به نام خدا

تمرین سوم درس شبکه عصبی

زهرا دهقانیان

۹۸۱۳۱۰۵۹

سوال اول)

شبکه های خود سازمانده و یا SOM شبکه هایی هستند که در نگاه کلی نگاشت روش k نزدیک همسایگی و یا KNN در شبکه های عصبی است. در حقیقت به کمک این شبکه ها میتوان یک نمود تصویری از وضعیت داده ها بدست آورد. ساختار این شبکه های به این صورت است که یک سری نود با همسایگی خطی و یا دوبعدی و یا بیشتر (معمولا در همان ۲ بعدی و یا نهایتا ۳ بعدی خواهد بود) هستند که در طی روند آموزش و در نهایت هر کدام نماینده یک مرکز خوشه هستند و ویژگی خاص این شبکه، همسایه شدن و در نزدیک هم قرار گرفتن خوشه های شبیه به هم است. این شبکه به طور کلی از دو لایه ورودی و خروجی تشکیل شده و هیچ لایه مخفی وجود ندارد. وزن در این شبکه به صورت بروزرسانی می شود که ابتدا به ازای هر داده ابتدا نورون برنده یافته می شود: روش پیدا کردن نورون برنده به این صورت است که فاصله اقلیدسی داده مورد نظر تا همه نود ها حساب میشود و نودی که کمترین فاصله را داشته باشد و نزدیک تر باشد به عنوان نود برنده انتخاب میشود. سپس وزن های نورون برنده و همسایه های موجود در همسایگی آن (همسایگی ممکن است مربعی، دایره ای، خطی و یا مدل های دیگر باشد) با نرخ معکوس با فاصله (یعنی هر چه نزدیک تر باشد ، میزان بیشتری) اپدیت میشود. میزان اپدیت به این صورت است که به اندازه اختلاف در نرخ حساب شده در مرحله قبل و ضرب در نرخ یادگیری اپدیت مقادیر MAP انجام میشود. در یک چرخه به تعداد iteration های وارد شده، اپدیت می شود و در حین این اموزش ها، اندازه همسایگی ها کوچک میشود تا در نهایت تنها خود نورون در همسایگی باشد. البته اموزش در هنگامی که مپ به نسبت مرحله قبل هیچ تغییری نکند نیز میتواند متوقف بشود.

کاربرد برای کاهش حجم داده به این صورت است که میتوان از بردار فاصله هر داده با همه نورون های مپ به عنوان بردار بازنمایی یک داده روی مپ استفاده کرد و عملا در این صورت به جای نگه داری تعداد زیاد feature میتوان تنها فاصله تا نورون ها را نگه داشت.

سوال دوم)

در خوشه بندی، به این صورت عمل میکنیم که پس آموزش و یافتن دقیق مقادیر نورون در مپ (که در واقع همان مراکز خوشه بندی هستند). برای خوشه بندی هر ورودی جدید فاصله تا همه عناصر مپ را حساب میکنیم و متعلق به خوشه این میشود که فاصله تا مرکز آن خوشه (یعنی تا نورون مربوط به آن خوشه) کمترین مقدار باشد. در اینجا میتوان به کمک روش های سلسه مراتبی نورون های همسایه را ادغام کرد که در این صورت تعداد خوشه ها لزوما به تعداد نورون های نقشه نیست. اگر در حالت دسته بندی باشیم ، پس از آموزش شبکه میتوان به کمک داده های آموزش ، برچسبی که بیشتر از همه به یک نورون نگاشت شده را به عنوان مقدار کلاس ان نورون در نظر گرفت که در این صورت برای هر داده جدید، کلاس میشود کلاس نورون که کمترین فاصله تا داده را دارد.

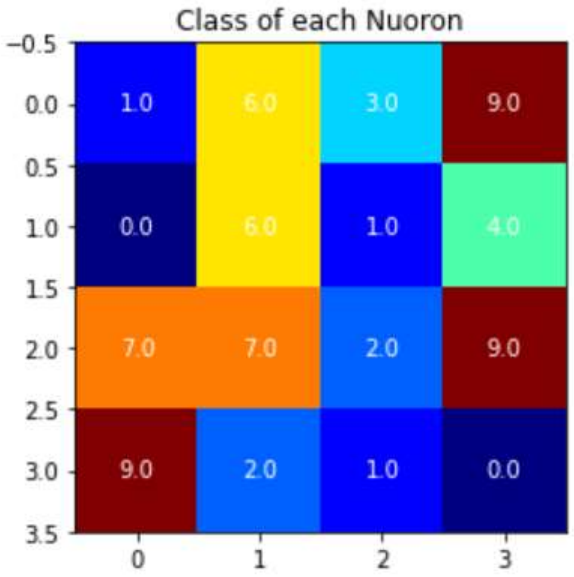
در حالت کاهش بعد، به نوعی انگار داریم به یک فضای جدید با تعداد نورون های نقشه نگاشت میکنیم و در آن فضا مقدار هر داده به این صورت محاسبه می شود که فاصله ان داده تا مراکز خوشه را بدست می اوریم.

شباهت این دو روش در این است که در هر دو روش ابتدا باید مراکز خوشه هارا با اموزش کامل بدست آورد و سپس فاصله تمام نورون ها تا بردار ورودی مورد نظر را بدست آورد. تفاوت در ادامه ماجراست در حالت کاهش بعد، میتوان همین بردار و یا معکوس بردار فاصله ها (برای این که هر چه به یک نورون شبیه تر باشیم مقدار بیشتر باشد) را به عنوان خروجی شبکه در نظر گرفت اما برای خوشه بندی، یک مرحله مینیم گیری هم داریم و خوشه داده ورودی ، خوشه و یا نورون با مینیمم فاصله است.

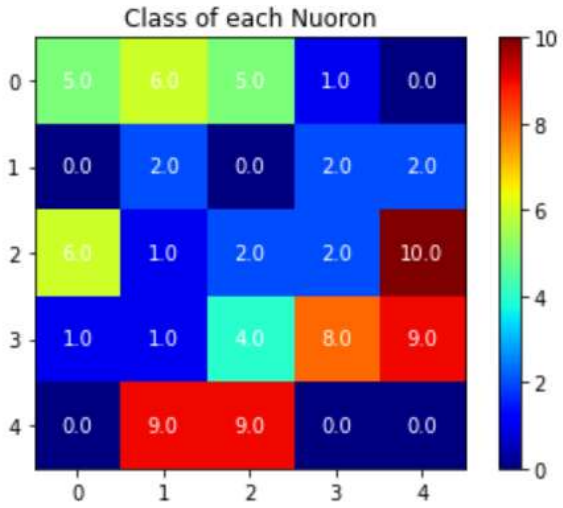
سوال سوم)

در اینجا ساختار شبکه بر اساس ویدیو های کلاس تدریس یار پیاده سازی شده است. در این جا cpu لپ تاپ بنده با این که زمانی در حدود ۲۴ ساعت برای آموزش شبکه در نظر می گرفتم اما شبکه بزرگتر از ۴ در ۴ را جواب نمیداد. به همین دلیل کد را بر روی گوگل کولب بردم و در آن جا هم با ۵۰۰۰ ایپک برای نقشه ۶ در ۶ حدودا ۱۰ ساعت زمان برد. الان که این گزارش را مینویسم حدودا ۴۶ ساعت که برای نقشه ۷ در ۷ کولب در حال اجراست ولی همچنان به جواب نرسیده است. در بین نقشه های بدست آمده نقشه های کوچک تر از ۴ که به نظر مناسب نمی رسد، زیرا از کلاس های داده هم تعداد نورون کمتری دارد. در ادامه نقشه ها با اندازه های ۴ و ۵ و ۶ آمده است:

نقشه ۴ در ۴ :

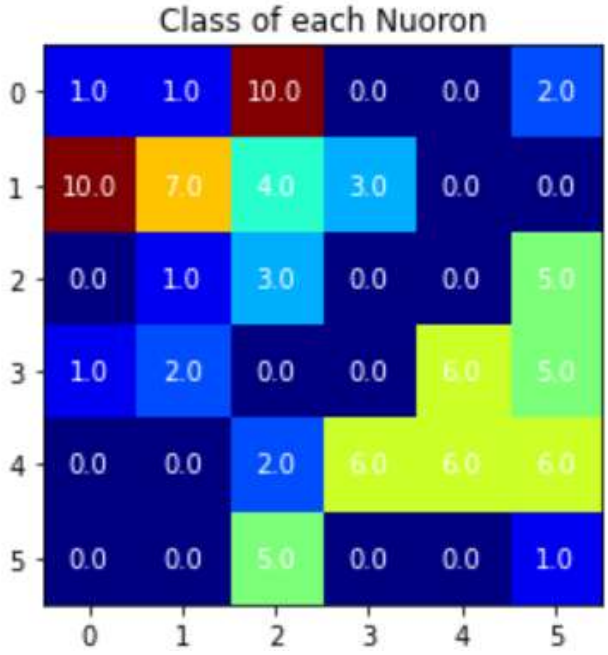


دقت برابر با ۰,۰۹۴۳ است. نقشه ۵ در ۵ به صورت زیر است:



دقت در این مدل برابر با ۰,۰۶۸۵ است.

نقشه ۶ در ۶ :



دقت برابر با ۰,۰۶۶۷ است. همان طور که میبینید به نظر می رسد نقشه با ابعاد ۴ در ۴ دقت بالاتری داشته است. البته که هیچ کدام از نقشه به نتیجه خیلی خوبی نرسیده است.

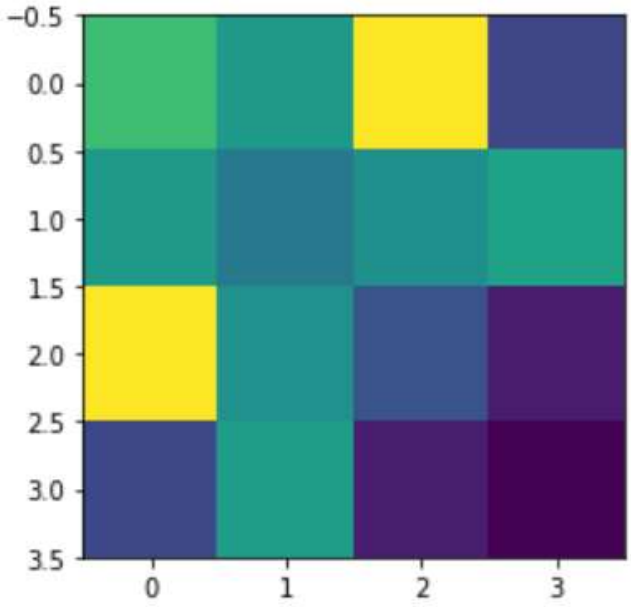
برای مقدار همسایگی بهینه ، بنده همسایگی با اندازه نصف ، ۳/۱ و با اندازه ۱ را بررسی کردم و به نظر می رسید حالتی که از نصف شروع کنیم و در طی اموزش به صورت خطی کم کنیم جواب هایی با دقت بیشتر بدست می آورد

برای مقایسه این شبکه ها ایده ای که به ذهن بنده رسید این بود که پس از اموزش شبکه ، برای هر نورون کلاسی که بیشتر داده های آن خوشه متعلق به آن هستند را به عنوان کلاس آن نورون درنظر بگیریم و سپس برای داده ها ببینیم چه تعدادی از داده های تست، کلاس درست برایشان در نظر گرفته می شود یعنی همان پارامتر accuracy پارامتر ارزیابی در میان این شبکه هاست.

** قابل توجه است که به نظر خروجی هیچ کدام از شبکه ها خروجی مطلوب و مناسب که عکس ها کاملا خوشه بندی شده و به ترتیب شباهت قرار گیرند را نیافته است. به همین دلیل برای شبکه ۴ در ۴ کد با تعداد ۲۰۰۰۰ چرخه را اجرا کردیم اما باز هم به نتیجه خوبی نرسیدیم و علت این امر برای بنده مشخص نشده است.

سوال چهارم)

همان طور در سوال ۱ توضیح داده شد، بردار بازنمایی کاهش ابعاد یافته در اینجا در واقع همان بردار فاصله ها تا هر کدام از نورون های نقشه است. بازنمایی که حاصل از این کاهش ابعاد است، بازنمایی با ابعادی هم اندازه نقشه است. به عنوان مثال خروجی برای داده اول در نقشه ۴ در ۴ به صورت زیر بوده است :



همان طور که می بینید، این داده به احتمال زیاد به کلاس نورون روشن تر تعلق دارد ، زیرا فاصله آن از بقیه خوشه ها به نسبت زیاد بوده و خیلی به این نورون شبیه بوده است و با توجه به تصویر نقشه ۴ در ۴ در بخش قبل ، این نورون متعلق به کلاس ۲ است و یا ۵ یعنی دو کلاس روشن تر است و برچسب این داده، ۲ است و نکته قابل توجه این است که کلاس ۲ normal و کلاس ۵ no_glasses است که این دو کلاس نیز شبیه هم هستند و این بازنمایی تا حد خوبی برای ما قابل قبول است.

بدین ترتیب برای محاسبه ورودی بخش بعدی، تابع extract_feature را برای همه داده صدا میزنیم و سپس تالع flatten را نیز فراخوانی میکنیم تا از شکل ۲ بعدی به ۱ بعدی تبدیل شوند.

سوال پنجم)

برای خوشه بندی شبکه ای که به کار بردیم ساختار (۱۱، ۳۰ ، ۱۰۰ ، تعداد ورودی) دارد. با توجه به این که مسیله چند کلاسه است تابع activation تابع softmax و برای تابع ضرر از تابع sparse_categorical_crossentropy استفاده شده است. حال یک بار مجموعه داده اولیه و بار دیگر مجموعه داده ساخته شده در بخش قبل را به عنوان ورودی به این شبکه می دهیم و ۱۰۰۰ ایپک آموزش می دهیم.

خروجی کد در حالت کاهش بعد یافته به صورت زیر است :

```
Model: "sequential_24"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_62 (Dense)             (None, 100)               1700
_________________________________________________________________
dropout_21 (Dropout)         (None, 100)               0
_________________________________________________________________
dense_63 (Dense)             (None, 30)                3030
_________________________________________________________________
dropout_22 (Dropout)         (None, 30)                0
_________________________________________________________________
dense_64 (Dense)             (None, 11)                341
=================================================================
Total params: 5,071
Trainable params: 5,071
Non-trainable params: 0
_________________________________________________________________

Accuracy reduced model : 0.212
```

و خروجی کد در حالت عادی به صورت زیر است:

```
Model: "sequential_25"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_65 (Dense)             (None, 100)               7776100
_________________________________________________________________
dropout_23 (Dropout)         (None, 100)               0
_________________________________________________________________
dense_66 (Dense)             (None, 30)                3030
_________________________________________________________________
dropout_24 (Dropout)         (None, 30)                0
_________________________________________________________________
dense_67 (Dense)             (None, 11)                341
=================================================================
Total params: 7,779,471
Trainable params: 7,779,471
Non-trainable params: 0
_________________________________________________________________

Accuracy normal model : 0.09
```

همانطور که در خروجی کدها امده است، دقت در حالت کاهش یافته ۰,۲۱ و دقت در حالت شده Mp عادی ۰,۰۹ بوده است و همانطور که میبینید این کاهش بعد سبب بهبود عملکرد است. البته بهبود اصلی حاصل از این روش در کاهش چندین برابری زمان آموزش است؛ به طوری که در حالت کاهش یافته مدل در حدودا ۱ دقیقه و در حالت عادی حدودا ۱۰ دقیقه طول می کشد.