

به نام خدا

دانشگاه صنعتی امیرکبیر
دانشکده مهندسی کامپیوتر

پاسخ تمرین سری سوم شبکه‌های عصبی

استاد:

دکتر صفابخش

دانشجو:

حلیمه رحیمی

شماره دانشجویی:

۹۹۱۳۱۰۴۳

بهار ۱۴۰۰

۱- شبکه خودسازمانده کوهونن نوعی شبکه‌ی بدون نظارت است که در آن نورون‌ها برای برنده شدن رقابت می‌کنند. این شبکه از دو لایه ساخته شده که لایه‌ی اول ورودی و لایه‌ی دوم خروجی است. لایه‌ی دوم می‌تواند یک، دو یا سه بعدی باشد. هر یک از داده‌های ورودی با توجه به فاصله‌ی آن تا نورون (فاصله‌ی ویژگی‌هایش تا وزن‌های نورون) به یکی از نورون‌ها منتسب می‌شود و وزن نورون‌ها تا همسایگی مشخصی از آن (یا حتی فقط خود آن نورون) را بروز می‌کند. هر یک از ورودی‌ها به همه‌ی نورون‌های خروجی وصل اند. لازم به ذکر است تابع فعالیت نیز می‌تواند معیار انتخاب نورون برنده باشد.

بروزرسانی وزن‌ها در آن طبق فرمول زیر اتفاق می‌افتد:

$$w_j(t) = w_j(t-1) + \beta(t) NS(d, t) [x(t) - w_j(t-1)]$$

که در آن $w_j(t)$ وزن‌های نورون j ، $\beta(t)$ درجه یادگیری، $x(t)$ ورودی، و $NS(d, t)$ ماتریس ضریب همسایگی نرونی تا فاصله‌ی d در زمان t می‌باشد.

می‌توان درجه یادگیری و d را متغیر در نظر گرفت و با افزایش زمان کاهش داد. دو نوع تابع زیر از جمله توابعی است که می‌توان استفاده کرد. در پیاده سازی خود از تابع اول در هر یک استفاده کردم.

$$\begin{aligned} \beta_t &= \beta_0 \left(1 - \frac{t}{T}\right) & \sigma_t &= \sigma_0 \left(1 - \frac{t}{T}\right) \\ \beta_t &= \beta_0 \exp\left(-\frac{t}{T}\right) & \sigma_t &= \sigma_0 \exp\left(-\frac{t}{T}\right) \end{aligned}$$

همچنین برای تعیین ضریب همسایگی می‌توان از توابع مختلفی استفاده کرد؛ برای مثال در این تمرین از تابع گاوسی بهره برده ایم. از انواع دیگر آن می‌توان به تابع خطی و نمایی اشاره کرد. علاوه بر این می‌توان تا شعاع و شکل خاصی از اطراف نورون برنده را به عنوان همسایه در نظر گرفت؛ منظور از شکل خاص، به طور مثال در نظر گرفتن همسایه‌ها فقط در جهت‌های عمودی و افقی، یا در نظر گرفتن به شکل مربع یا دایره است.

برای تابع گاوسی از فرمول زیر استفاده می‌کنیم:

$$NS = \exp\left(\frac{-(x(t) - w_j(t))^2}{2\sigma_t^2}\right)$$

لازم به ذکر است با نسبت دادن یک نورون به داده‌ی ورودی، در نهایت ممکن است چندین داده به یک نورون نسبت داده شوند که آن نورون نماینده‌ی آنها خواهد بود. نورون‌ها سعی خواهند کرد داده‌های مشابه را جذب کنند. وقتی وزن‌های همسایه‌های نورون برنده را نیز آپدیت می‌کنیم، این مسئله قابل درک است که چطور اگر داده‌ها مشابه یکدیگر باشند نورون‌هایی که به آنها نسبت داده شده کنار هم خواهند افتاد.

۲- تفاوتی در آموزش برای خوشه‌بندی و کشف ویژگی وجود ندارد اما مسلماً به دلیل آنکه هدف در هر یک متفاوت است، با توجه به نوع داده‌هایی که داریم ممکن است نوع انتخاب پارامترها تغییر کند. نحوه‌ی تحلیل نیز مسلماً متفاوت خواهد بود و نوع نمایش نتیجه نیز در آنها متفاوت است. البته مشخص است که در کاهش ابعاد، تعداد نورون‌ها نباید از تعداد ویژگی‌های داده بیشتر باشد. برای خوشه‌بندی لازم است از ماتریس U استفاده کنیم. در این ماتریس فاصله‌ی نورون‌ها با نورون‌های همسایه مشخص خواهد بود

و می‌توان نورون‌هایی را که از یکدیگر فاصله‌ی کمی دارند یک خوشه در نظر گرفت. البته می‌توان با داشتن برچسب هر یک از داده‌ها ماتریس SOM را نیز برای خوشه‌بندی رسم کرد و از نحوه پراکندگی داده‌های هر کلاس مطلع شد. برای کاهش ابعاد می‌توان با استفاده از ماتریس SOM از آنچه ویژگی‌های حاصل ارائه می‌دهند آگاه شد. برای کاهش ابعاد می‌توان از میزان خلوص برای سنجش شبکه استفاده کرد و برای خوشه‌بندی نیز در صورت داشتن برچسب‌ها بهره‌گیری از آن مناسب است. در غیر این صورت می‌توان از فاصله هر داده به وزن نورون منتسب استفاده کرد و یا پس از خوشه‌بندی فاصله‌ی درون خوشه‌ای و برون خوشه‌ای را معیار سنجش قرار داد.

۳- پیش از هر چیز لازم است توضیحاتی را در مورد داده‌های ورودی بدهیم. این مجموعه داده شامل تصاویر اشخاص مختلف در حالات مختلف است بنابراین ممکن است شبکه، تصاویر هر فرد را بیشتر مشابه یکدیگر در نظر بگیرد (نه حالات مختلف). تنها دو کلاس نور از راست و نور از چپ، چنان تفاوتی از دیگر کلاس‌ها دارند (یک سمت تصویر سایه است) که تشخیصشان راحت‌تر از کلاس‌های دیگر است و برای کلاس‌های دیگر، تصاویر هر فرد به نورون‌های کنار یکدیگر منتسب خواهند بود.

داده‌ها را به سه گروه آموزش، اعتبارسنجی و آزمون تقسیم کردم و SOM را بر روی گروه اول آموزش دادم. دلیل این امر تنها هماهنگی با سوالات بعدی می‌باشد. چرا که برای دسته‌بندی اینطور در نظر داشتیم که تنها داده‌های آموزش را داریم و بنابراین کاهش ابعاد داده‌های اعتبارسنجی و تست را از طریق SOM آموزش دیده بر روی گروه آموزش بدست آوردم.

در صورت بروز رسانی فقط خود نورون برنده، داشتن تعداد نورون بیشتر از تعداد داده‌ها تنها موجب افزایش نورون‌های مرده می‌شد. دلیل آنچه می‌گویم این است که در بدترین حالت هر داده به یک نورون نسبت داده می‌شود. بیشتر بودن آن با توجه به نوع تعریف شبکه در این تمرین مشکلی ایجاد نخواهد کرد. بلکه موجب خواهد شد با آمدن داده‌ای جدید که کمی با داده‌های قبلی متفاوت است به یکی از نورون‌هایی نسبت داده شود که حین آموزش اصلاً به آن داده‌ای منتسب نشده است.

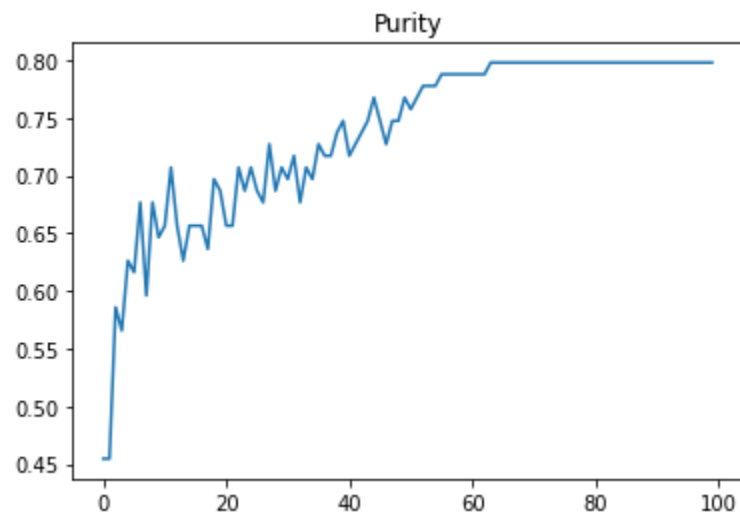
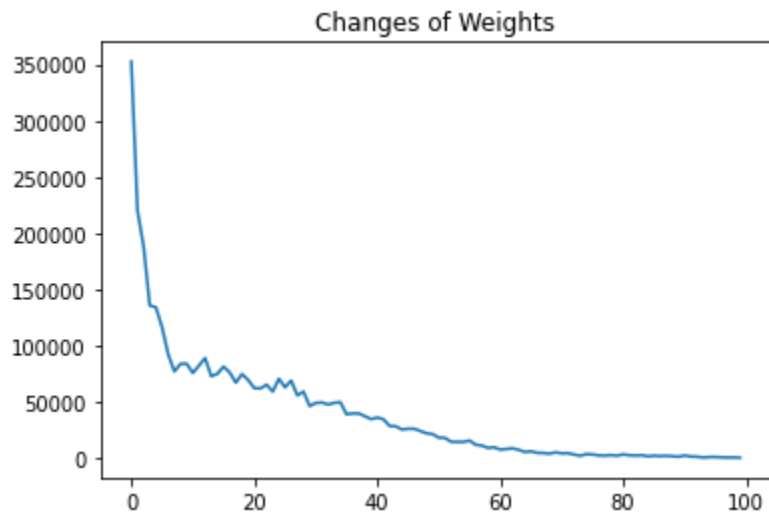
در اینجا همان شبکه‌ای را برای خوشه‌بندی در نظر گرفتم که برای کاهش ابعاد استفاده کردم. استفاده از ابعاد بالاتر زمان بیشتری برای آموزش می‌برد و ابعاد پایین‌تر مقدار خلوص کمتری را حاصل می‌دهد. خلوص پایین‌تر به همان دلیلی است که بالاتر بیان کردم؛ شبکه داده‌های مربوط به هر فرد را نزدیک‌تر به یکدیگر می‌داند تا حالات افراد مختلف. بنابراین با تعداد نورون کمتر، مسلماً بیشتر مواقع داده‌هایی که به یک نورون منتسب می‌شوند از کلاس یک حالت نیستند.

تعداد تکرار را ۱۰۰ در نظر گرفتیم. تمامی نتایج تمرین با همین تعداد نمایش داده شده است. درجه یادگیری را نیز در ابتدا یک قرار دادم و شعاع همسایگی را به عنوان سیگما به تابع گاوسی دادم. برای همه‌ی نورون‌ها ضریب همسایگی به دست خواهد آمد. تعداد نورون‌ها را نیز ۱۱ در ۱۱ و به عبارتی ۱۲۱ در نظر گرفتیم.

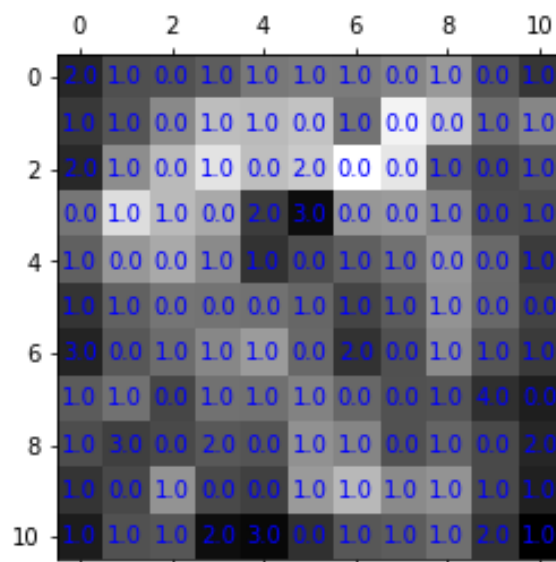
SOM را به شکل یک ماژول تعریف کردم و تحت عنوان `som_module` در کد خوشه‌بندی بهره‌بردم. سپس در همان کد، ویژگی‌های جدید را به شکل CSV به همراه برچسب‌هایشان ذخیره کردم تا در `mlp` از آنها استفاده کنم.

نتایج را در زیر مشاهده می‌کنید:

```
iter: 0 changes in weights: 352424.6029572759 purity: 0.4545454545454543
iter: 10 changes in weights: 75645.89714778806 purity: 0.6565656565656566
iter: 20 changes in weights: 62181.16683622475 purity: 0.6565656565656566
iter: 30 changes in weights: 48969.667021896596 purity: 0.6969696969696967
iter: 40 changes in weights: 36002.87364893574 purity: 0.7171717171717171
iter: 50 changes in weights: 18135.59943891592 purity: 0.7575757575757576
iter: 60 changes in weights: 7527.418247972612 purity: 0.7878787878787878
iter: 70 changes in weights: 4168.945111977284 purity: 0.7979797979797979
iter: 80 changes in weights: 3232.952573925521 purity: 0.7979797979797979
iter: 90 changes in weights: 2257.513048084371 purity: 0.7979797979797979
```



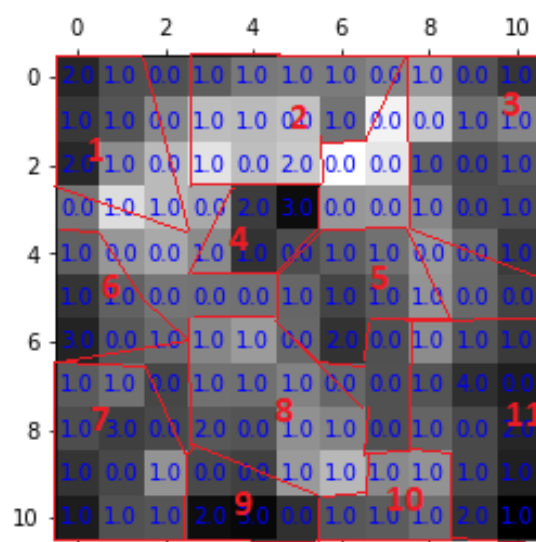
مشهود است که با تکرار بیشتر، مقدار خلوص بالاتر رفته و تغییرات وزن‌ها نیز کاهش یافته است.



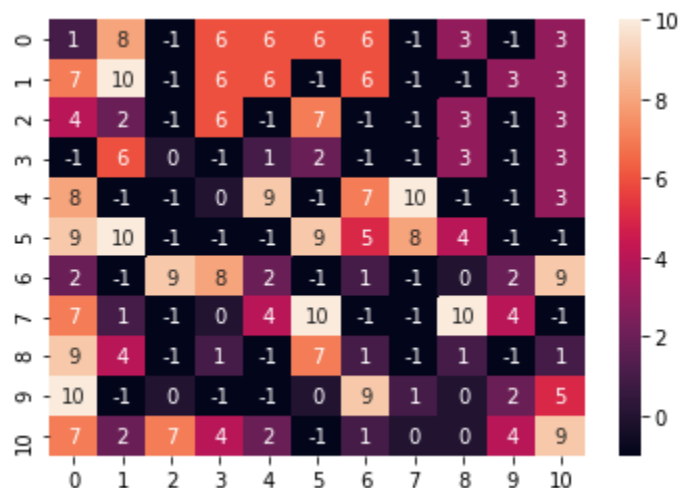
پس از صد تکرار، ماتریس U به شکل بالا حاصل می‌گردد. در این ماتریس فواصل نزدیکتر به یکدیگر با رنگ‌های تیره‌تر ظاهر می‌شود. از آن جهت که برای خوشه‌بندی در پی فاصله‌ی درون خوشه‌ای کمتر و فاصله‌ی برون خوشه‌ای بیشتر هستیم، می‌توان خانه‌های تیره‌ی نزدیک به هم را یک خوشه دانست که با تغییر رنگ خانه‌های حاشیه‌ای (در بیشتر مواقع کاملاً سفید) از یکدیگر جدا شده‌اند.

در تصویر زیر خوشه‌هایی را که در نظر گرفته‌ام را مشخص کرده‌ام. مسلماً با تکرار بیشتر این خوشه‌ها مجزاتر می‌شوند.

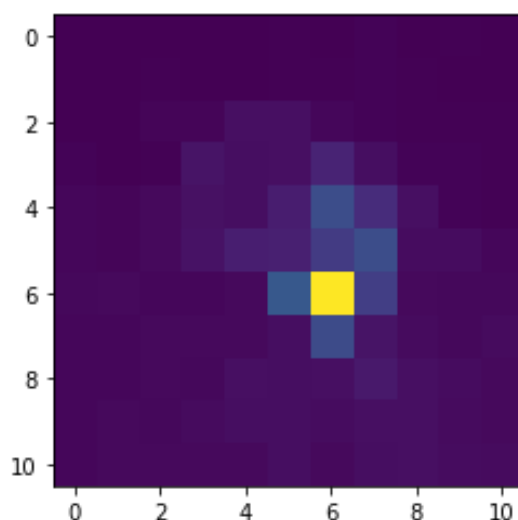
تعداد داده‌های منتسب به هر نورون در خانه‌ی مربوطه نوشته شده است.



۴- نورون‌هایی که هیچ داده‌ای به آنها نسبت داده نشده با 1- مشخص‌اند. باقی نورون‌ها با برجسب کلاس خود مشخص شده‌اند. همانطور که پیش‌بینی می‌شد کلاس ۳ و ۶ که مربوط به نور از چپ و نور از راست است، به خوبی دور هم جمع شده‌اند و در باقی خوشه‌ها می‌توان انواع کلاس‌ها را کنار یکدیگر مشاهده کرد. علت را همانی که بالاتر بیان کردم می‌دانم.



نتیجه کاهش بعد را برای یک داده از کلاس ۱ در زیر می‌بینید که در آن بیشترین مقدار مرتبط با نورونی است که نماینده‌ی کلاس ۱ بوده و در نقشه SOM مشخص است.



۵- برای مقایسه تصمیم گرفتیم هر دو شبکه و تعداد تکرار را یکسان در نظر بگیریم. هر دو نوع داده را استاندارد کردم. شبکه دو لایه‌ی مخفی ۱۲۸ و ۳۲ نورونی با تابع فعالیت Relu و لایه‌ی خروجی با تابع فعالیت Softmax دارد.

نتایج برای حالت بدون کاهش بعد به صورت زیر است.

بیشترین مقدار دقت برای آموزش:

Epoch 23/25
4/4 [=====] - 0s 74ms/step - loss: 1.0361 - accuracy: 0.6381 - val_loss: 10.9658 - val_accuracy: 0.2045

بیشترین دقت برای اعتبارسنجی (به شرط آنکه مقدار دقت آموزش نیز بالاتر از باقی مواردی باشد که اعتبارسنجی بیشترین دقت را بدست آورده):

```
Epoch 10/25
4/4 [=====] - 0s 76ms/step - loss: 1.9571 - accuracy: 0.5014 - val_loss: 13.0016 - val_accuracy: 0.2500
```

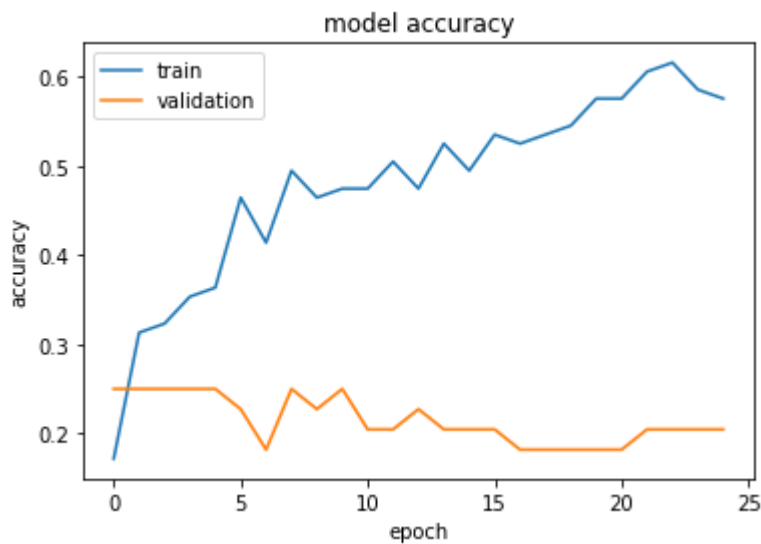
دقت بعد از ۲۵ تکرار:

```
Epoch 25/25
4/4 [=====] - 0s 75ms/step - loss: 1.2409 - accuracy: 0.5636 - val_loss: 11.0494 - val_accuracy: 0.2045
```

ماتریس درهم‌ریختگی به همراه دقت مجموعه تست:

```
[[1 0 0 0 0 0 0 0 0 0 0 1]
 [1 1 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 1]
 [0 0 0 2 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 1 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 1]
 [0 0 0 0 0 0 2 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 2]
 [1 0 0 0 0 0 0 0 0 1 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 1]
 [1 0 0 0 0 0 0 0 0 0 1 0]]
0.3181818181818182
```

نمودار دقت مدل در هر تکرار:



نتایج برای حالت با کاهش بعد را در ادامه می‌بینید.

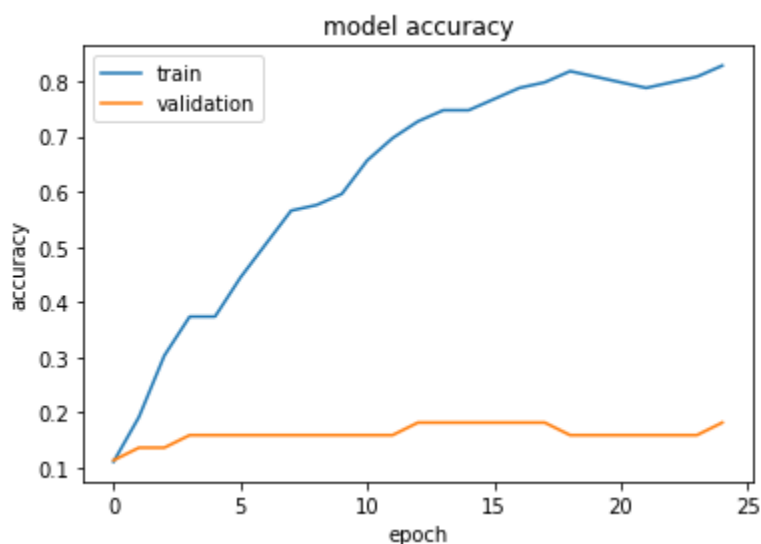
بیشترین مقدار دقت برای آموزش و بیشترین دقت برای اعتبارسنجی در تکرار ۲۵ داده است:

Epoch 25/25
4/4 [=====] - 0s 16ms/step - loss: 0.7412 - accuracy: 0.8324 - val_loss: 2.2393 - val_accuracy: 0.1818

ماتریس درهم‌ریختگی به همراه دقت مجموعه تست:

```
[[1 0 1 0 0 0 0 0 0 0 0]
 [1 0 1 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 1 0 0]
 [0 0 0 2 0 0 0 0 0 0 0]
 [1 0 1 0 0 0 0 0 0 0 0]
 [0 0 1 0 1 0 0 0 0 0 0]
 [0 0 0 0 0 0 2 0 0 0 0]
 [0 0 1 0 0 0 0 1 0 1 0]
 [1 0 1 0 0 0 0 0 0 0 0]
 [1 0 1 0 0 0 0 0 0 0 0]
 [1 0 1 0 0 0 0 0 0 0 0]]
0.2727272727272727
```

نمودار دقت مدل در هر تکرار:



از نتایج مشخص است که SOM باعث بهبود دقت گروه آموزش شده است. البته احتمال بر این است که به دلیل آموزش ندادن SOM بر روی داده‌های اعتبارسنجی و آزمون، کاهش ابعاد این داده‌ها به خوبی گروه آموزش اتفاق نیفتاده و در نتیجه در هنگام آموزش MLP تاثیر داشته و مانع از کسب دقتی به خوبی شبکه‌ی آموزش یافته بر روی خود تصاویر شده است. مشهود است که در هر دو حالت شبکه دچار بیش برآزش شده است.

لازم به ذکر است که شبکه بر روی داده‌های با کاهش بعد همانطور که انتظار می‌رود سریعتر عمل می‌کند.