها داشته باشد، بردار ورودی شباهت بیشتری نسب به نورون i یا همان بردار پایه e_i خواهد داشت. همچنین بایاس شبکه نیز که برابر تعداد ابعاد بردارهای پایه (برابر با تعداد ابعاد ورودی است) تقسیم به دو است که برای تمامی نورون‌های خروجی مقدار 3 را به عنوان بایاس لحاظ می‌کنیم. لذا نهایتا ساختار نورون‌ها را به صورت زیر خواهیم داشت.



پاسخ بخش 4-2-ب)

طبق خواسته‌ی سوال بردارهای زیر را به عنوان ورودی در نظر گرفته و به شبکه‌ی طراحی شده همینگ‌نت خود اعمال میکنیم.

$v0$=[1,−1,1,1,−1,1]
$v1$=[−1,−1,1,−1,1,−1]
$v2$=[1,1,1,1,−1,−1]
$v3$=[−1,−1,−1,1,1,−1]
$v5$=[1,1,1,−1,−1,−1]
$v6$=[1,−1,−1,1,1,1]
$v7$=[−1, 1, −1, −1, −1, 1]

مشاهده می‌فرمایید که نتایج زیر حاصل شده است. توجه کنید که گاها چند بردار رفرنس دارای شباهت یکسان با ورودی است که یکی از بردارها به عنوان مشابه‌ترین بردار را چاپ می‌کنیم. لذا داریم:

```
Input :[-1 -1  1 -1  1 -1] Reference [ 1. -1.  1. -1. -1. -1.]
Input :[ 1  1  1  1 -1 -1] Reference [ 1. -1.  1. -1. -1. -1.]
Input :[-1 -1 -1  1  1 -1] Reference [ 1. -1.  1. -1. -1. -1.]
Input :[ 1 -1  1  1 -1  1] Reference [-1. -1.  1.  1. -1.  1.]
Input :[ 1  1  1 -1 -1 -1] Reference [ 1. -1.  1. -1. -1. -1.]
Input :[ 1 -1 -1  1  1  1] Reference [ 1. -1.  1. -1. -1. -1.]
Input :[-1  1 -1 -1 -1  1] Reference [ 1. -1.  1. -1. -1. -1.]
```

شکل 7 نتایج پیاده‌سازی شبکه همینگ نت طراحی شده

تمامی شبیه‌سازیهای انجام گرفته برای سوال چهارم در پوشه‌ی codes و در فایل Q4.ipynb ضمیمه شده است.

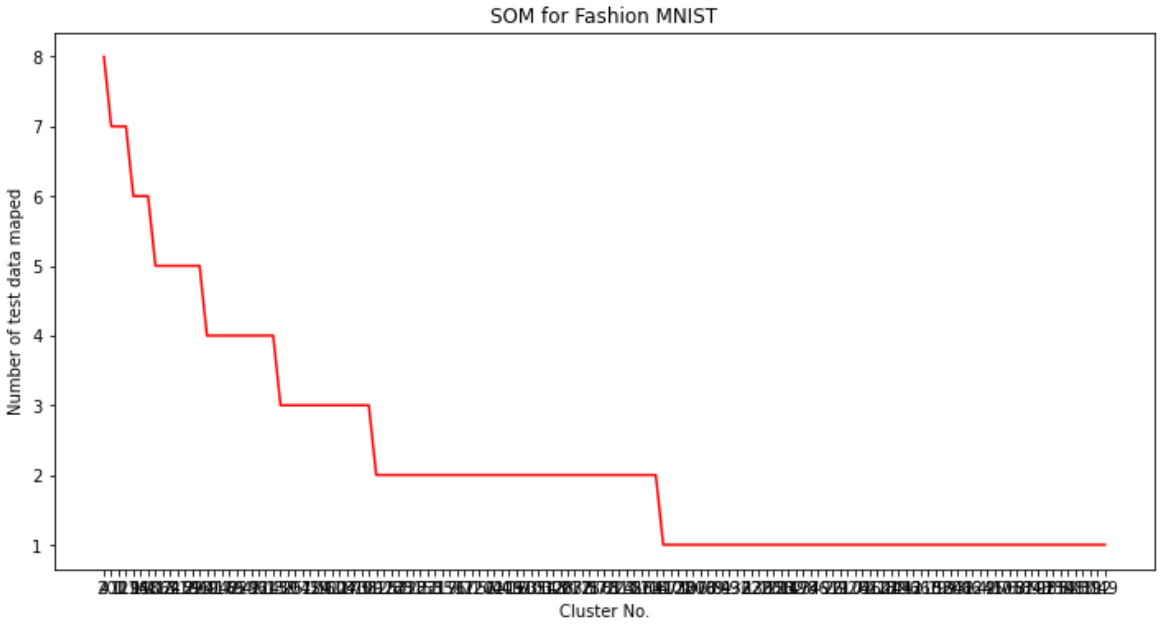
سوال 5 – SOM

بخش الف)

طبق خواسته‌ی سوال به جهت کلاسترینگ دیتاست fashion mnist از یک شبکه SOM را با 225 نورون استفاده کردیم. لذا کلاسی را با نام SOMFashionMNIST پیاده کردیم. همچنین از یک فایل csv شامل نمونه‌های این دیتاست استفاده کرده و 1000 داده اول آن را به عنوان دیتای آموزش یا train در نظر گرفتیم و نیز 300 داده بعدی را طبق صورت مسأله به عنوان تست در لحاظ نمودیم.

پاسخ بخش ب)

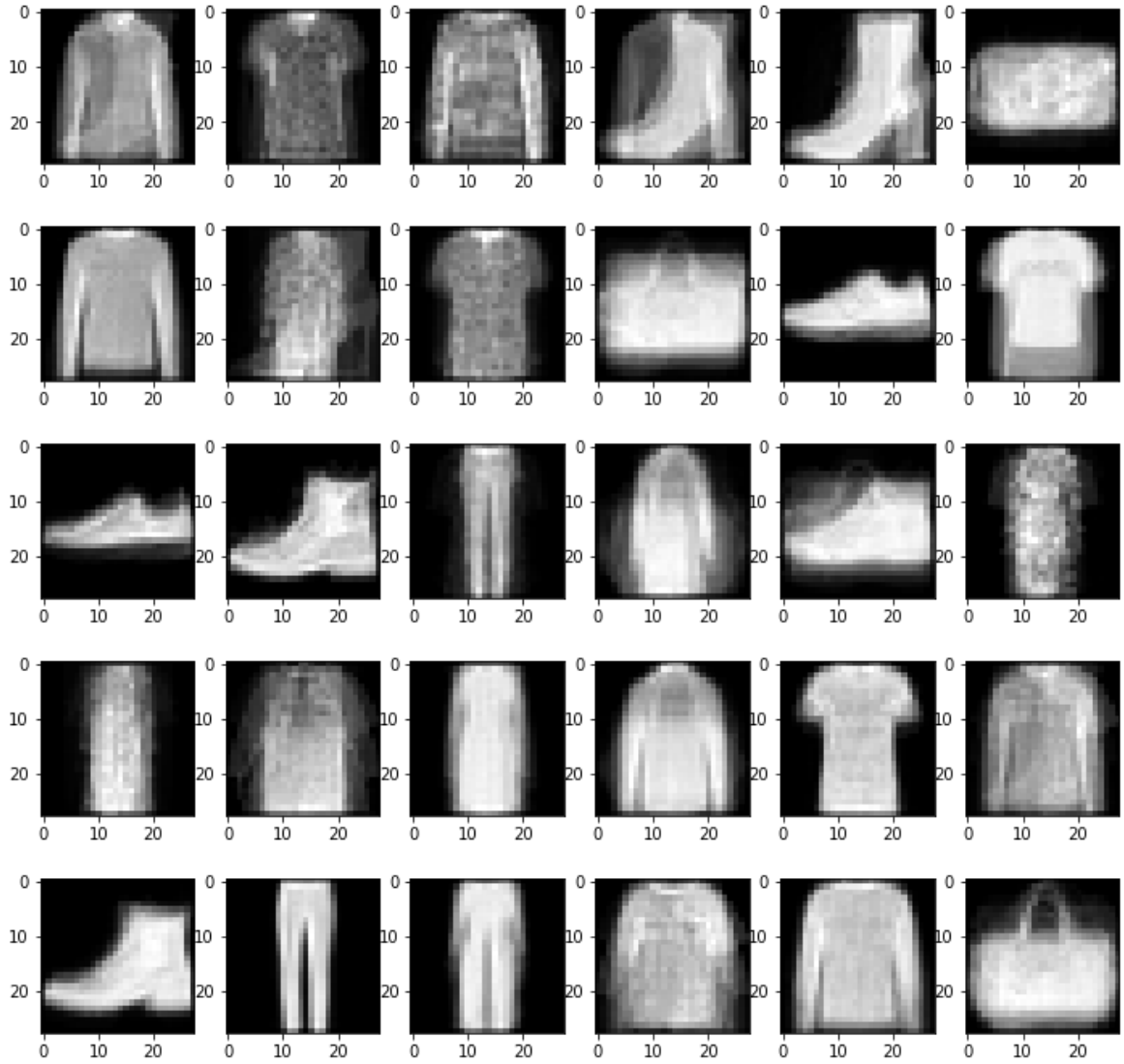
طبق خواسته‌ی سوال نمودار داده‌گان هر خوشه را رسم نمودیم که نتایج به شرح زیر است. در واقع محور افقی شماره خوشه و محور عمودی تعداد داده‌های تست نگاشت شده به خوشه‌ی مربوطه می‌باشد.



شکل 8 نمودار تعداد داده‌های هر خوشه

پاسخ بخش ج)

این الگوریتم در هر ایپاک داده‌هایی که دارای فاصله اقلیدسی کمتری با وزن‌های یک نورون خروجی دارند را در یک کلاستر قرار دهد. بدین صورت به نظر می‌رسد که وزن‌ها به عنوان بردارهای مرجعی تبدیل می‌شوند که می‌توانند عامل تعیین کننده شباهن بین ورودی‌ها و نتیجتا قرار دادن آنها در کلاسترها بر حسب شباهت‌شان باشند. پس انتظار می‌رود که وزن‌ها در هر ایپاک به گونه‌ای آپدیت شوند که به یک عامل شباهت و یا به عبارتی مرجع برای سایر ورودی‌ها تبدیل شوند.



پاسخ بخش د)

تصاویر وزن‌های خوشه‌های چگال که در واقع کلاسترهایی با بیشترین تعداد عضو و نیز دارای شباهت زیاد به مرجع هستند، به صورت زیر می‌باشد.