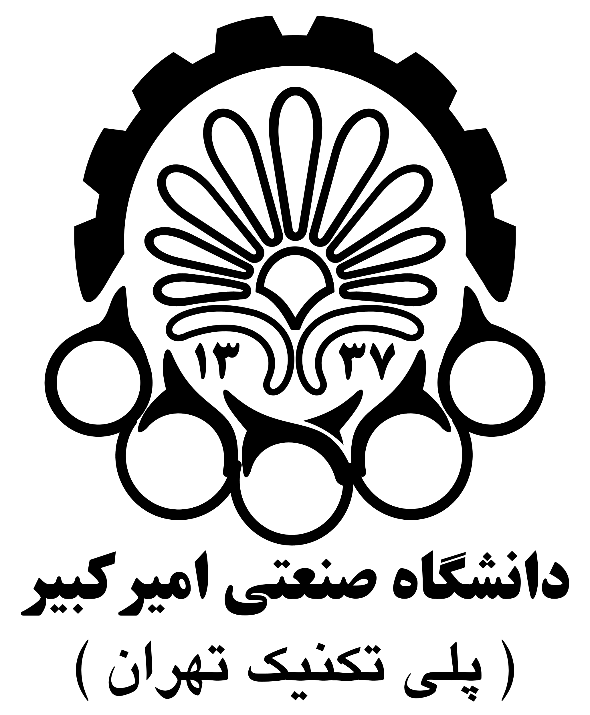
****

**دانشکده مهندسی کامپیوتر**

**تمر ین سوم درس شبکه های عصبی**

**دکتر صفابخش**

**غلامرضا دار 400131018**

**بهار 1401**

فهرست مطالب

[سوال 1) 3](#_Toc101877403)

[سوال 2) 4](#_Toc101877404)

[سوال 3) 5](#_Toc101877405)

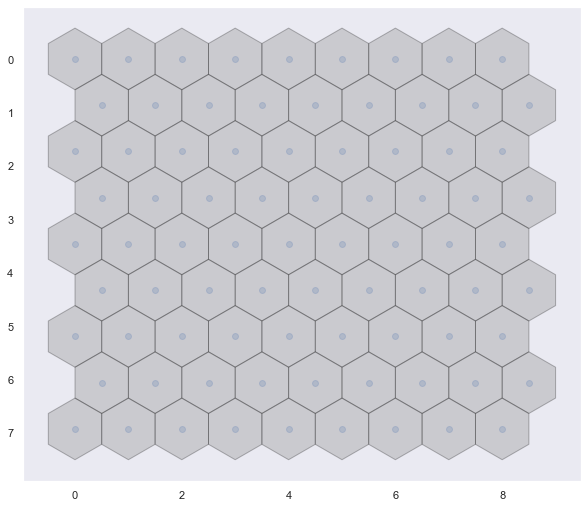
[سوال 4) 6](#_Toc101877406)

[سوال 5) 7](#_Toc101877407)

[سوال 6) 15](#_Toc101877408)

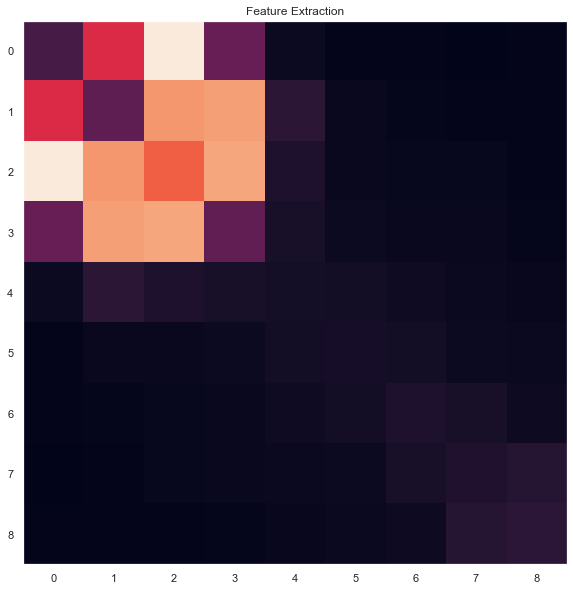
## سوال 1)

همانطور که می­دانیم، دو کار اصلی که شبکه­های خودسازماندهِ کوهونن انجام می­دهند، کاهش بُعد و خوشه بندی است. فرض کنید هر بردار در مجموعه داده ورودی دارای 561 بعد باشد (به عنوان مثال در همین تمرین) و یک نقشه 9 در 9 را برای این شبکه در نظر گرفته­ایم.



شکل 1- نقشه 9 در 9

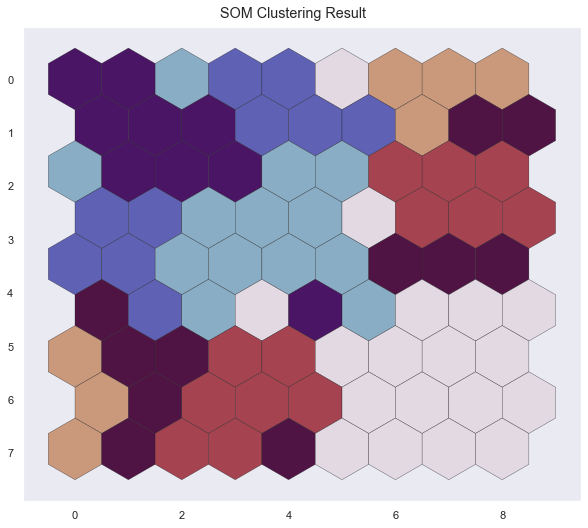
هر نورون در این نقشه­ی 9 در 9، یک وزن به ابعاد 561 دارد. در طی آموزش شبکه، هر بردار از مجموعه داده ورودی به یک نورون در شبکه، که وزن آن با مقدار نورون شباهت بیشتری دارد، نسبت داده می­شود. پس از اتمام آموزش، به ازای هر داده ورودی، میتوانیم فاصله­ی آن داده تا تمام نورون­های شبکه را محاسبه و به عنوان داده ورودی جدید استفاده کنیم. واضح است که این داده ورودی جدید دارای ابعاد 9 در 9 (ابعاد نقشه) است. لازم به ذکر است که معمولا به جای استفاده از فاصله داده تا هر نورون، از معکوس فاصله استفاده می‎شود تا نورون­های نزدیک­تر وزن بیشتری بگیرند.



شکل 2- داده ورودی تبدیل شده به ابعاد 9 در 9

## سوال 2)

از بین نورون­های موجود در نقشه، نورون هایی که هیچگاه به عنوان نورون "**برنده**" انتخاب نمی­شوند، نورون مرده نام دارند. این نورون ها باعث هدر­ رفتن فضای بازنمایی[[1]](#footnote-1) می­شوند زیرا هیچ داده­ای هیچ­وقت به این نورون­ها نسبت داده نمی­شود و موقعی که از شبکه برای کاهش بعد استفاده میکنیم، دارای شبکه ای با اندازه­ی موثر کمتری هستیم.



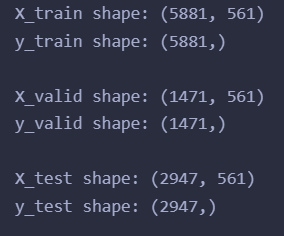
شکل 3- نورون­های مرده با رنگ سفید مشخص شده اند

مشکل دیگری که این نورون­ها دارند این است که با توجه به الگوریتم خاصِ آموزش، نورون­های مرده به ندرت به زندگی باز­می­گردند. به این معنی که اگر یک نورون مرد(!)، داده های جدید به نورون­های دیگر با وزن مناسب­تر(نزدیک­تر) نسبت داده می­شوند و در مراحل بعد، همین نورون­ها هستند که وزنشان بهبود می­یابد و نورون­های مرده همچنان مرده باقی می­مانند. مخصوصا اگر در همسایگی نورون­های مرده دیگر باشند.

از دلایل اصلی بوجود آمدن نورون­های مرده، مقداردهی اولیه اشتباه است. بنابراین یکی از راه­حل­های کاهش نورون­های مرده، مقداردهی اولیه بهتر مانند **مقداردهی اولیه تصادفی** است. راه دیگری که برای کاهش نورون­های مرده وجود دارد، این است که **به صورت رندوم هر از گاهی تعدادی از نورون­های مرده را به طور مصنوعی به سمت داده­ها ببریم.** با این کار انتظار می­رود که تعدادی از نورون­هایی مرده، که در همسایگی نورون­های برنده دیگر نیز قرار ندارند، شانس دوباره ای داشته باشند و با تغییر وزنشان به نورون برنده تبدیل شوند. همانطور که مشخص است، این کار باعث می­شود در مراحل بعدی نورون­های مرده­ای که اطراف این نورون مرده قرار داشتند نیز، به روند آموزش بازگردند(به دلیل آپدیت وزن نورون­های همسایه در هر مرحله).

## سوال 3)

مجموعه داده شامل حدوداً 7500 داده آموزش و 3000 داده آزمون است. این داده­ها به کمک کتابخانه Pandas فراخوانی شدند و پس از انجام مرحله Shuffling، به سه دسته **آموزش**، **اعتبارسنجی** و **آزمون** با ابعاد زیر تقسیم شدند.



شکل 4- تقسیم بندی مجموعه داده به دسته های آموزش، اعتبارسنجی و آزمون

همچنین در ادامه این گزارش، از رنگ­های ذکر شده در تصویر زیر برای نمایش کلاس­های مختلف استفاده می­شود.



شکل 5- رنگ­های متناظر با کلاس­های مختلف

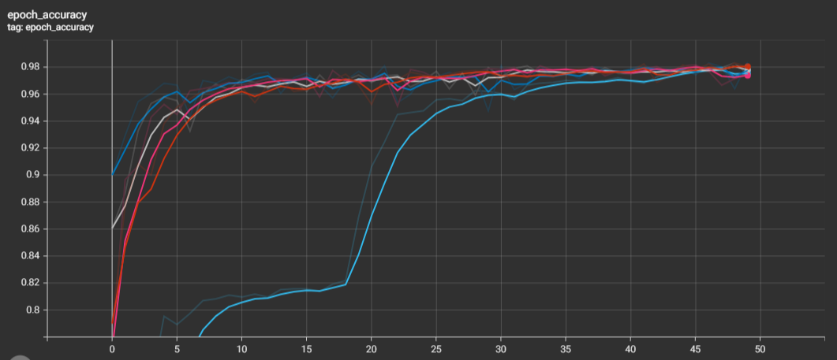
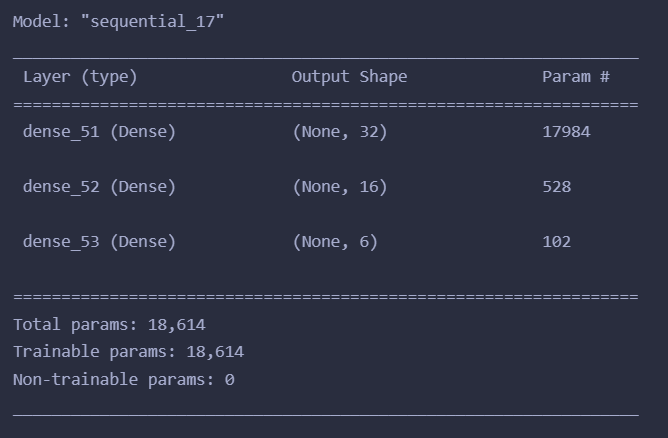
## سوال 4)

در این مرحله مشابه تمرین­ سری 2، با استفاده از کتابخانه Tensor Flow و با کمک Keras به پیاده­سازی و آزمایش چندین شبکه MLP با ابعاد و ویژگی­های مختلف می­پردازیم. این مدل­ها را با داده­ی آموزش، آموزش می­دهیم و در نهایت با داده­­ی آزمون، بررسی می­کنیم.

در این بخش می­توانید تعدادی از مدل های آزمایش­شده را مشاهده کنید.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Accuracy** | **Epoch** | **Learning Rate** | **Model** |
| **0.954** | **50** | **0.001** | **32 (sigmoid) – 16(sigmoid) – 6(softmax)** |
| 0.952 | 50 | 0.001 | 128 (sigmoid) – 64(sigmoid) – 6(softmax) |
| 0.949 | 50 | 0.001 | 64 (sigmoid) – 32(sigmoid) – 6(softmax) |
| 0.9444 | 20 | 0.01 | 256 (sigmoid) – 256(sigmoid) – 6(softmax) |
| 0.09345 | 20 | 0.01 | 1024 (sigmoid) – 512(sigmoid) – 6(softmax) |

در نهایت بهترین نتیجه را از مدلی با مشخصات زیر گرفتیم.



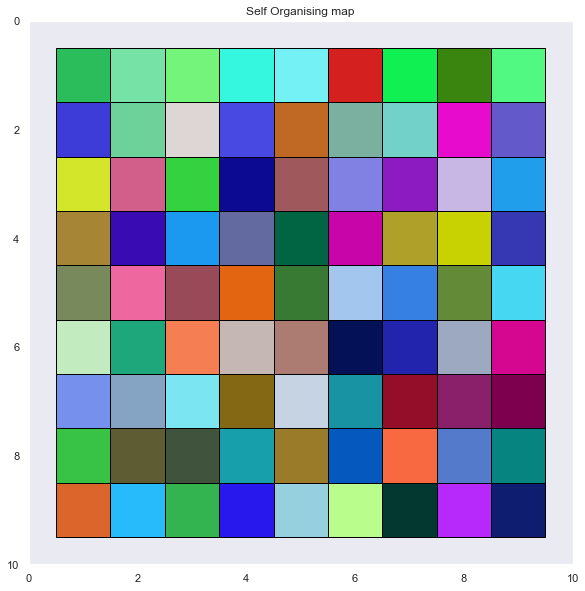
بنابراین بهترین دقت دسته بندی، با استفاده از داده­ی خام و بدون استفاده از شبکه­ی خودسازمانده کوهونن برابر با **95%** بود.

## سوال 5)

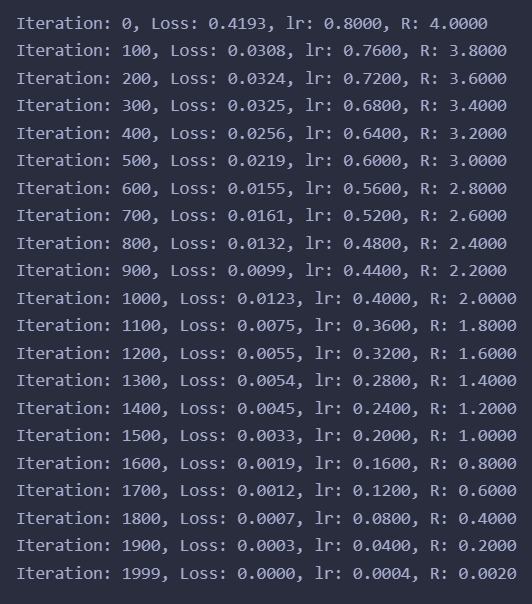
در این مرحله به پیاده­سازی شبکه خودسازمانده کوهونن می­پردازیم و سپس تعدادی تابع جهت مصورسازی[[2]](#footnote-2) این شبکه تولید میکنیم.

قبل از اینکه این شبکه را برروی مجموعه داده اصلی سوال که ابعاد بالایی دارد اجرا کنیم، با یک مجموعه داده ساده تر از صحت کارکرد شبکه اطمینان حاصل میکنیم. مجموعه ای که برای این بخش درنظر گرفته­ایم، مجموعه­ای از 200 بردار با مقادیر تصادفی در فضای RGB است. از آنجایی که هر بردار را میتوان با یک رنگ نمایش داد، به راحتی میتوانیم عملکرد این شبکه را مشاهده کنیم. هر نورون در این شبکه دارای یک وزن با ابعاد 1 در 3 خواهد بود که در طی آموزش سعی می­شود این بردار وزن، به رنگ­های موجود در مجموعه داده نزدیک شود.

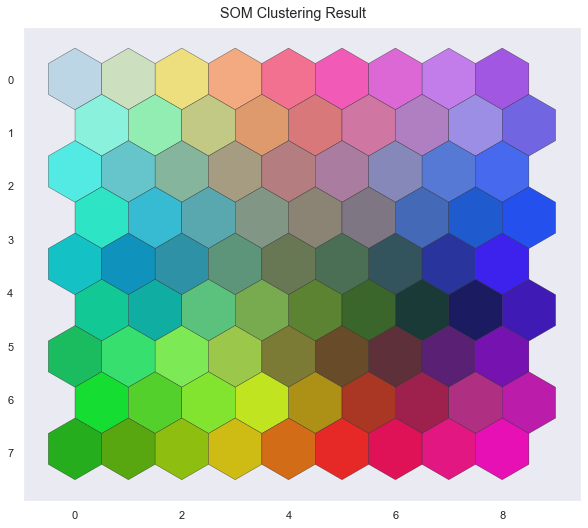
در تصویر زیر وزن­های اولیه نورو­ن­ها در نقشه­ی 9 در 9 را مشاهده میکنید.



شکل 6- وزن­های اولیه شبکه خودسازمانده کوهونن در مثال رنگ­ها



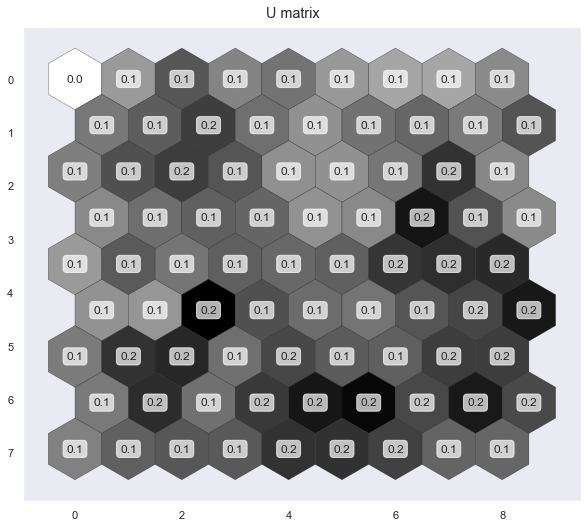
پس از آموزش مدل به اندازه­ی 2000 مرحله، شبکه ما توانست رنگ­های مشابه را به نورون­های نزدیک به هم نسبت دهد.



شکل 7- نتیجه آموزش شبکه خودسازمانده کوهونن بر روی مجموعه داده رنگ­ها

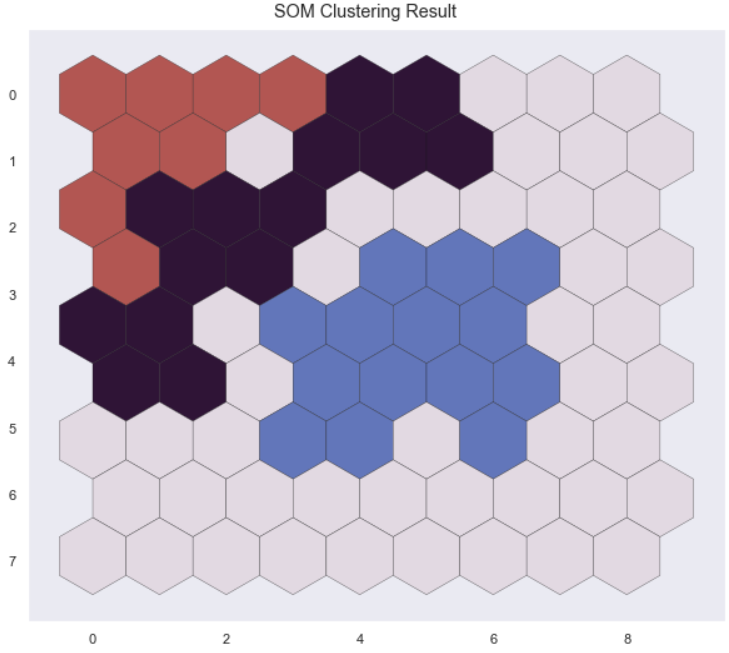
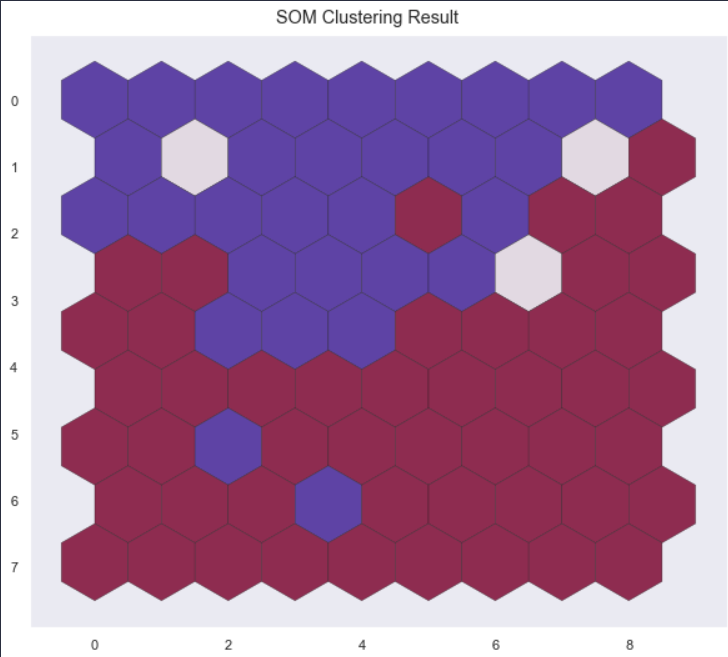
لازم به ذکر است که فضای رنگی RGB بهترین گزینه برای مواقعی که به دنبال فاصله اقلیدسی رنگ­های مختلف هستیم نیست. در این مثال بهتر بود رنگ­ها را در یک فضای رنگی مناسب تر مانند L\*a\*b یا YUV نمایش می­دادیم. اما با این حال نتیجه قابل قبولی گرفتیم به این معنی که شبکه به خوبی کار می­کند.

همچنین نمودار U-Matrix متناظر را در تصویر زیر مشاهده می­کنید.



شکل 8- نمایش U-Matrix متناظر با مثال رنگ­ها

در ادامه بررسی هایی که انجام دادیم، این شبکه را برروی مجموعه داده­های Iris, US Congress نیز آزمایش کردیم. تصویر سمت راست نتیجه خوشه­بندی روی مجموعه داده US Congress است که جمهوری­خواه­ها از دموکرات ها جدا شده اند. همچنین تصویر سمت چپ نتیجه خوشه­بندی سه نوع گل در مجموعه داده Iris است. وقت زیادی بر­روی بهینه کردن پارامترها برای این دو مثال صرف نشد و صرفا از نتیجه این مثال­ها برای اطمینان از صحت کارکرد شبکه استفاده شد.

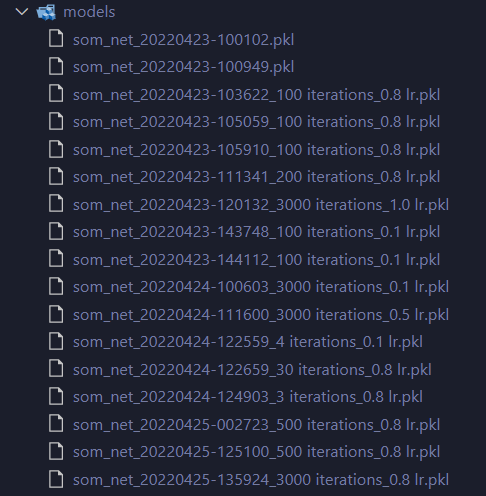


در نهایت نوبت به مجموعه داده اصلی سوال رسید. برای این مثال، از یک نقشه **9 در 9** استفاده شد. طبق آزمایش هایی که انجام دادیم، نقشه کوچکتر تعداد نورون های مرده را کاهش می­داد ولی دقت کافی را نداشت و تعدادی از کلاس­ها هم­پوشانی زیادی داشتند.

در این سوال علاوه بر نمودارهای خواسته شده توسط سوال، یک نمودار "اکثریت آرا" نیز رسم شد که کمک به سزایی به تحلیل نتیجه مدل میکند. برای رسم این نمودار، به ازای هر نورون در نقشه، رنگ متناظر با فراوان­ترین کلاس از بین داده های ورودی نسبت داده شده به آن نورون را استفاده کردیم. بار دیگر جدول هر کلاس و رنگ متناظر آن را مشاهده می­کنیم(کلاس **صفر** نشان دهنده **نورون های مرده** است).

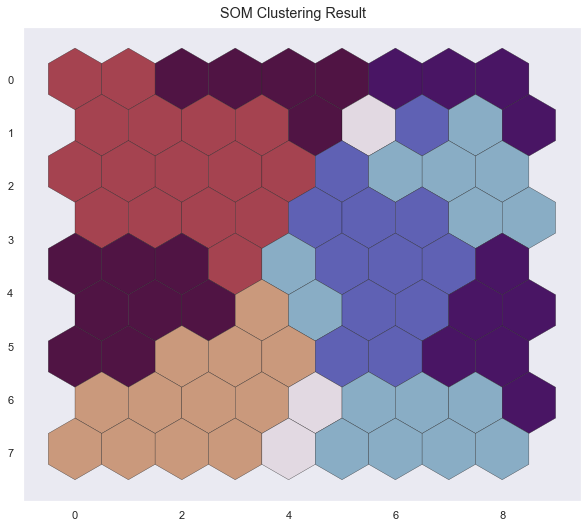


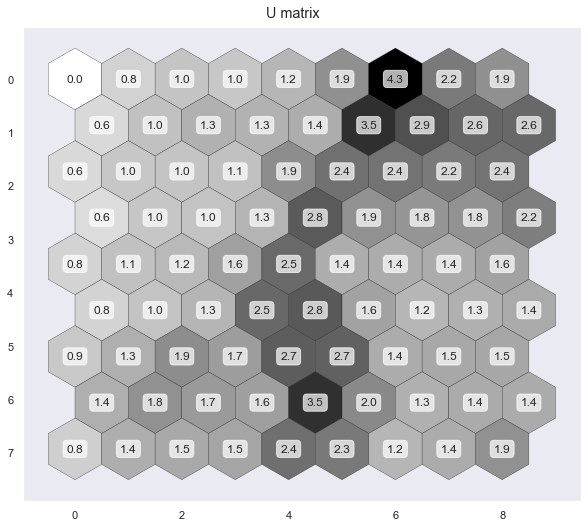
از آنجایی که آموزش با تمام داده­های آموزش وقت زیادی می­گرفت، تصمیم بر آن شد که از بخشی (حدود 20 درصد) از داده های آموزش برای آموزش دادن شبکه استفاده کنیم. این کار باعث شد بتوانیم تعداد آزمایش­های بیشتری انجام دهیم و همچنین آموزش را تا Iteration های بیشتری ادامه دهیم. در طی آموزش این شبکه متوجه شدیم که ادامه دادن آموزش تا Iteration های بیشتر تاثیر خوبی بر کاهش تعداد نورون­های مرده دارد. همچنین در این آزمایش­ها شعاع همسایگی و نرخ یادگیری را به طور خطی کاهش دادیم. در اکثر آزمایش ها شعاع همسایگی اولیه برابر با نصف اندازه نقشه است و نرخ یادگیری اولیه برابر با عدد 0.8 می­باشد.

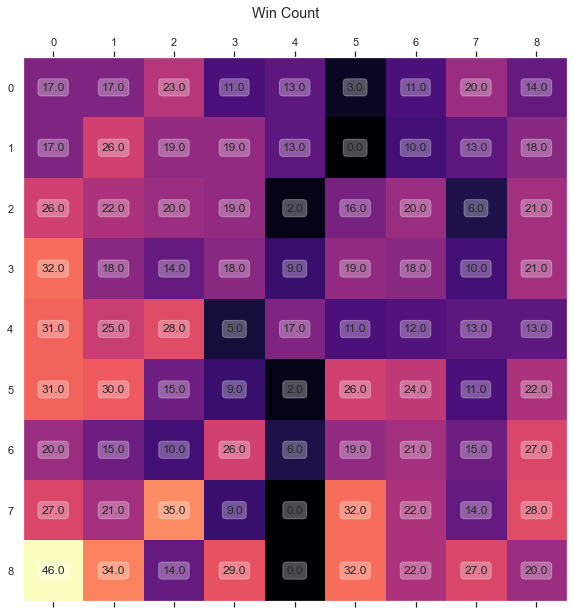


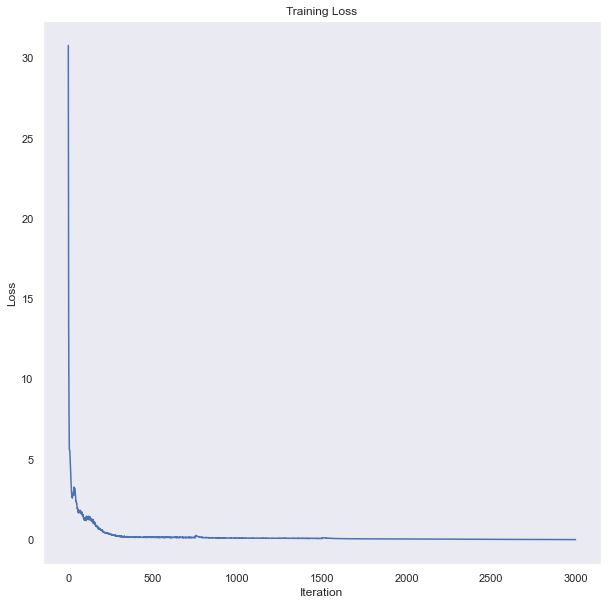
شکل 9- تعدادی از مدل­های آموزش داده شده با پارامتر­های مختلف(ذخیره شده جهت استفاده مجدد)

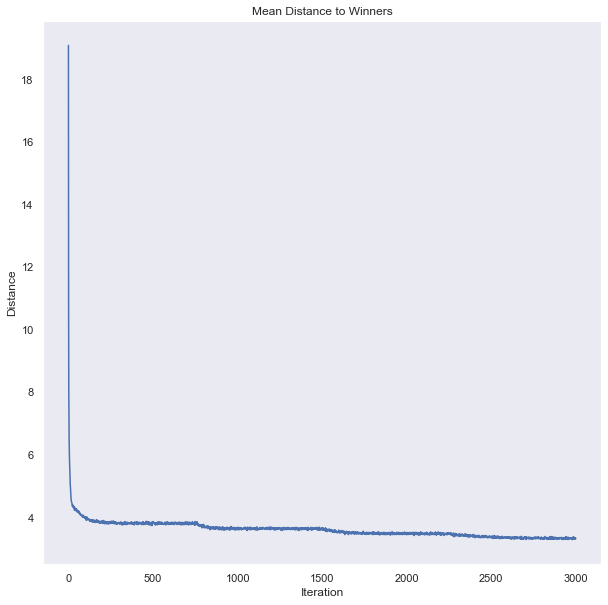
در این بخش نمودارهای مختلفی که صورت سوال از ما خواسته به علاوه نمودار اکثر آرا را برای بهترین مدل، نمایش میدهیم.

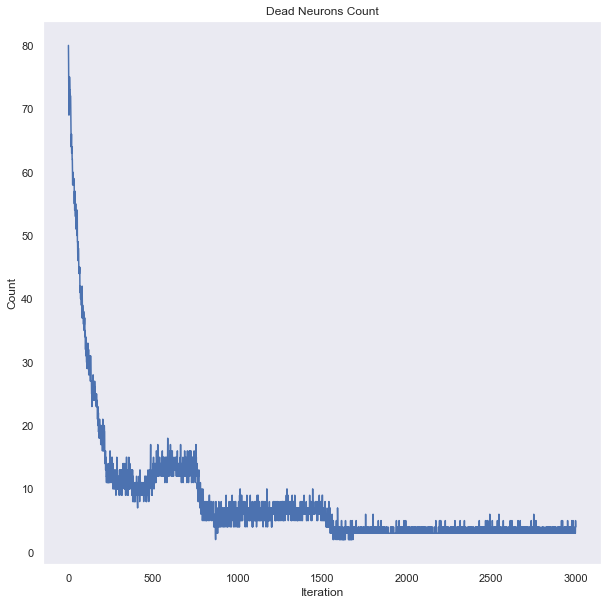


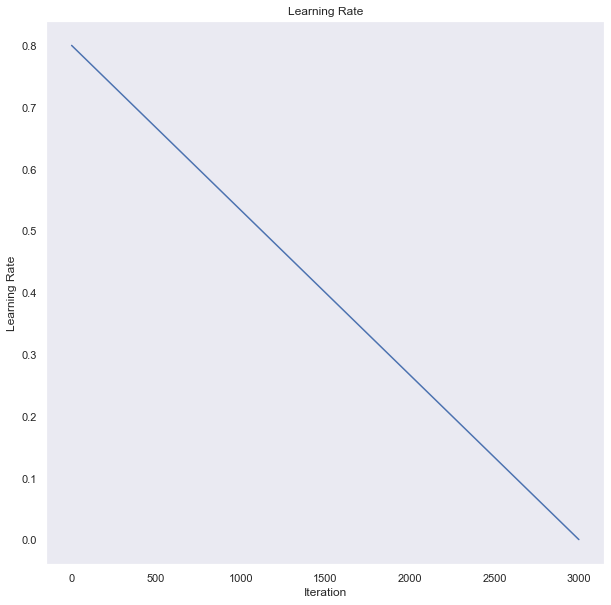


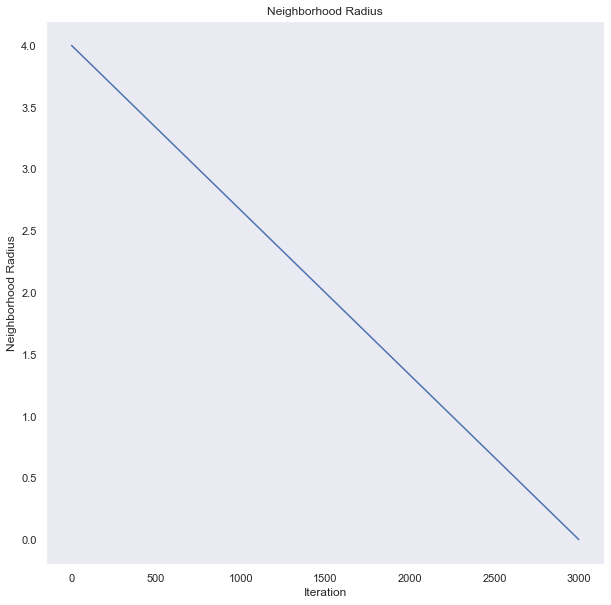












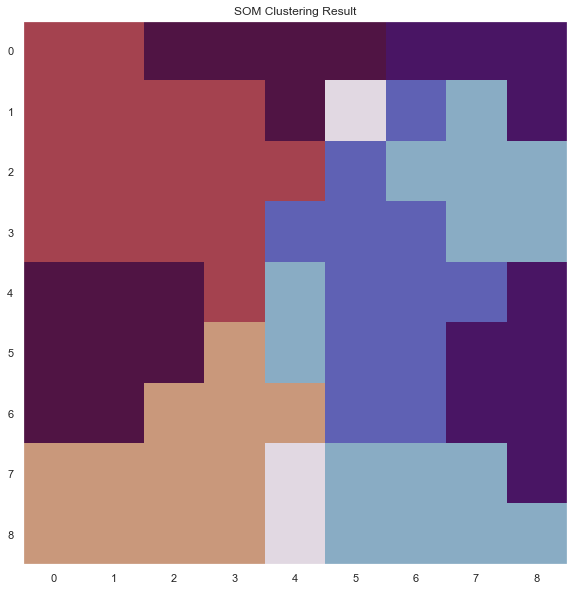
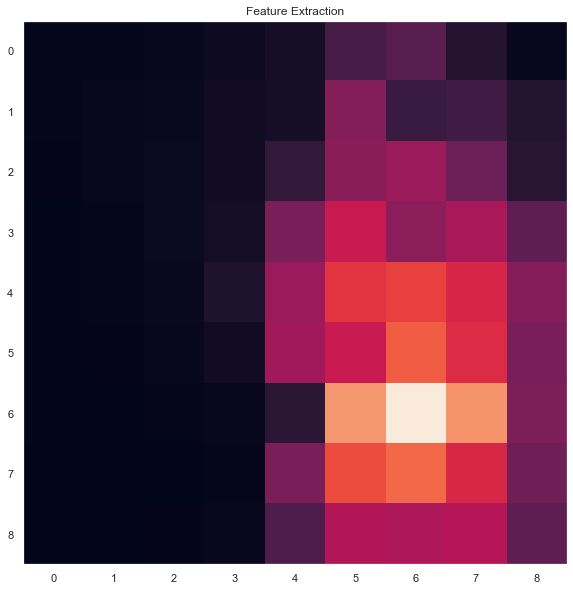
## سوال 6)

پس از آموزش دادن مدل، نوبت به استخراج ویژگی رسید. روند کلی استخراج ویژگی به این شکل است:

1. یک داده ورودی انتخاب میکنیم
2. یک نقشه، هم اندازه با نقشه SOM تولید میکنیم
3. فاصله مقدار این داده را با وزن تمامی نورون ها در نقشه SOM محاسبه میکنیم و در سلول متناظر در نقشه جدید قرار میدهیم.
4. چون میخواهیم اعداد بزرگ­تر نشان دهنده شباهت بیشتری باشند(نه فاصله بیشتر) به جای قرار دادن فاصله در هر سلول، معکوس فاصله را قرار میدهیم.

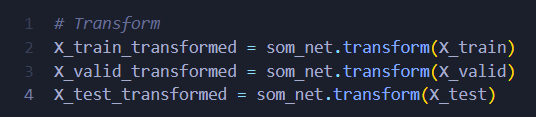
به عنوان مثال برای یکی از داده های ورودی که کلاس متناظر آن **2** (بالارفتن از پله) بود این ویژگی را استخراج کردیم و نقشه زیر را نتیجه گرفتیم.





همانطور که می­بینید، قسمت­های روشن در نقشه جدید متناظر با نورون­هایی هستند که بیشتر از بقیه نورون­ها به کلاس متناظر با این داده ورودی خاص نزدیک بوده اند.

به عبارتی ما توانستیم داده ورودی با ابعاد 561 را به یک بردار 81 بعدی کاهش دهیم و در عین حال ارتباط بین کلاس متناظر را حفظ کرده ایم. در ادامه این کار را به ازای کل داده­های موجود در مجموعه داده انجام می­دهیم و شبکه MLP سوال 4 را این­بار با این داده های جدید آموزش می­دهیم.

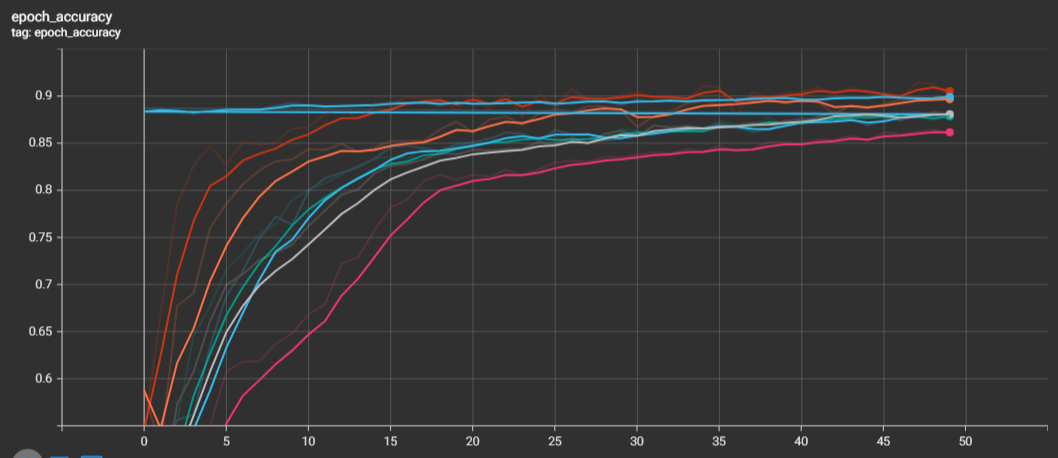
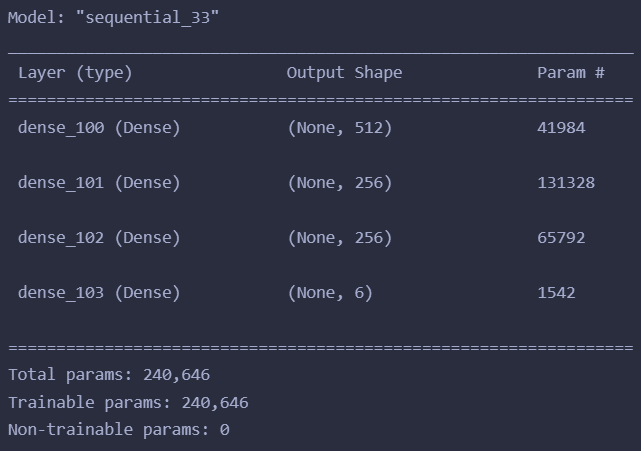


شکل 10- انتقال داده­ها از فضای 561 بعدی به 81 بعدی

در ادامه تعدادی از آزمایش­های انجام شده برای این بخش را نمایش می­دهیم.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Accuracy** | **Epoch** | **Learning Rate** | **Model** |
| **0.878** | **50** | **0.001** | **512 (sigmoid) – 256x2(sigmoid) – 6(softmax)** |
| 0.870 | 50 | 0.001 | 32 (sigmoid) – 16(sigmoid) – 6(softmax) |
| 0.867 | 50 | 0.001 | 128 (sigmoid) – 64(sigmoid) – 6(softmax) |
| 0.852 | 20 | 0.001 | 64 (sigmoid) – 32(sigmoid) – 6(softmax) |
| 0.840 | 20 | 0.001 | 12 (sigmoid) – 8(sigmoid) – 6(softmax) |

در نهایت بهترین نتیجه را از مدلی با مشخصات زیر گرفتیم.



**لازم به ذکر است که این نتایج قطعا بهترین نتایج نیستند و با آزمایش­های بیشتر و با ایده­های مختلفِ دیگر می­توان نتیجه را تا حدی بهبود داد. همچنین به دلیل مشکل زمان آموزش، شبکه را بر روی بخشی از داده ها آموزش دادیم.**

**نتیجه­گیری:** با وجود اینکه داده ها را از 561 بعد به 81 بعد کاهش دادیم، دقت مدل در تشخیص کاهش بسیار کمی داشت. این کاهش بعد باعث سریع تر شدن روند آموزش و اجرای مدل شد و در نتیجه با استفاده از خاصیت خوشه­بندی SOM توانستیم به شکل زیبایی داده­ها را در ابعاد کمتر مشاهده کنیم.

پایان

1. Representation [↑](#footnote-ref-1)
2. Visualization [↑](#footnote-ref-2)