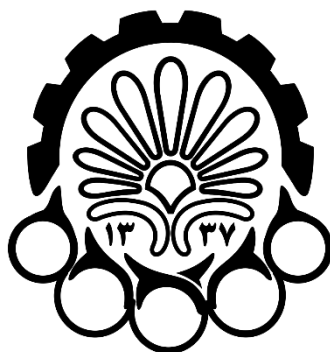


به نام خدا



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

تمرین درس شبکه‌های عصبی-سری سوم

فردین آیار

شماره دانشجویی: ۹۹۱۳۱۰۴۰

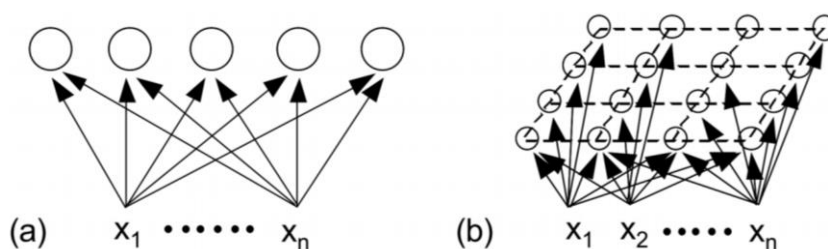
استاد: دکتر صفابخش

دانشکده کامپیوتر- زمستان ۹۹

۱) شبکه خودسازمانده کوهونن (SOM) شامل یک لایه ورودی و یک لایه خروجی می‌باشد. چپش لایه‌های خروجی می‌تواند یک، دو یا سه بعدی باشد. در این شبکه، نرونی که وزن‌های آن کمترین فاصله را با داده ورودی دارد، به عنوان نرون برنده انتخاب می‌شود و وزن آن و نرون‌های همسایه آن، به سمت داده ورودی بروزرسانی می‌شود. به صورت دقیق‌تر، وزن نرون j در مرحله t ، به صورت زیر بروزرسانی می‌شود:

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

در رابطه بالا $NS(d,t)$ ضریب همسایگی نرونی با فاصله d از نرون برنده را در مرحله t مشخص می‌کند و $\beta(t)$ ، ضریب یادگیری در مرحله t است. هر دو تابع NS و β با افزایش t کاهش می‌یابند. این کاهش می‌تواند خطی یا نمایی باشد. همچنین NS با d رابطه معکوس دارد و این رابطه می‌تواند گاوسی، خطی یا ... باشد. دو نمونه شبکه SOM با خروجی یک و دو بعدی در شکل ۱ نشان داده شده است. لازم به ذکر است همسایگی نرون‌ها می‌توان چهارتایی یا شش‌تایی باشد.



شکل ۱

به طور کلی سه روش برای استفاده از خروجی SOM برای کاهش ابعاد وجود دارد:

۱) استفاده از مختصات/وزن‌های نرون برنده

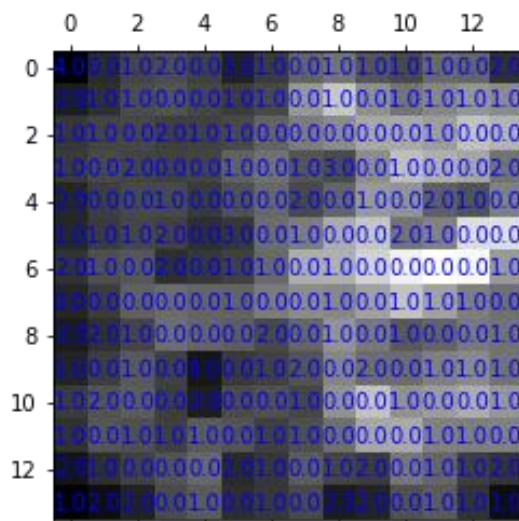
۲) استفاده از مختصات/وزن‌های نرون برنده و همسایه‌های آن

۳) استفاده از نقشه استخراج شده (فاصله هر داده تا نرون‌ها)

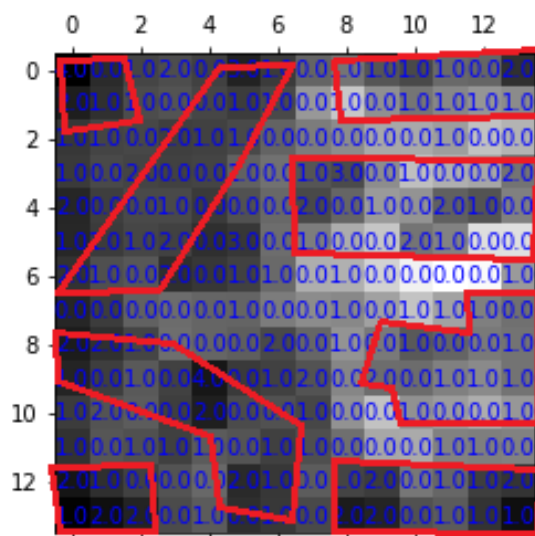
۲) روند کلی خوشه‌بندی و کاهش بعد در شبکه‌های SOM یکسان است. تفاوت اصلی در نحوه نمایش خروجی است. در مسئله خوشه‌بندی از آن‌جا که برچسب داده‌ها مشخص نیست، معمولاً از ماتریس U برای نمایش خروجی و تحلیل استفاده می‌شود. برای کاهش بعد در حالت کلی اگرچه نیاز به برچسب داده‌ها نیست اما در صورت وجود برچسب‌ها، می‌توان با نمایش برچسب هر نرون در مورد بعد مناسب برای کاهش داده‌ها تصمیم گرفت. همچنین در هر دو حالت می‌توان از معیار **purity** برای تصمیم‌گیری استفاده کرد.

۳) کد مربوط به این سوال در فایل **som.py** قرار دارد. پس از استخراج داده‌ها و تقسیم آن‌ها به دو دسته آموزش و آزمون، ابعاد مناسب برای شبکه 14×14 مشخص گردید. برای انتخاب این ابعاد معیار **purity** ملاک تصمیم‌گیری بوده است. همچنین تابع همسایگی گاوسی و کاهش ضریب یادگیری و پارامتر همسایگی، به صورت نمایی تنظیم شده‌است. با آزمون و خطا مقدار مناسب برای ضریب یادگیری اولیه (LR)، ۱ و برای پارامتر همسایگی (R)، یک هشتم عرض شبکه تعیین شد. در شکل ۲، خروجی ماتریس U بعد از ۲۰ تکرار نمایش داده شده است. مقدار **purity** برای این شبکه ۷۵ درصد می‌باشد. همانطور که مشاهده می‌شود به علت کم بودن تعداد داده‌ها نسبت به تعداد برچسب‌ها، همچنین زیاد بودن تعداد فیچرهای ورودی، ماتریس U بسیار پراکنده بوده و اکثر نرون‌های آن کم‌جمعیت است. برای حل این مشکل نمی‌توان تعداد نرون‌ها را کاهش داد، زیرا با این کار **purity** به شدت کاهش می‌یابد. به طور خلاصه، زیاد بودن فیچرهای ورودی باعث پراکندگی بالای خروجی شده است. به بیان بهتر به نظر می‌رسد داده‌ها به صورت خطی در فضای دو بعدی جداپذیری مناسبی ندارند. (البته با توجه به کم بودن تعداد داده‌ها نمی‌توان نظر قطعی داد)

برای یافتن خوشه‌ها در ماتریس U نکته اصلی این است که معمولاً یک خوشه، مجموعه نرون‌هایی است که اختلاف وزن کمی دارند و توسط نرون‌های مرده یا سفید رنگ محاصره شده‌اند. با این توضیحات در شکل ۳، چند خوشه فرضی برای داده‌ها تعیین شده است. به هر حال همانطور که گفته شده خوشه‌ها جدایی‌پذیری مناسبی ندارند و تعداد آن‌ها از کلاس‌ها کمتر است. در صورت افزایش بیشتر ابعاد شبکه، تشخیص خوشه به علت کم بودن داده‌ها سخت‌تر خواهد شد.



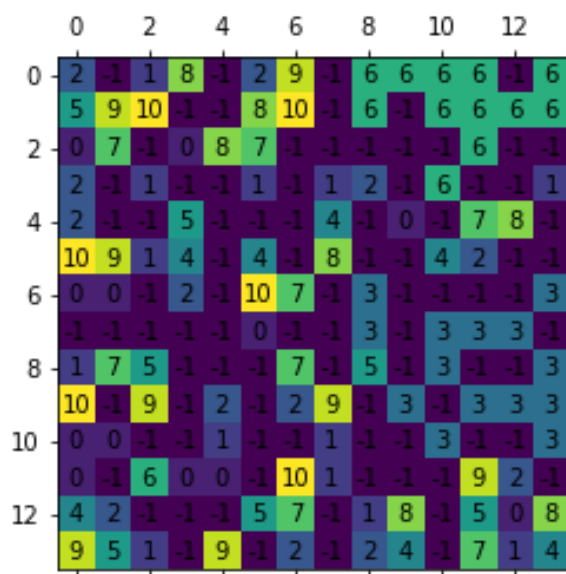
شکل ۲- اعداد درون نرون‌ها، تعداد داده‌های موجود در آن است.



شکل ۳

۴) کد مربوط به این سوال در فایل som.py قرار دارد. در این بخش نیاز به تغییر خاصی در مدل سوال ۳ وجود ندارد و همان شبکه را با خروجی متفاوت استفاده می‌کنیم. برای همان پارامترهای گفته شده در سوال ۳، نمای بصری توزیع کلاس‌ها در شکل ۴ نمایش داده شده است. همانطور که مشاهده می‌شود داده‌ها، به جز کلاس ۳ و ۶، جدایی‌پذیری مناسبی ندارند و به صورت خطی جداپذیر نیستند. محل تقریبی خوشه‌ها تقریباً با شکل ۳ تطابق دارد اما روشن است

که خوشه‌بندی انجام شده بر اساس کلاس‌ها انجام نشده‌است. به هر حال با توجه به کم بودن ابعاد شبکه نسبت به ابعاد اصلی داده‌ها، استفاده از شبکه استخراج شده به عنوان ویژگی، سرعت و دقت شبکه را افزایش خواهد داد. با توجه به اینکه **purity** این شبکه ۷۵ درصد است، انتظار می‌رود حداکثر دقت آموزش شبکه در حدود ۷۵ درصد باشد.



شکل ۴

در انتها ذکر می‌شود که خوشه‌های ۳ و ۶ به ترتیب مربوط به کلاس **leftlight** و **rightlight** می‌باشد. به نظر می‌رسد جهت نورپردازی بهتر از سایر حالت‌ها توسط شبکه قابل تشخیص بوده است. برای بررسی علت، دو نمونه از داده‌های کلاس **leftlight** در شکل ۵ نمایش داده شده است. با توجه به شکل‌ها، احتمالاً وجود تعداد زیادی پیکسل سایه موجب تفکیک پذیری بالای این کلاس‌ها شده است.



شکل ۵

(۵) کد مربوط به این سوال در فایل **hw3_cl.py** قرار دارد.

داده‌های اصلی: بعد از نرمال‌سازی داده‌ها، از یک مدل سه لایه ۷۷۷۶۰-۱۰۰-۱۱ برای دسته‌بندی استفاده می‌کنیم. برای این کار لایه خروجی دارای تابع فعالیته **softmax** خواهد بود. همچنین تابع هزینه **CategoricalCrossentropy** می‌باشد. شرط توقف آموزش عدم بهبود **val-loss** در ۵ تکرار متوالی می‌باشد. با این توضیحات، نتایج خروجی شبکه به این صورت است:

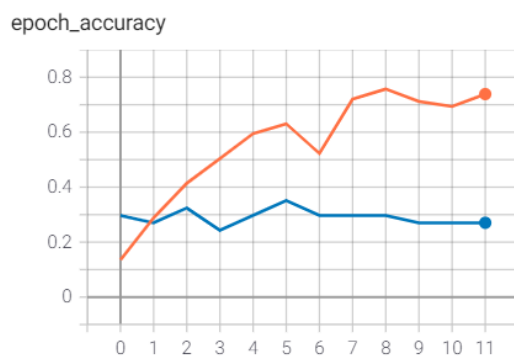
```
loss: 14.9253 - accuracy: 0.7081 - val_loss: 151.5855 - val_accuracy: 0.2973
```

همچنین دقت و ماتریس درهم ریختگی داده‌های تست به این صورت می‌باشد.

Accuracy for test: 0.17647058823529413

	0	1	2	4	5	6	7	8	9	10
0	0	0	1	1	0	0	0	1	0	0
1	0	0	1	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0
4	0	0	1	0	0	2	0	0	0	1
5	0	0	0	0	1	0	0	0	0	0
6	0	0	0	0	0	0	1	0	0	0
7	0	0	0	1	0	0	0	0	0	0
8	0	0	0	0	0	0	1	0	0	0
9	0	0	0	0	0	0	1	1	0	0
10	0	0	0	0	0	0	0	1	0	1

اگرچه با توجه به کمبودن داده‌های تست نمی‌توان تحلیل دقیقی روی ماتریس درهم‌ریختگی داشت، اما با توجه به دقت سه مجموعه آموزش، اعتبارسنجی و آزمون می‌توان گفت مدل دچار بیش‌برازش شده‌است. اگرچه با توجه زیادن بودن ویژگی‌ها و کلاس‌ها در مقایسه با تعداد داده‌ها، این مورد به دور از انتظار نبود. در نهایت نمودار دقت بر حسب تکرار در شکل ۶ رسم شده‌است. وقوع بیش‌برازش در این نمودار نیز مشهود است.



شکل ۶

داده‌های کاهش یافته: (دیتاست کاهش یافته در پوشه re موجود است)

فیچرهای استخراج شده از شبکه SOM به صورت معکوس فاصله داده از هر نرون می‌باشد. بنابراین داده‌ها در فضای جدید دارای ابعاد ۱۹۶ می‌باشد که به نسبت فضای قبلی به شدت کاهش یافته است. پس از نرمال‌سازی داده‌ها از شبکه‌ای مشابه قسمت قبل استفاده می‌کنیم (به جز لایه ورودی که ۱۹۶ نرون دارد) نتایج به شرح زیر است.

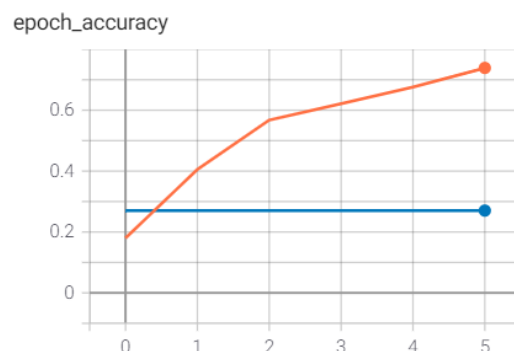
loss: 0.7706 - accuracy: 0.7747 - val_loss: 3.7068 - val_accuracy: 0.2703

همچنین دقت و ماتریس درهم‌ریختگی داده‌های تست به این صورت می‌باشد.

Accuracy for test: 0.23529411764705882

	0	1	2	4	5	6	7	8	9	10
0	1	0	0	1	0	1	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0
2	0	0	0	0	0	0	0	1	0	0
4	0	0	1	0	1	0	0	0	1	0
5	0	0	0	1	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0
7	0	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	1	0	0	0
9	0	0	0	0	1	0	0	0	1	0
10	0	0	0	1	0	0	0	1	0	0

در اینجا نیز با توجه به نتایج می‌توان گفت مدل دچار بیش‌برازش شده‌است. همچنین این مورد در نمودار دقت بر حسب تکرار که در شکل ۷ رسم شده مشهود است.



شکل ۲

نتیجه گیری و مقایسه:

اگرچه شبکه از نظر دقت در داده‌های کاهش یافته اندکی بهتر عمل کرده است (به ویژه در داده‌های آزمون)، اما با توجه به کم بودن تعداد داده‌ها، در اجراهای مختلف ممکن است این نتایج متفاوت باشند و بنابراین نمی‌توان برتری خاصی در این زمینه به داده‌های کاهش یافته با SOM داد. البته با توجه به کم بودن تعداد داده‌ها نمی‌توان در این مورد نظر قاطعانه داد. در مقابل همانطور که انتظار می‌رود در داده‌های کاهش یافته شبکه بسیار سریع‌تر عمل می‌کند. (هم از نظر سرعت هر تکرار و هم از نظر سرعت همگرایی) بنابراین می‌توان انتظار داشت در دیتاست‌هایی با تعداد بالای داده، استفاده از شبکه SOM می‌تواند بسیار موثر باشد.