به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

شبکه های عصبی و یادگیری عمیق

تمرین چهارم

نام و نام خانوادگی	علی ایزدی
شماره دانشجویی	۸۱۰۱۹۹۱۰۲
تاریخ ارسال گزارش	۱۲ بهمن ۱۴۰۰

سوال ۴- maxnet&hammingnet

قسمت ۱:

الف)

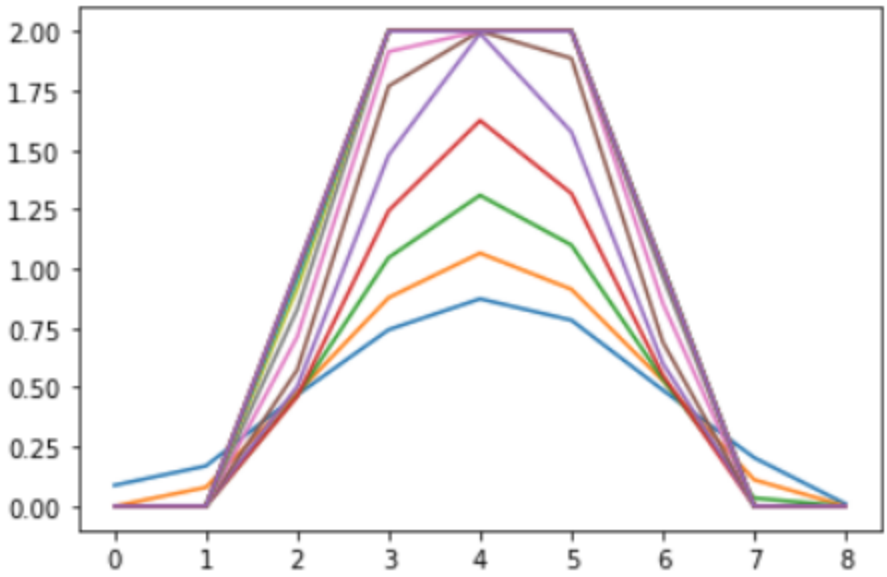
شبکه MexicanHat یک ماکسیمم نرم با توجه به همسایگی اعداد را به دست می آورد.
در این شبکه نودهای همکار با هم نودهایی هستند که در فاصله R1 از هم قرار دارند و نودها رقیب نیز در در فاصله R1 تا R2 از هم قرار داشته و بقیه نودهای با فاصله بیشتر از R2 از هر نود در نظر گرفته نمیشوند.

به روزرسانی مقادیر نودهای شبکه بر اساس وزن های مثبت در فاصله R1 از هر نود و وزن های منفی در فاصله R2-R1 از هر نود و وزن صفر در فاصله بزرگتر از R2 انجام میشود.

با به روزرسانی نودهای شبکه تا t_max ماکسیمم نرم نودها بر اساس همسایه هایشان به دست می آید.

در صورت سوال تابع داده شده قسمت اولش x<2 در نظر گرفته شده است که فکر میکنم اشتباه تایپی رخ داده و در پیاده سازی x<0 در نظر گرفته شد.

با پیاده سازی الگوریتم با پارامترهای R1=1 و R2=4 و c1=0.5 و c2=-0.1 و t_max=15 مقادیر نودهای شبکه در هر iteration=t در شکل ۴-۱ آورده شده است.
همان طور که مشاهده میشود الگوریتم توانسته است کلاه مکزیکی یا همان ماکسیمم نرم را بر روی عنصر پنجم که مقدار ۰.۸ و ماکسیمم داشت را ایجاد کند.



شکل ۴-۱ نمودار سیگنال خروجی الگوریتم mexican-hat

ب)

الگوریتم Mexican hat زمانی مثل الگوریتم maxnet عمل میکند که اولا وزن های زیر را برای C1 و C2 در نظر بگیریم:

C1 = 1

C2 = -e where e< 1/m => e= 0.1 < 1/9 => **C2 = -0.1**

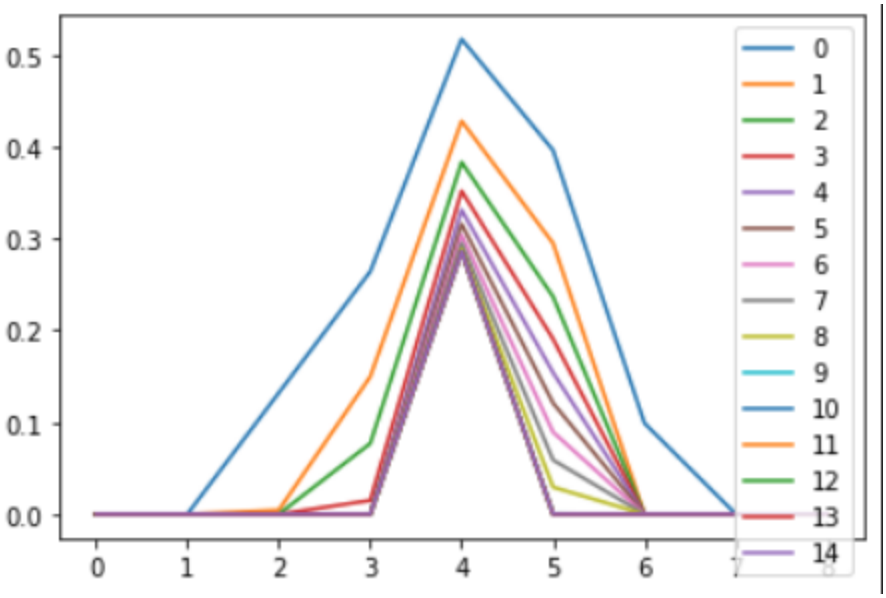
حال هر نود به خودش تنها باید وزن C1=1 بگیرد بنابراین شعاع R1 را برابر یک در نظر میگیریم و بقیه نودها باید وزن e- بگیرند که بنابراین شعاع R2 را ۹ یعنی تعداد داده ها در نظر میگیریم.

R1 = 0

R2 = 9

T_max را هم مشابه با قسمت الف ۱۵ در نظر میگیریم.

در شکل ۴-۲ نمودار سیگنال خروجی آورده شده است.. همان طور که مشاهده میشود بر خلاف قبل در آخر فقط مقدار پنجمین عنصر بزرگتر از صفر است و بقیه خانه ها صفر هستند و شبکه توانسته است maximum شبکه را پیدا کند.

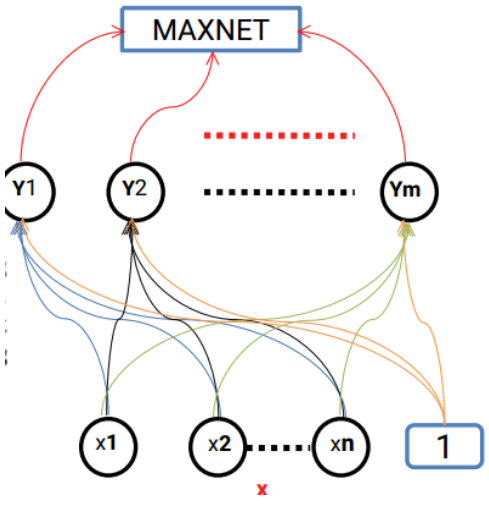


شکل ۴-۲ نمودار سیگنال خروجی الگوریتم mexican hat در حالتی که مشابه با maxnet عمل میکند.

قسمت ۲:

الف)

از شبکه Hamming net برای پیاده سازی مسئله خواسته شده استفاده خواهیم کرد.
معماری شبکه مطابق با شکل ۴-۳ است.



شکل ۴-۳ معماری شبکه Hamming net

هدف این شبکه این است که m نود برای بردارهای Y1تا Ym که بردارهای پایه هستند در نظر بگیرد و به ازای بردارهای ورودی x که n بعدی هستتند تشخیص دهد که شبیه ترین بردار پایه به بردار ورودی x کدام است.

وزن های شبکه بر اساس رابطه زیر مقداردهی میشوند که e همان بردارهای پایه هستند تا شبیه ترین بردارهای ورودی به بردارهای پایه بر اساس فاصله hamming پیدا شوند.

$$w_{ij} = \frac{e_i(j)}{2}, (i = 1, \ldots, n; j = 1, \ldots, m).$$

هم چنین مقدار bias بر اساس رابطه زیر محاسبه میشود که n تعداد ابعاد ورودی است.

$$b_j = \frac{n}{2}, (j = 1, \ldots, m).$$

سپس مطابق با الگوریتم زیر به ازای هر بردار ورودی x مقادیر y1 تا ym طبق رابطه داده شده محاسبه میشوند و در مرحله آخر برای پیدا کردن شبیه ترین بردار y به بردار بر روی mعدد y یک الگوریتم maxnet اجرا میشود.

The application procedure for the Hamming net is:

Step 1. For each vector **x**, do Steps 2–4.

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

$$y\_in_j = b_j + \sum_i x_i w_{ij}, (j = 1, \ldots, m).$$

Step 3. Initialize activations for MAXNET:

$$y_j(0) = y\_in_j, (j = 1, \ldots, m).$$

Step 4. MAXNET iterates to find the best match exemplar

ب)

نتایج زیر برای بردارهای v1 تا v7 در جدول ۴-۱ آورده شده است.
همان طور که مشاهده میشود خروجی Y_out به ازای هر ورودی آورده شده است سپس بزرگترین مقدار Y_out توسط الگوریتم Maxnet محاسبه شده است که چون این الگوریتم فقط یک مقدار به عنوان ماکسیمم برمیگرداند ممکن است برداری ورودی به چند بردار شبیه باشند که در ستون بعدی جدول بقیه بردارهای شبیه نیز آورده شده است در واقع یعنی چند مقدار maximum داشته ایم.

V	Y_out	Max Net	Others similar
v1	[4. 3. 3. 2.]	e1	e2
v2	[4. 3. 3. 4.]	e1	e4
v3	[2. 5. 3. 4.]	e2	-
v4	[4. 1. 5. 2.]	e3	-
v5	[5. 2. 2. 3.]	e1	-
v6	[2. 3. 3. 4.]	e4	-
v7	[2. 3. 3. 2.]	e2	e3

جدول ۴-۱ خروجی الگوریتم Hamming net برای بردارهای v1 تا v7

- برای پیاده سازی maxnet نیز در هر iteration مقادیر a مطابق با رابطه زیر به روز رسانی شده اند.

$$\boldsymbol{a}^{new} = \boldsymbol{f}\left(\begin{bmatrix} 1 & -\varepsilon & \cdots & -\varepsilon \\ -\varepsilon & 1 & \cdots & -\varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ -\varepsilon & -\varepsilon & \cdots & 1 \end{bmatrix} \boldsymbol{a}^{old}\right)$$

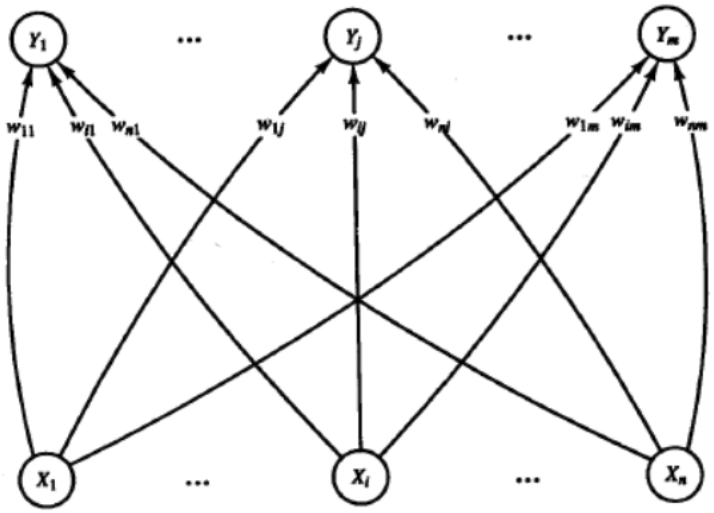
که در آن epsilon برابر 0.2 و f مطابق با زیر محاسبه شده است.

$$f(x) = \begin{cases} x & \text{if } x \geqslant 0 \\ 0 & \text{otherwise.} \end{cases}$$

در نهایت نیز پس از این که تنها یک مقدار a غیر صفر بود iterationها متوقف میشود و در نهایت index خانه ای که غیر صفر است به عنوان عدد maximum برگردانده میشود.

سوال ۵- SOM

روش SOM یک شبکه مطابق با شکل ۵-۱ ایجاد میکند.
تا هر داده را به یکی از m کلاستر y تعلق دهد.



شکل ۵-۱ شبکه SOM

در این شبکه نورون های مجاور با یک تعریف همسایگی یک بعدی یا دو بعدی ابتدا با هم همکاری میکنند و در نهایت با هم رقابت میکنند.

به ازای هر ورودی D_j را برای هر کلاستر مطابق با رابطه زیر محاسبه میکنیم.

$$D(j) = \sum_i (w_{ij} - x_i)^2.$$

سپس D_j مینیمم را پیدا میکنیم و سپس وزن های نود jام و همسایه های آن را طبق رابطه زیر به روز رسانی میکنیم.

$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + \alpha[x_i - w_{ij}(\text{old})].$$

پیاده سازی

از دیتاست fashin mnist کتابخانه keras استفاده کرده ایم.
هزار تای اول را به عنوان داده آموزش و ۳۰۰ تای بعدی را به عنوان داده تست در نظر گرفته ایم.
داده های را بر ۲۵۵ تقسیم کرده ایم تا نرمال شوند.

مقدار alpha=0.6 در نظر گرفته ایم.

الف)

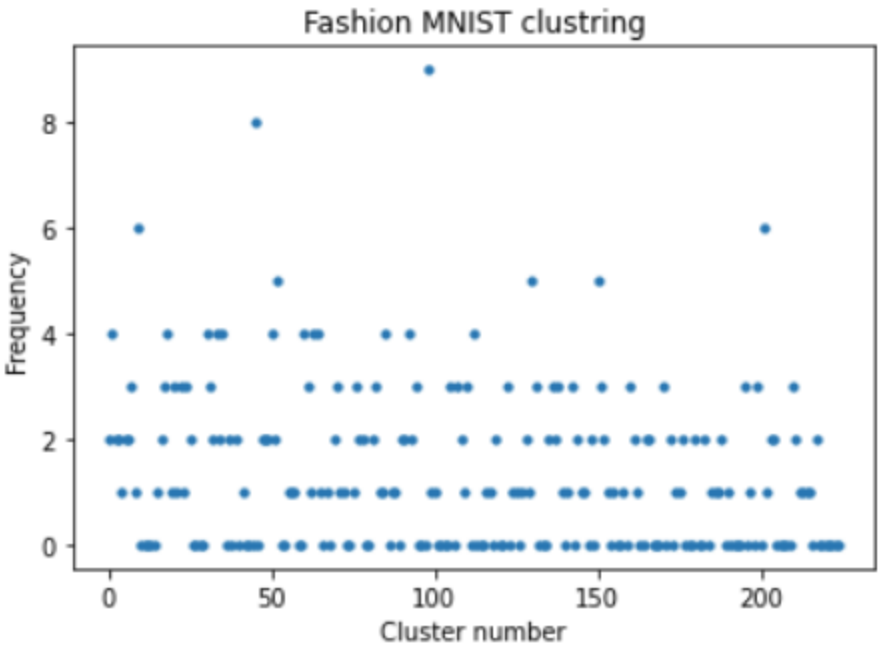
طبق تعریف همسایگی گفته شده نورون ها را بر روی یک شبکه ۱۵*۱۵ با فرم مجاورت مربعی با شعاع R=1 در نظر گرفته ایم.

برای این کار به روز رسانی را مطابق با شبه کد زیر علاوه بر خود نود چهار نود اطراف آن را نیز به روز رسانی کرده ایم.

```
W[m][n] += alpha*(x - W[m][n])
if m != W.shape[0]-1:
    W[m+1][n] += alpha*(x - W[m+1][n])
if m != 0 :
    W[m-1][n] += alpha*(x - W[m-1][n])
if n != W.shape[1]-1:
    W[m][n+1] += alpha*(x - W[m][n+1])
if n != 0 :
    W[m][n-1] += alpha*(x - W[m][n-1])
```

ب)

بعد از فرآیند آموزش داده های تست را ورودی شبکه داده ایم و min D_j به عنوان کلاستر انتخاب میشود.
در شکل ۵.۲ نمودار تعداد اعضای هر کلاستر و شماره کلاستر آورده شده است.
همان طور که مشاهده میشود بین صفر تا ۹ تعداد عناصر کلاسترها متغیر است.



شکل ۵-۲ نمودار تعداد داده های هر خوشه

ج)

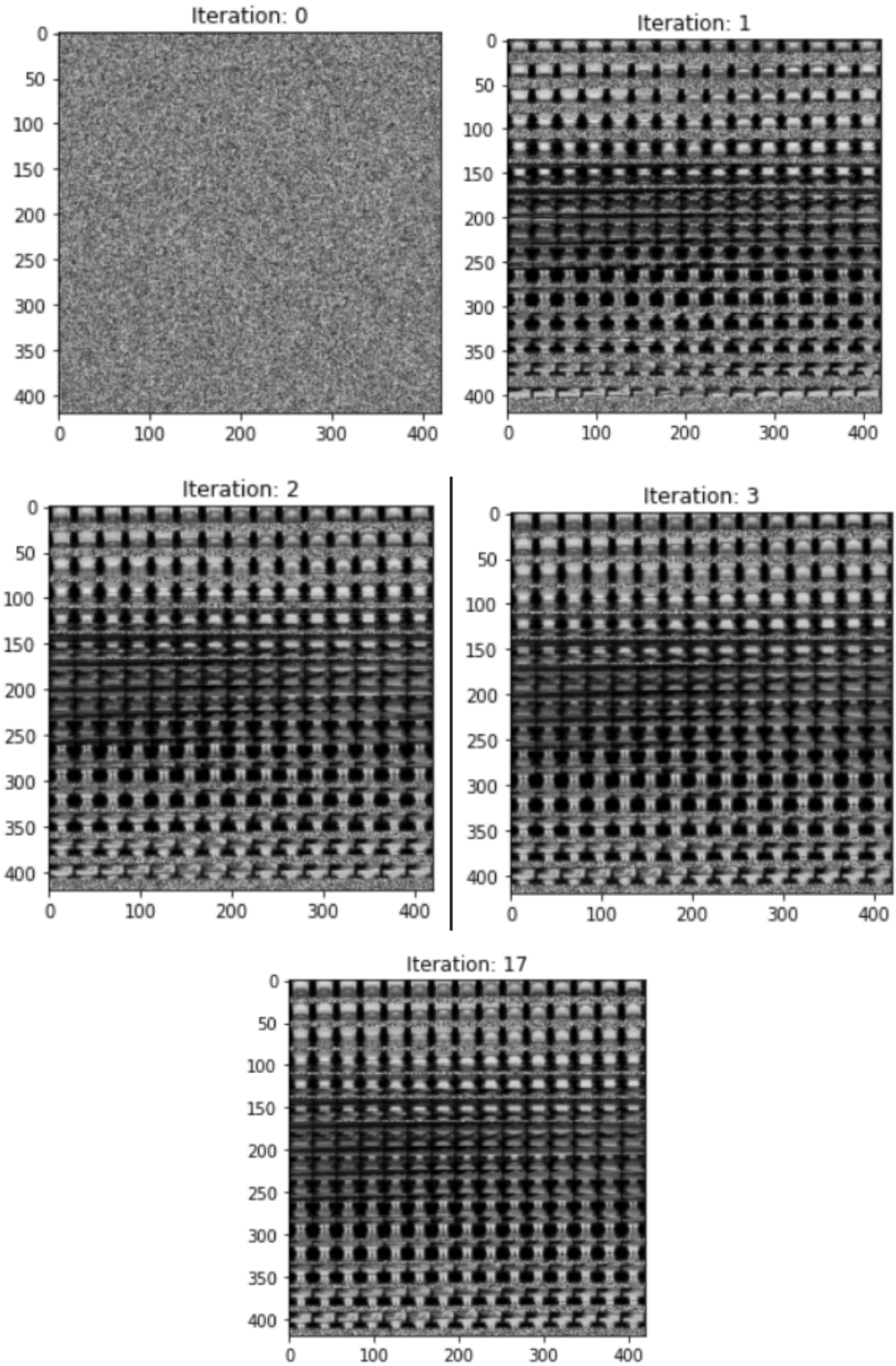
روند تغییر خوشه بندی برای همه کلاسترها در یک عکس 420*420 برای هر epoch آورده شده است.

برای آموزش شرط توقف را این گذاشته ایم که نرم تغییرات وزن های Wکمتر از 0.001 شود.

که بعد از ۱۷ epoch آموزش همگرا میشود.

در شکل ۵-۳ به ترتیب از چپ به راست و بالا به پایین روند تغییر خوشه بندی ها را برای ۴ epoch اول و در نهایت ۱۷ام میبینیم.

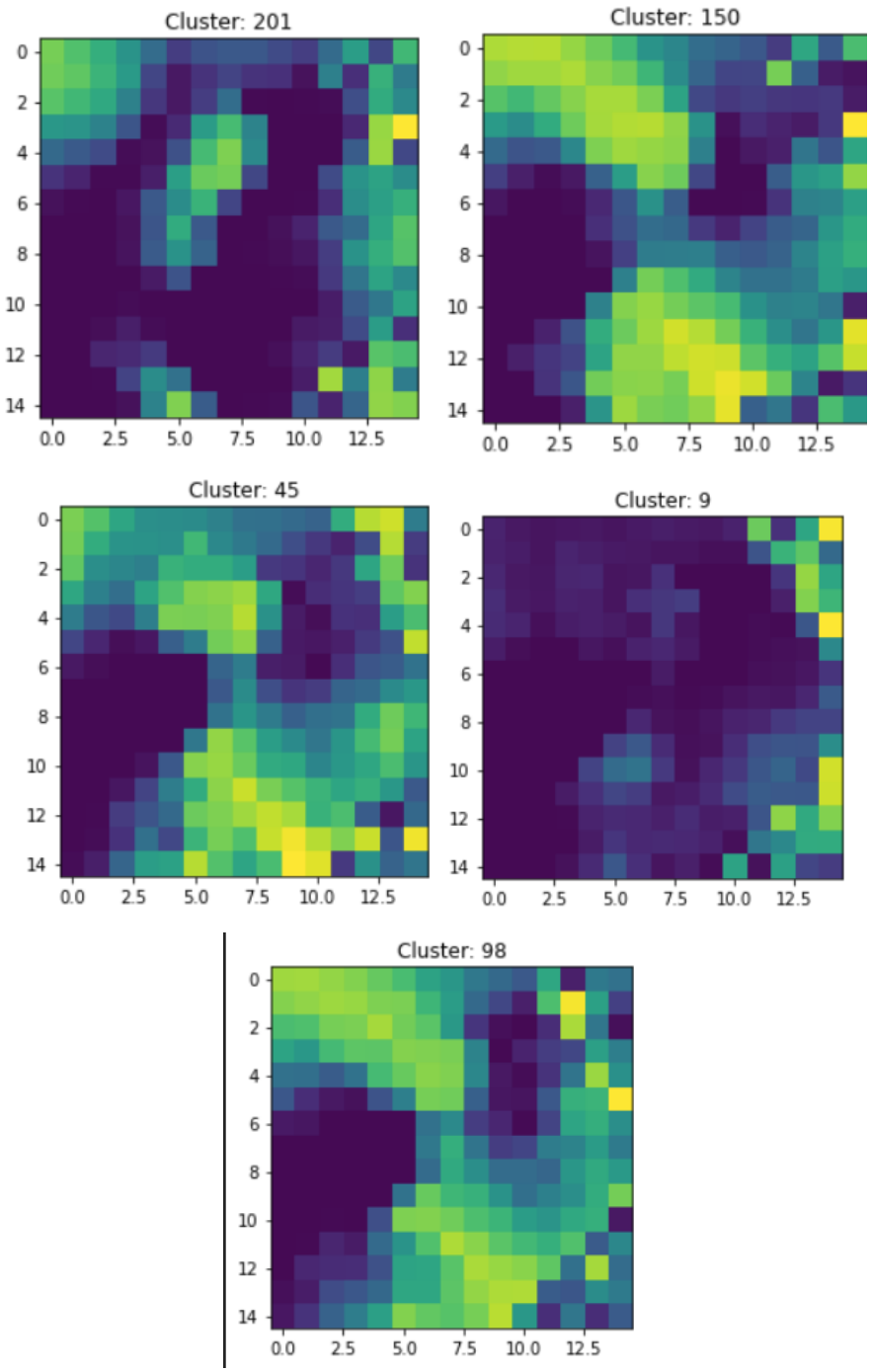
همان طور که مشاهده میشود وزن های کلاسترها از نویز توانسته اند در همان کلاستر اول مقادیر مناسبی بگیرند و در نهایت در ۱۷epochام وزن های مشخص تری مشاهده میشود.



شکل ۵-۳ نمودار تغییرات وزن های خوشه ها

د)

در شکل ۵-۴ وزن های ۵ خوشه چگال یعنی خوشه هایی با بیشترین تعداد عنصر آورده شده است.



شکل ۵-۴ وزن خوشه های چگال