به نام خدا

دانشگاه تهران
دانشکده مهندسی برق و کامپیوتر

شبکه های عصبی و یادگیری عمیق

تمرین سری ۲

نام و نام خانوادگی	سجاد پاکدامن ساوجی
شماره دانشجویی	۸۱۰۱۹۵۵۱۷
تاریخ ارسال گزارش	۱۵ فروردین

فهرست گزارش سوالات

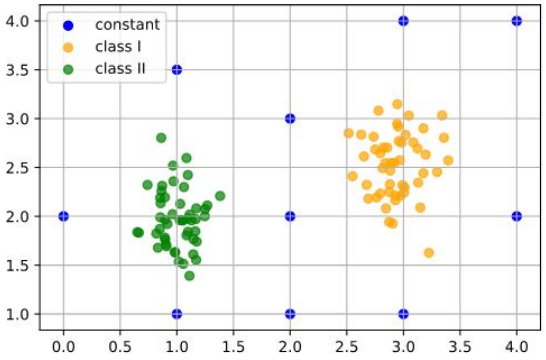
سوال 1 – Madeline

داده ها با شرایطی که خواسته شده است، تولید شد. شرایط داده ها در زیر آمده است

* نقاط آبی رنگ ثابت هستند

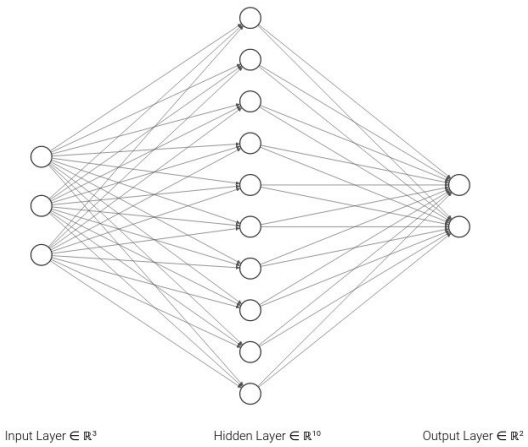
* داده های نارنجی شامل ۵۰ نقطه هستند که مولفه X آن‌ها دارای میانگین ۳ و واریانس ۰.۲ است. مولفه Y این داده ها نیز دارای میانگین ۲.۵ و انحراف از معیار ۰.۳ می‌باشد.

* داده های سبز شامل ۵۰ نقطه هستند که مولفه X آن‌ها دارای میانگین ۱ و واریانس ۰.۲ است. مولفه Y این داده ها نیز دارای میانگین ۲ و انحراف از معیار ۰.۲ می‌باشد.



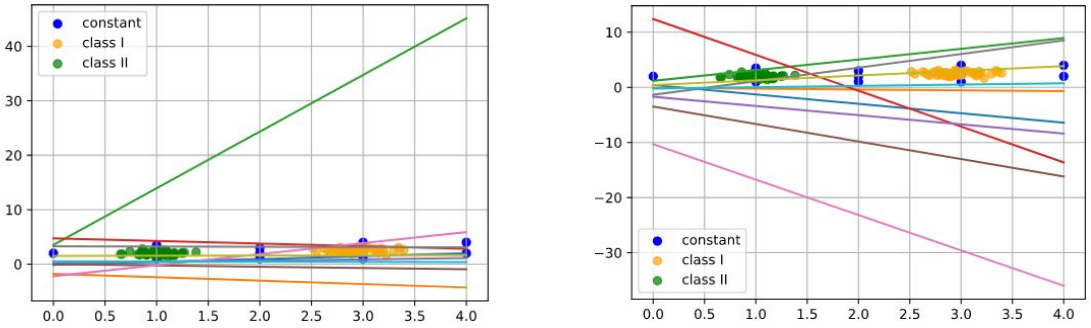
شکل ۱. داده های تولید شده

در ادامه با استفاده از یک شبکه Madeline داده ها تفکیک شده اند. ورودی این شبکه بعد ۳ دارد (۲ بعد برای x و y و یک بعد برای بایاس). لایه میانی شبکه دارای ۱۰ نرون خواهد بود و لایه خروجی شبکه ۲ نرون دارد. نورون های شبکه از تابع فعال سازی علامت استفاده می‌کنند.



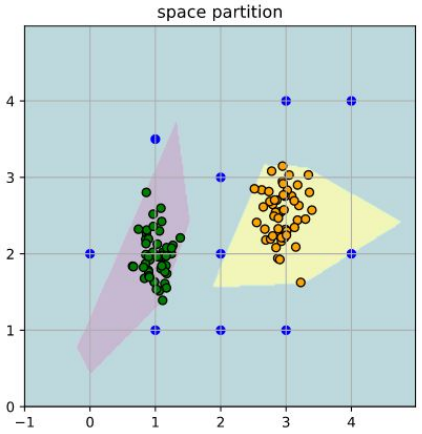
شکل ۲. معماری شبکه madeline

برای آموزش شبکه از الگوریتم MRI استفاده شده است که در آن لایه آخر (AND) فرض می‌شود و تلاش می‌شود با اعمال سیاست min disturbance شبکه را با استفاده از روش یادگیری دلتا آموزش داد. برای آموزش شبکه شبکه مربوط به هر کلاس جداگانه آموزش داده شد. شکل ۳ و ۴ خطا های جدا کننده را برای هر یک از کلاس ها نشان می‌دهند.



شکل۳و ۴ . خطوط جدا کننده فضا برای هر یک از کلاس ها

برای این که صحت عملکرد شبه بررسی شود فضا را با استفاده از شبکه بخش بندی کرده ایم. توجه شود در مکان هایی که هر دو نرون ۰ بازگردانند، کلاس آب تشخیص داده می‌شود.



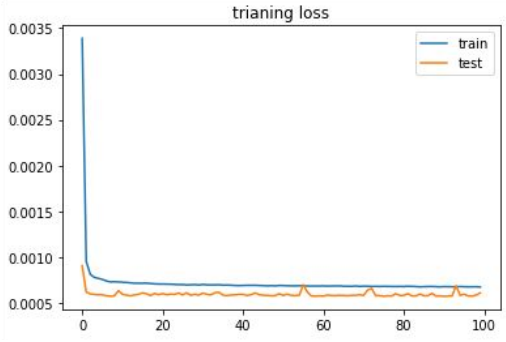
شکل ۵. بخش بندی فضا توسط شبکه Madeline

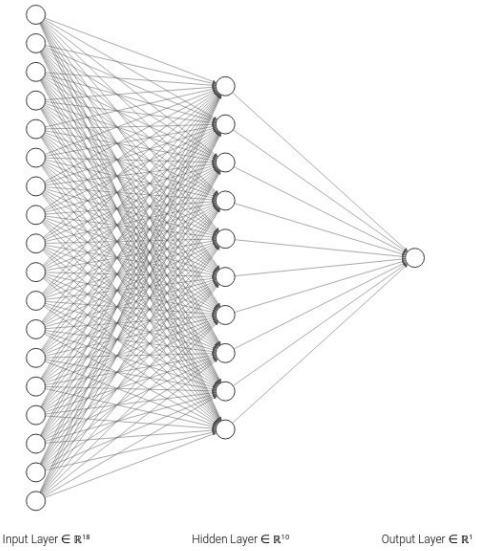
سوال ۲ – House Sales

در این سوال ۵۰۰۰ داده اول مجموعه داده housesales.csv را که در اختیار ما قرار داده شده است را جدا می‌کنیم و از ۸۰ درصد آن برای آموزش و از ۲۰ درصد باقی برای آزمایش استفاده می‌کنیم.

در ابتدا برای این که داده ها در عملکرد شبکه بایاس ایجاد نکنند، تمامی ویژگی ها را استاندارد کردیم و مقادیر آن ها را بین ۰ و ۱ قرار دادیم. قیمت ها نیز استاندارد شدند. پس از پیش‌پردازش های اولیه ۱۸ ویژگی برای آموزش شبکه انتخاب شدند.

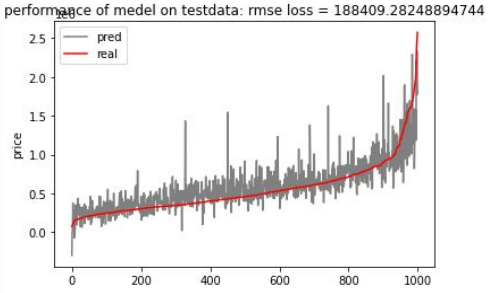
برای آموزش شبکه MLP تک لایه از ۱۰ نورون مخفی استفاده شد. معماری شبکه در شکل ۶ آمده است. تابع هزینه MSELOSS انتخاب شد و نرخ یادگیری در هر epoch با شروع از ۰.۵ ضرب در ۰.۹ می‌شود. نمودار خطا برای ۱۰۰ epoch در شکل ۷ آمده است.

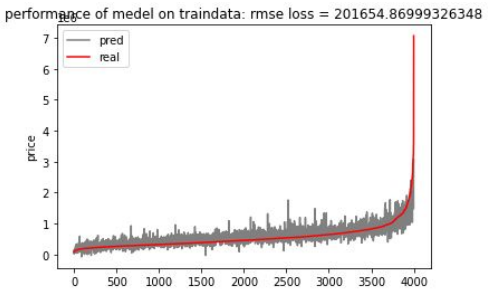




شکل ۶و ۷. ساختار شبکه تک‌لایه طراحی شده و نمودار هزینه برای داده های آموزش و آزمایش

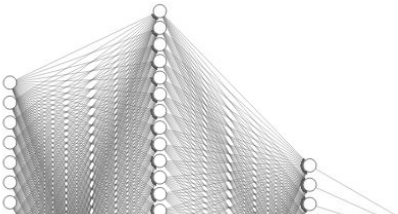
هزینه روی داده های آموزش پس از اتمام آموزش برابر ۰.۰۰۱ شد و خطا برای داده های آزمایش برابر با ۰.۰۰۰۷ شد. در ادامه عملکرد شبکه روی داده های آموزش و آزمایش نشان داده شده است.



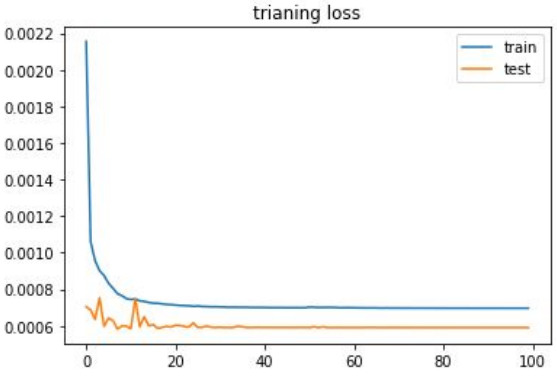


شکل ۸و ۹. عملکرد شبکه بر داده های آموزش و آزمایش

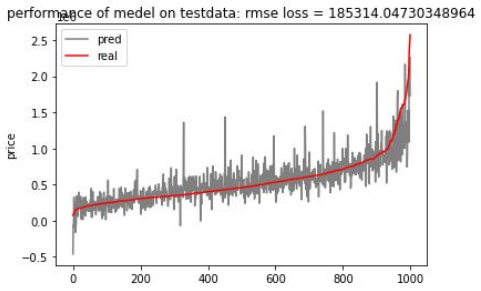
در ادامه شبکه دو لایه با ۳۰ نورون در لایه اول و ۱۰ نرون در لایه دوم ساخته شد. برای آموزش این شبکه کاملا همانند شبکه پیشین عمل کردیم. ساختار و عملکرد شبکه در شکل های ۱۰ و ۱۱ آمده است.

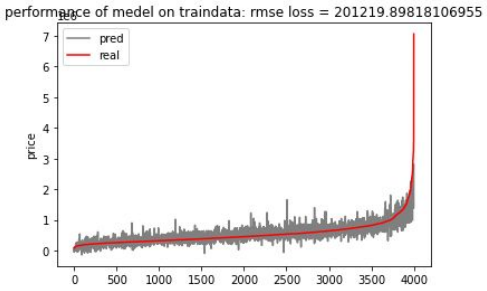


شکل ۱۰ و ۱۱. معماری شبکه دو لایه و عملکرد آن بر روی داده های آموزش و آزمایش



هزینه روی داده های آموزش پس از اتمام آموزش برابر ۰.۰۰۰۸ شد و خطا برای داده های آزمایش برابر با ۰.۰۰۰۶ شد. در ادامه عملکرد شبکه روی داده های آموزش و آزمایش نشان داده شده است.





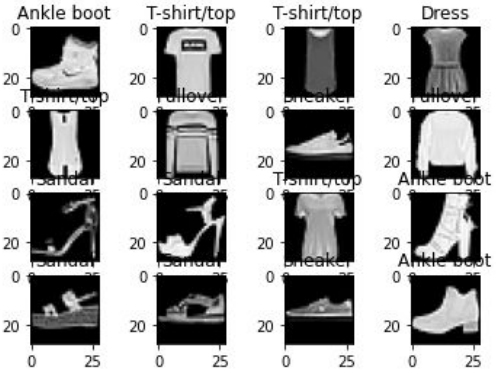
شکل ۱۲و ۱۳ . عملکرد شبکه دو لایه MLP بر روی داده های آموزش و آزمایش

سوال ۳ - Fashion MNIST

الف) درصد داده های آموزش، تست و ارزیابی بستگی به حجم داده ها دارد به این معنی که با توجه به ساختار شبکه و عملکرد آن تعیین می‌شود. برای مثال در تشخیص چهره باید تعداد داده های آموزش بسیار کم باشد و تعداد داده های تست بیشتر باشد زیرا در استفاده واقعی تعداد اندکی تصویر از هر فرد وجود دارد در صورتی که سیستم شناسایی ممکن است بار ها مورد استفاده قرار بگیرد(تست شود). نکته ای که در تقسیم این داده ها باید رعایت شود این است که هر سه مجموعه داده یک faire sample از مجموعه آماری باشند به این معنا که باید داده ها با توجه به توزیع احتمال اصلی نمونه برداری شده باشند تا هر مجموعه نماینده خوبی از کل مجموعه آماری باشد.

ب) در این قسمت تاثیر تعداد نرون های لایه های مخفی در عملکرد شبکه بررسی می‌شود. رد کل ۳ بار تعداد نرون های لایه مخفی تغییر پیدا کرده است.

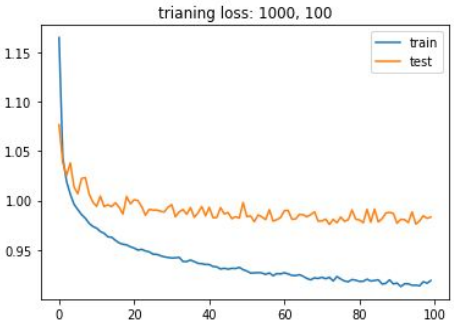
در مورد زمان آموزش شبکه باید این نکته را گفت که اگر با سخت افزار یکسان و تابع هزینه یکسان(دقیقا شرایطی که سوال از ما خواسته است) شبکه های مختلف را آموزش دهیم، زمان آموزش شبکه تابعی صعودی از تعداد یال‌ها(وزن‌ها) ی شبکه خواهد بود. پس در حالتی که ضرب h1 در h2 بزرگ‌تر شود زمان بیشتری برای آموزش شبکه نیاز است.



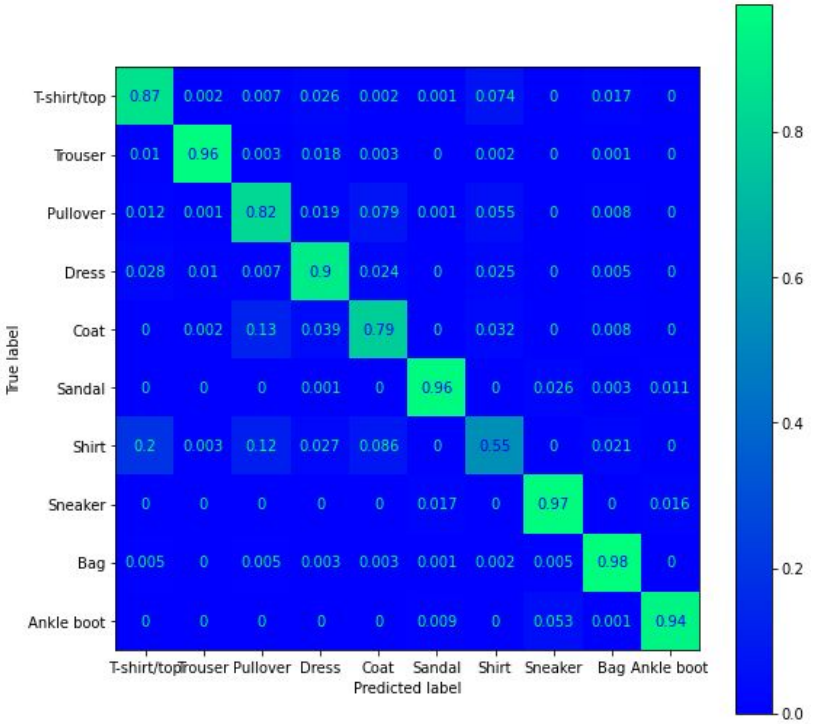
شکل ۱۴. تعدادی از تصاویر مجموعه داد fashion MNIST

در ادامه عملکرد ۳ شبکه طراحی شده گزارش می‌شود. در آموزش تمامی شبکه ها از SGDبه عنوان optimizer استفاده می‌شود و تابع هزینه CrossEntropyLoss است.

شبکه ۱. ۷۸۴ - ۱۰۰۰ - ۱۰۰ - ۱۰



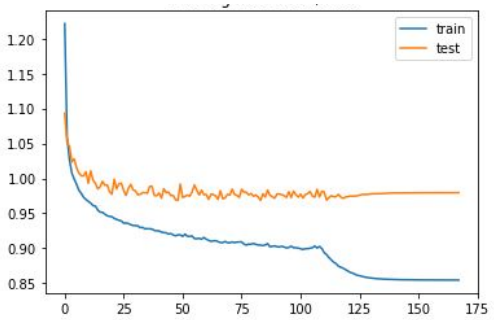
شکل ۱۵. نمودار دقت و هزینه برای شبکه ۱

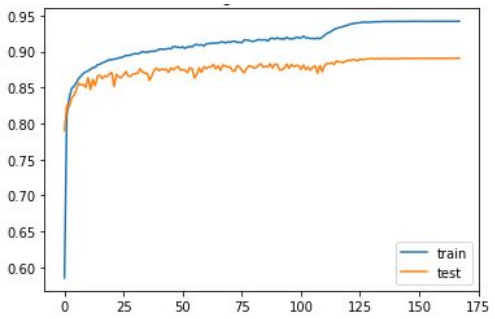


شکل ۱۶ . ماتریس آشفتگی برای شبکه ۱

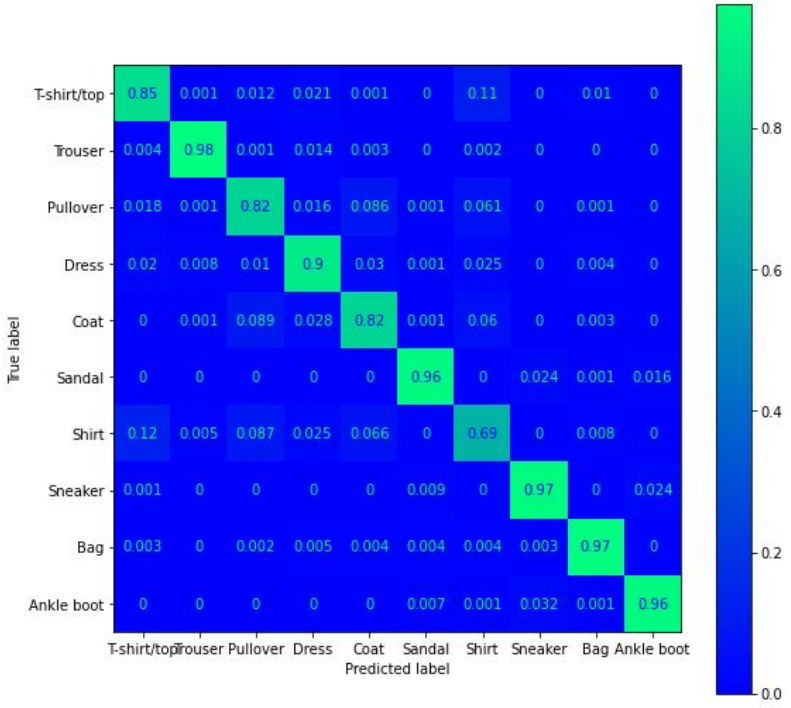
دقت شبکه روی داده تست برابر 0.8736 می‌باشد و هزینه آن مطابق شکل ۱۵ برابر ۱ می باشد.

شبکه۲. ۷۸۴ - ۵۰۰ - ۱۰۰ - ۱۰





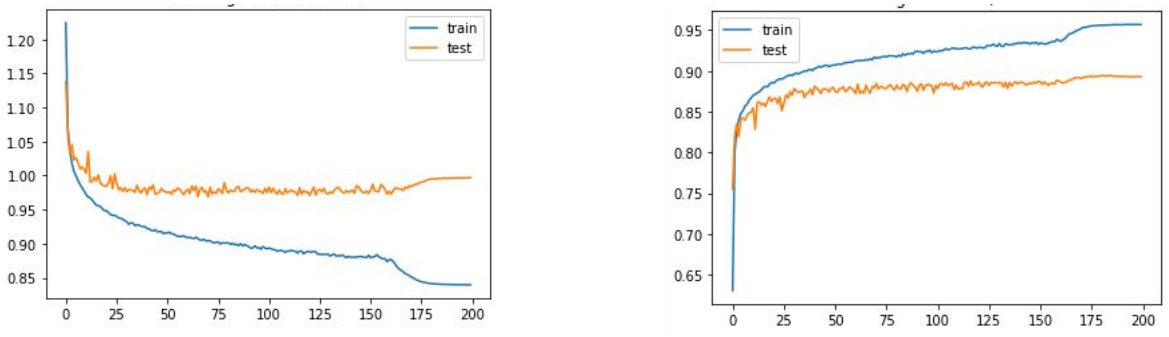
شکل ۱۷. نمودار دقت و هزینه برای شبکه ۲



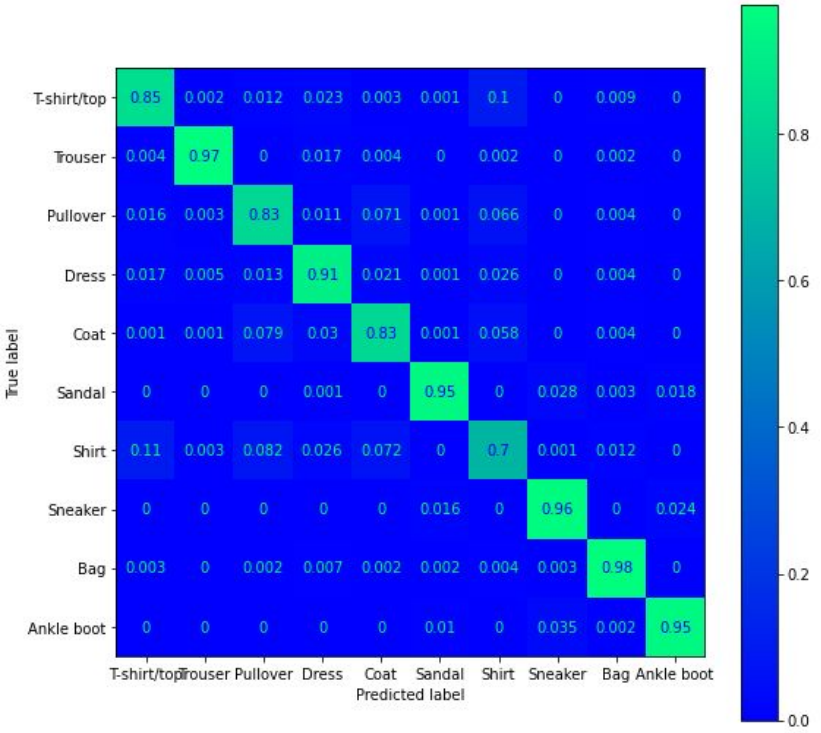
شکل ۱۸. ماتریس آشفتگی برای شبکه ۲

دقت شبکه روی داده تست برابر 0.8905 می‌باشد و هزینه آن مطابق شکل ۱۷ برابر ۰.۹۷ می باشد.

شبکه ۳. ۷۸۴ - ۵۰۰ - ۲۰۰ - ۱۰



شکل ۱۹ . دقت و هزینه برای شبکه ۳



شکل ۲۰ . ماتریس آشفتگی برای شبکه ۳

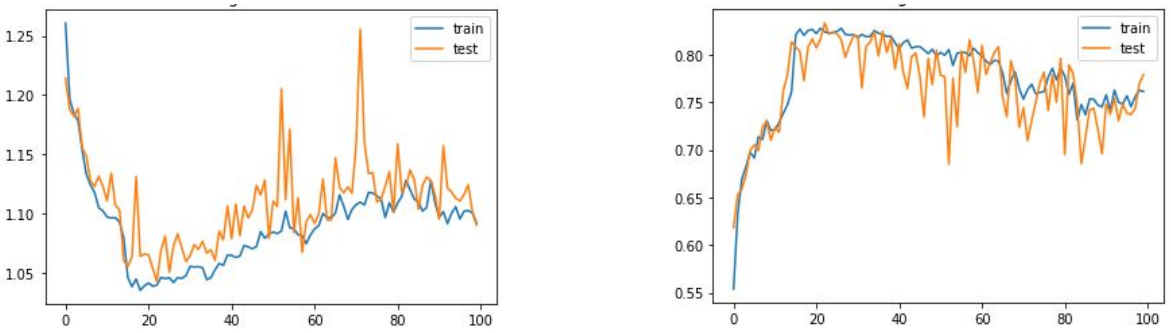
دقت شبکه روی داده تست برابر 0.8926 می‌باشد و هزینه آن مطابق شکل ۱۹ برابر ۰.۹۶ می باشد.

ج) در روش بهینه‌سازی SGD هرچه batch size بزرگ‌تر شود به روش بهینه‌سازی GD نزدیک تر میشویم که تاثیر آن را می‌توان در نرم تر شدن نمودار هزینه مشاهده کرد. علاوه بر آن وقتی batch size به اندازه کافی بزرگ انتخاب می‌شود به دلیل بهینه‌سازی های نرم‌افزاری در ضرب ماتریسی و بهینه‌سازی های سخت‌افزاری همانند

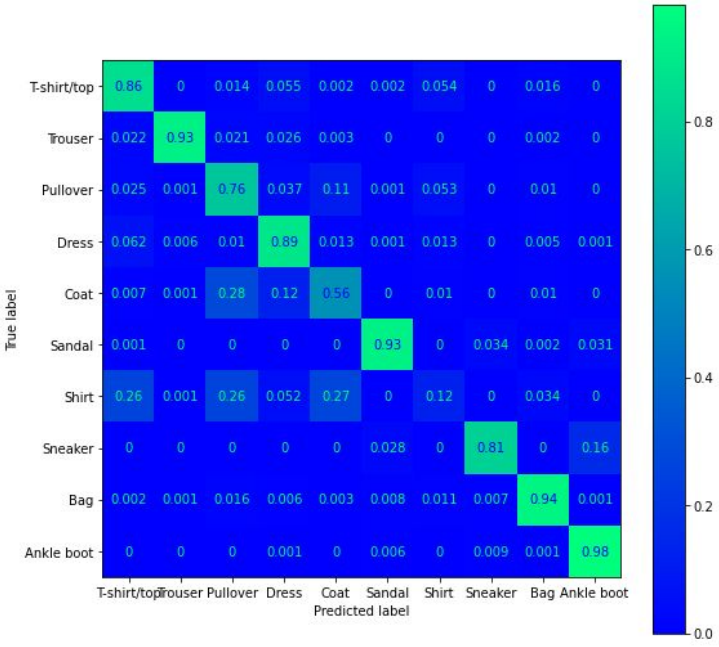
ضرب ماتریسی روی GPU زمان هر epoch کاهش می‌یابد. مشکل batch size های خیلی بزرگ آن‌جایی ظاهر می‌شود که یک batch در حافظه GPU جا نمی‌شود و باعث می‌شود که محاسبات دشوار تر انجام شوند.

در ادامه نمودار های دقت و هزینه برای batch size های مختلف آورده شده است.

حالت اول. batch size = 32



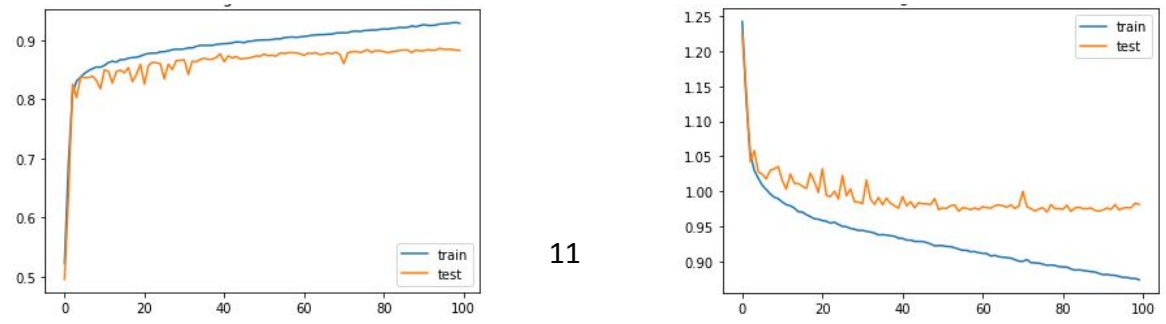
شکل ۲۱. نمودار دقت و هزینه برای batch size = 32



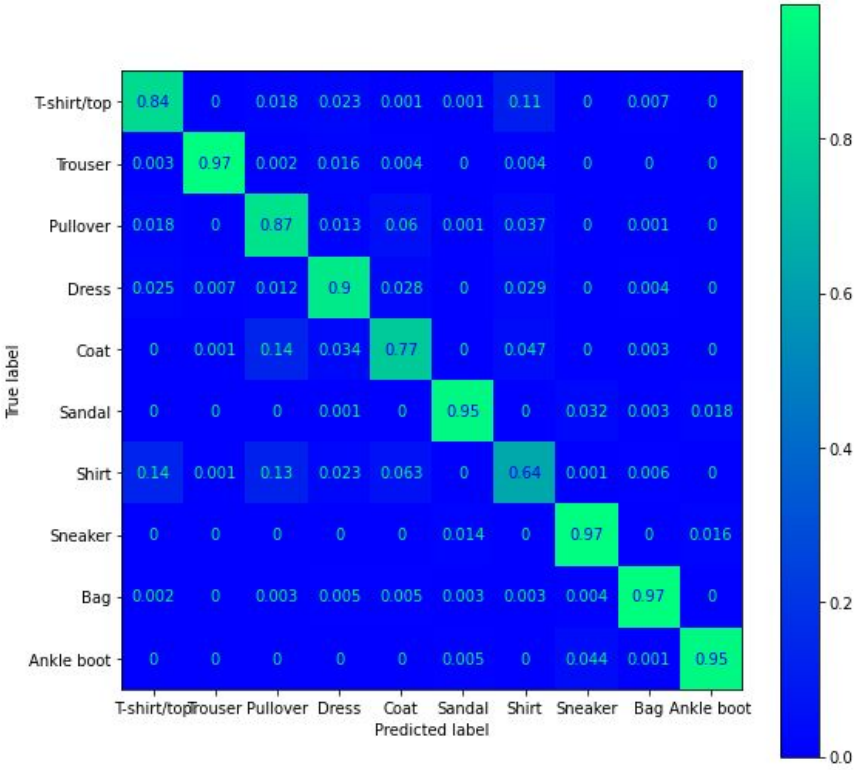
شکل ۲۲. ماتریس آشفتگی برای حالت batch size = 32

دقت شبکه روی داده تست برابر 0.779 می‌باشد و هزینه آن مطابق شکل ۲۱ برابر ۱. ۱ می باشد.

حالت دوم. batch size = 64



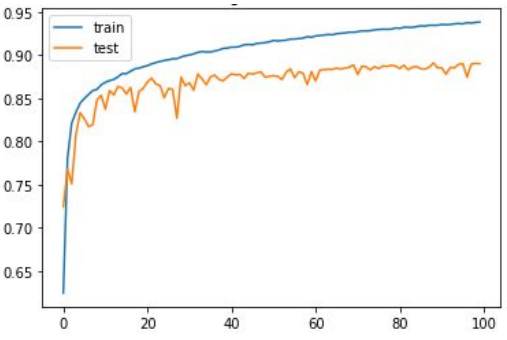
شکل ۲۳. نمودار هزینه و دقت برای حالت batch size = 64

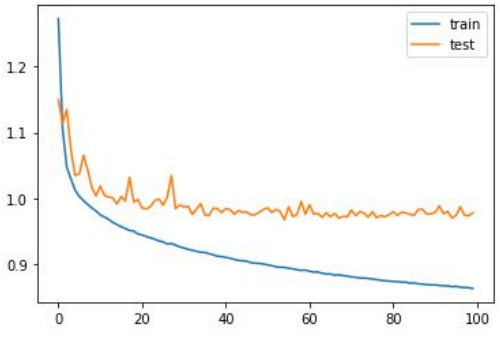


شکل ۲۴. ماتریس آشفتگی برای حالت batch size = 64

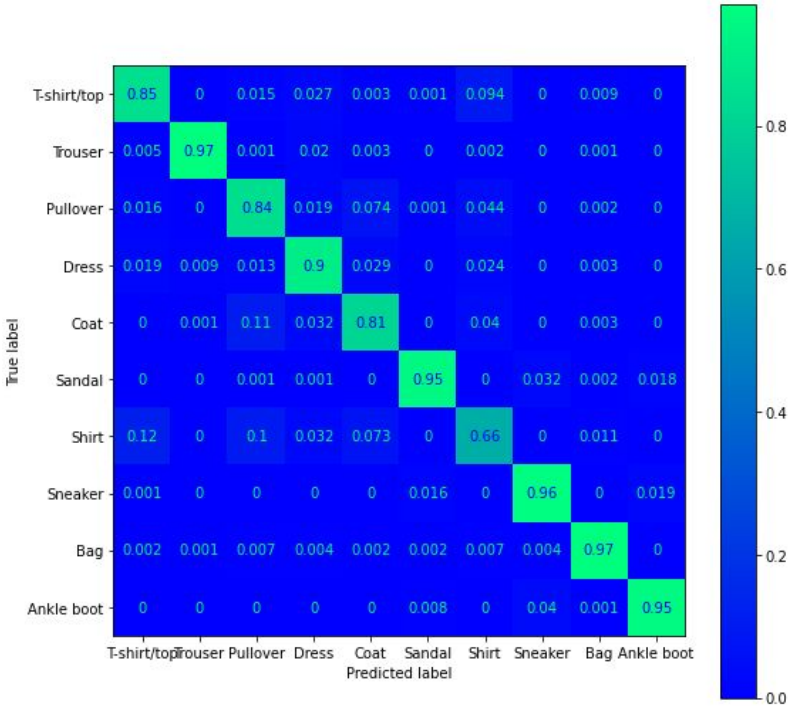
دقت شبکه روی داده تست برابر 0.8828 می‌باشد و هزینه آن مطابق شکل ۲۳ برابر ۱.۰۵می باشد.

حالت سوم. batch size = 256





شکل ۲۵. نمودار هزینه و دقت برای حالت batch size = 256



شکل ۲۶. ماتریس آشفتگی برای حالت batch size = 256

دقت شبکه روی داده تست برابر 0.8873 می‌باشد و هزینه آن مطابق شکل ۲۵ برابر ۱ می باشد.

د) برای حالتی که شبکه به صورت 784 - 500 - 200 - 10 بود و با batch size برابر با ۲۵۶ بهترین نتیجه گرفته شد. جدول ۱ در پایان سوال ۴ پر شده است.

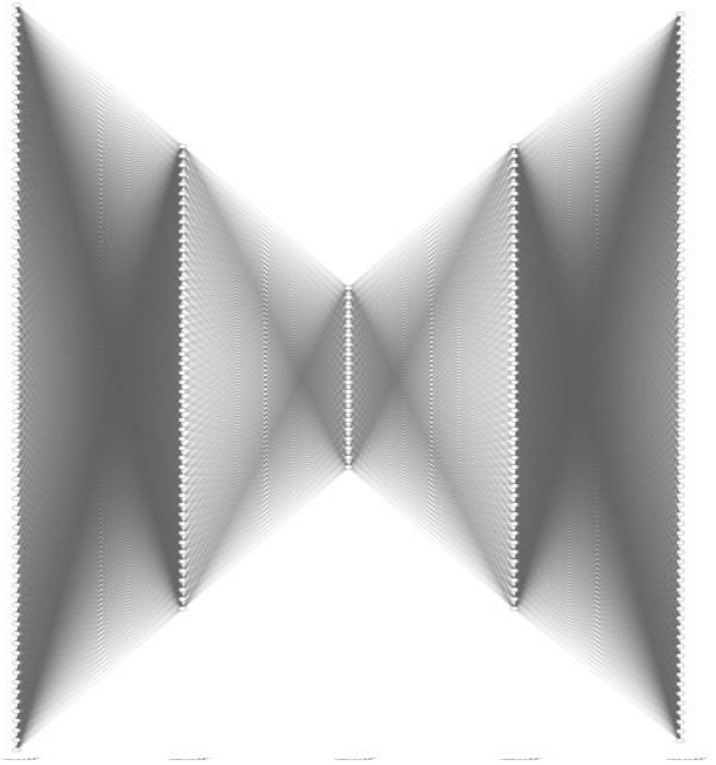
سوال ۴ - dimensionality reduction

در تمامی قسمت های سوال ۴ با توجه به این که در سوال ۳ بهترین شبکه تا بعد ۲۰۰ کاهش پیدا می‌کرد، بعد را ابتدا تا ۲۰۰ کاهش می‌دهیم و با استفاده از یک لایه خطی به بعد ۱۰ (تعداد کلاس ها) می‌آوریم.

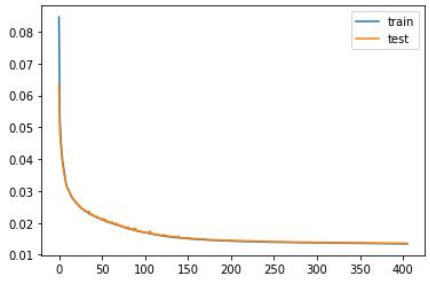
الف) Auto Encoder

برای کاهش بعد از auto encoder استفاده شده است که بعد را تا ۲۰۰ کاهش می‌دهد. ابتدا از ۷۸۴ به ۵۰۰ و سپس به ۲۰۰ کاهش پیدا می‌کند. برای آموزش auto encoder از بهینه‌ساز SGD استفاده شد و تابع هزینه MSELoss انتخاب شد. نمودار هزینه در زمان آموزش در شکل زیر آمده است.

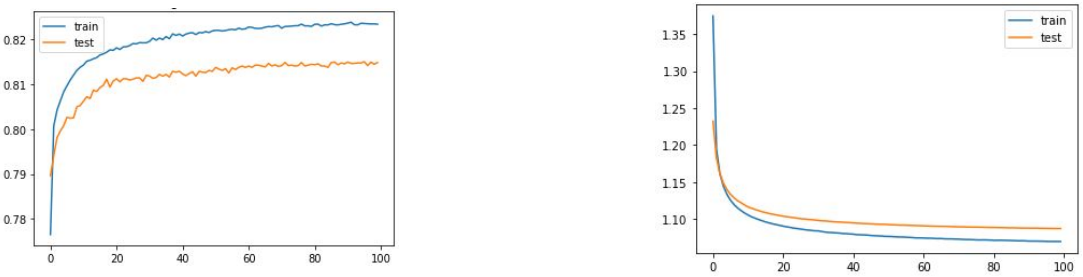
پس از این که autoencoder طراحی شد به پیاده سازی یک لایه از ۲۰۰ به ۱۰ می‌پردازیم که کار طبقه بندی را انجام می‌دهد. برای آموزش این لایه نیز از بهینه‌ساز SGD استفاده شد و تابع هزینه CrossEntropyLoss انتخاب شد.



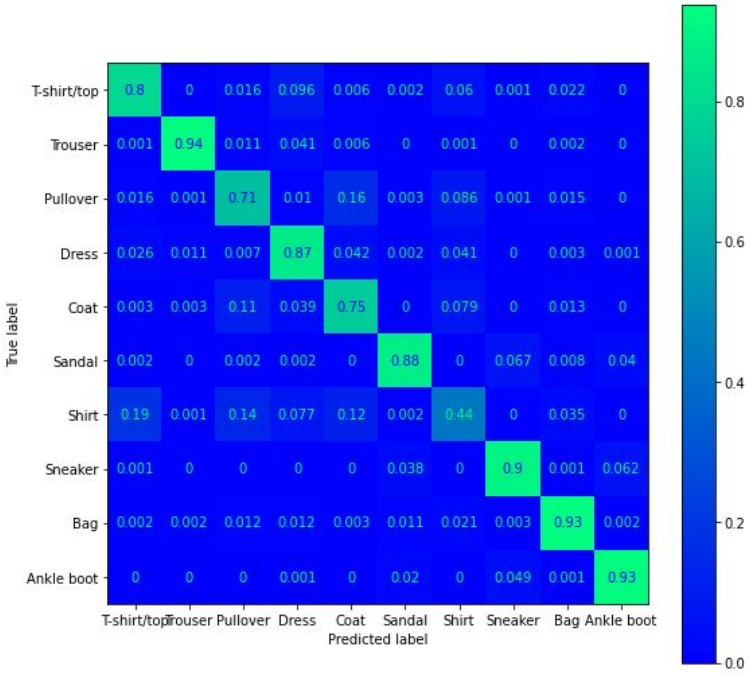
شکل ۲۷. معماری autoencoder



شکل ۲۸. نمودار هزینه در حال آموزش auto encoder



شکل ۲۹. نمودار دقت و هزینه برای شبکه DBN با استفاده از AutoEncoder

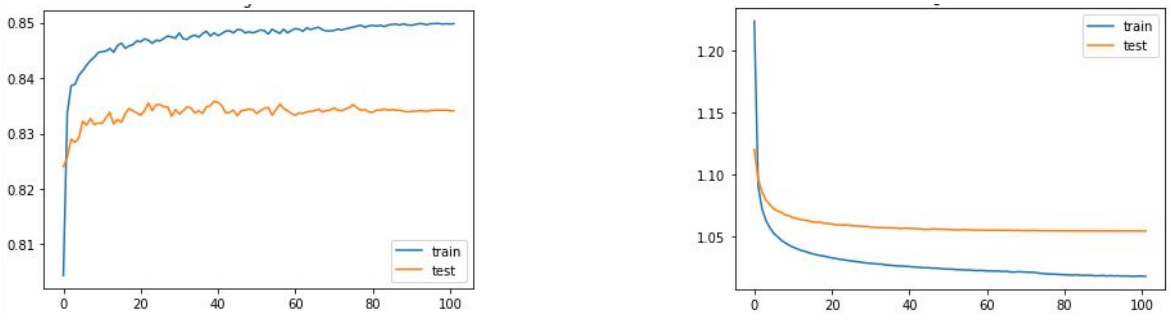


شکل ۳۰. ماتریس آشفتگی برای DBN با استفاده از AutoEncoder

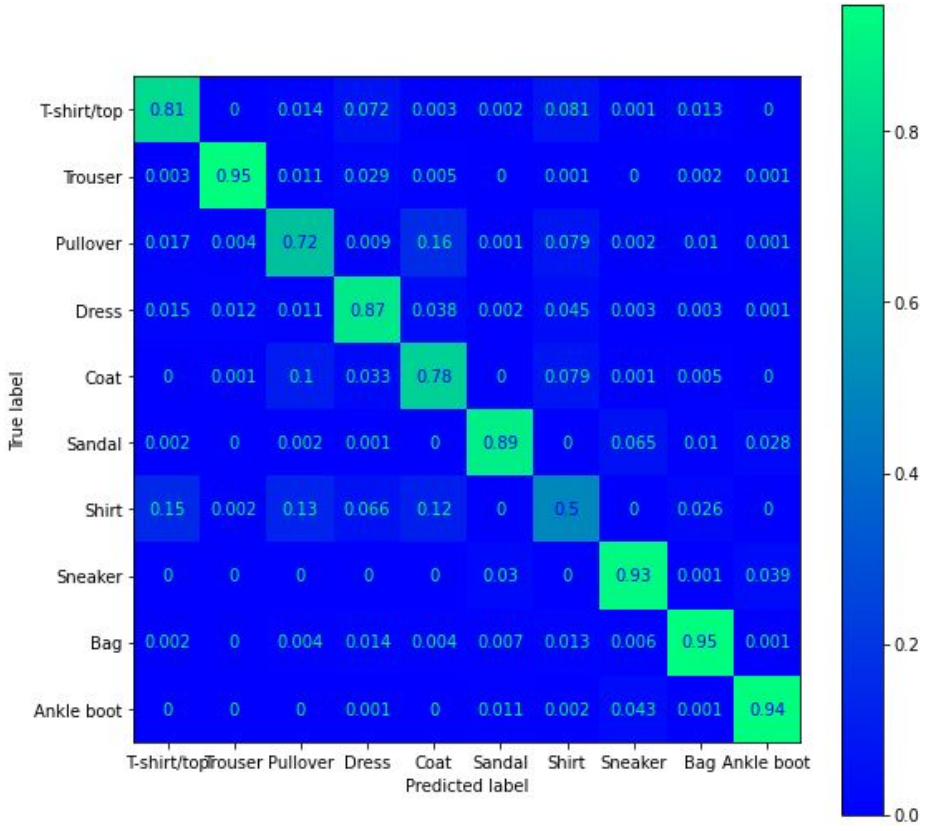
در این شبکه دقت روی داده های تست برابر با 0.8134 شد.

ب) PCA

این روش این‌گونه کار می‌کند که با استفاده از Eigenvalue Decomposition ابتدا جهت هایی که داده بیشترین پراکندگی را دارند بدست می‌آورد. این جهت ها همان بردارهای ویژه ماتریس کواریانس هستند. سپس با استفاده از یک تبدیل خطی پایه ای فضا را بر این بردار ها منطبق می‌کند، در این فرایند سفید سازی داده ها نیز امکان پذیر می‌باشد.



شکل ۳۱. نمودار دقت و هزینه برای شبکه با استفاده از PCA

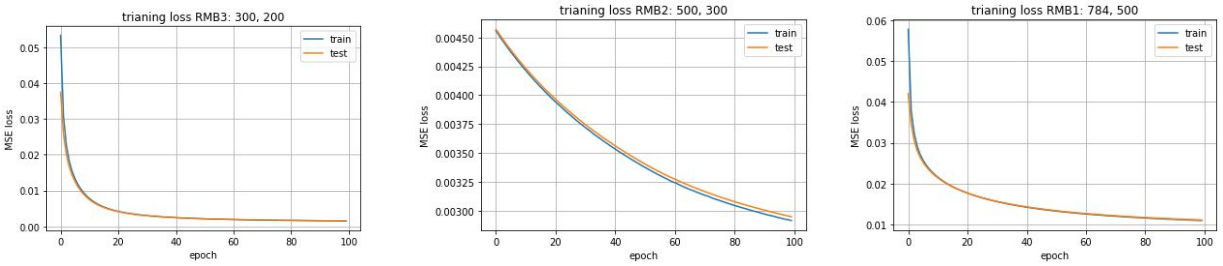


شکل ۳۲. ماتریس آشفتگی برای شبکه با استفاده از PCA

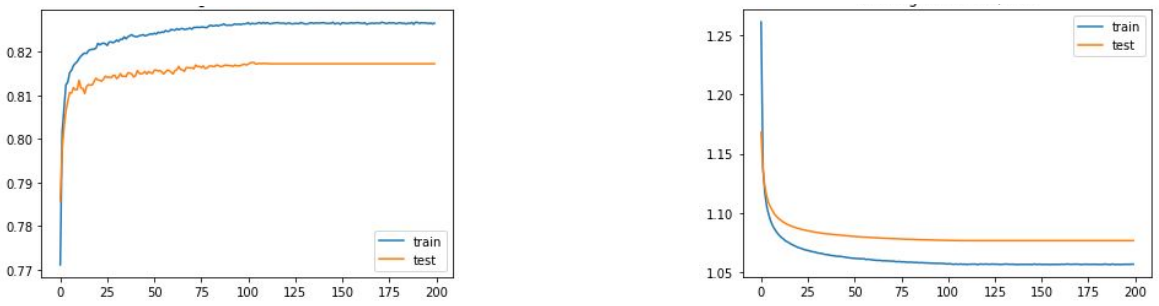
در این شبکه دقت روی داده های تست برابر با 0.8341 شد.

ج) stacked RBMs

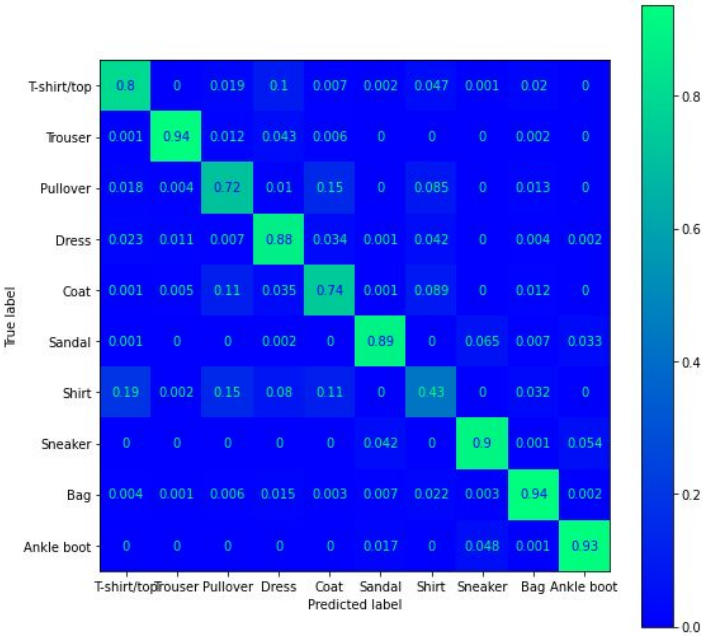
برای این که نتایج این قسمت با قسمت الف قابل مقایسه باشد ۳ RBM را جداگانه آموزش داده شده است که ابعاد با به صورت 784 - 500 - 200 به بعد مورد نظر می‌رسانند. در شکل زیر نمودار هزینه RBM ها در حال آموزش آورده شده است.



شکل ۳۳. نمودار هزینه در آموزش جداگانه RBM ها



شکل ۳۴. نمودار دقت و هزینه برای شبکه DBN با استفاده از Cascaded RMBs



شکل ۳۵. ماتریس آشفتگی برای شبکه DBN با استفاده از Cascaded RBMs

در این شبکه دقت روی داده های تست برابر با 0.8174 شد.

د) مقایسه روش های کاهش بعد

Loss of test	Accuracy on test	method
0.96	0.89	Q3
1.11	0.81	AutoEncoder
1.07	0.83	PCA
1.10	0.81	Cascaded RBM

در مورد عملکرد روش های کاهش بعد می‌توان به این مورد اشاره کرد که PCA روش بسیار مناسبی برای پیدا کردن فضای پوچی است، به این معنی که اگر بعد بردار ویژگی ما در مولفه ای تغییرات نداشته باشد و المان قطری مناسب با آن ویژگی در ماتریس کواریانس برابر ۰ باشد با استفاده از PCA به سادگی قابل تشخیص است. در حالی که PCA تنها می‌تواند پوچی های خطی را از بین ببرد، kernel PCA می‌تواند پوچی های غیر خطی را نیز از بین ببرد.

روش auto encoder نسبت به cascaded rbms بهتر کار کاهش بعد را انجام می‌دهد به این دلیل که تمامی نگاشت های هر لایه با هم آموزش می‌بینند و این روش توانایی مدل کردن نگاشتهای پیچیده تری را دارد.

در پایان در نتایج ما شبکه سوال ۳ پاسخ بهتری گرفت، علت ممکن است این باشد که در این روش ها تنها از یک لایه برای classification استفاده شده است و اگر تعداد لایه های classification را زیاد کنید، محتمل است که نتایج بهتری حاصل شود. این در حالی است که در سوال ۳ چون لایه های مختلف شبکه آزادانه آموزش دیده شده اند، هر لایه علاوه بر کاهش بعد ، وظیفه طبقه بندی را در نظر دارد و بنابراین کل سیستم نتیجه بهتری حاصل می‌کند.

سوال ۵ - مفاهیم

الف)

اگر تعداد داده هایی که شبکه از هر دسته می‌بیند متفاوت باشد (تعداد یکی از دسته ها زیاد تر باشد) شبکه به دلیل این که این داده را بیشتر می‌بیند وزن ها را به گونه ای تغییر می‌دهد که شبکه روی این مجموعه خاص دقت خوبی دارد. برای مثال از ۹۰ درصد داده ها از یک کلاس باشند و شبکه ممکن است به این سمت گرایش پیدا کند که تمامی داده ها را همان معرفی کند زیرا در این حالت دقت ۹۰ درصد خواهد شد. در صورتی که شبکه عملا طبقه بندی مناسبی را انجام نمی‌دهد.

برای رفع این مشکل می‌توان از توابع هزینه ای استفاده کردکه تعداد داده های در هر کلاس را تاثیر می‌دهند. یا روش مناسب دیگر این است که در هر epoch تمامی داده مشاهده نشوند و داده ها به گونه‌ای مشاهده شوند که وزن همسانی داشته باشند.

ب)

خیر،این که شبکه بر روی یک مجموعه داده خاص عملکرد مناسبی داشته باشد نشانه قدرت تعمیم‌دهی بهتر آن شبکه نیست، این حالت ممکن است بر اثر شانس و یا over fitness پیش بیاید.

چیزی که یک شبکه را برتر می‌کند قدرت تعمیم دهی آن است، برای مشاهده این مورد بهتر است از روش های cross validation مانند k fold استفاده شود.

ج)

انتخاب ویژگی های مهم از روش های مختلفی قابل دستیابی است که این روش ها تحت نام feature selection آورده می‌شوند. از این روش ها می‌توان به روش های empirical همانند forward selection اشاره کرد. در این روش مدل های کوچی روی تک‌تک ویژگی ها آموزش داده می‌شود و هر کدام که عملکرد بهتری داشت انتخاب می‌شود و در مرحله بعدی تلاش می‌شود ویژگی انتخاب شود که با ویژگی اول همزمان نتیجه بهتری حاصل نمایند و این فرایند آنقدر ادامه پیدا می‌کند تا اضافه شدن ویژگی جدید عملکرد شبکه را بهبود ندهد.

روش دیگری که برای این کار مرسوم است backward selection است. در این روش بر خلاف حالت قبل ویژگی ها حذف می‌شوند و مدل هایی با ویژگی های جدید آموزش داده می‌شوند. حذف هر کدام از ویژگی هایی که کمترین افت را در عملکرد داشت،آن ویژگی می‌تواند گزینه مناسبی برای حذف باشد. این کار آنقدر ادامه پیدا می‌کند تا حذف هر یک از ویژگی ها باعث افت شدید در عملکرد شبکه شود.

روش های دیگری هم برای feature selection وجود دارد که بر مبنای تحلیل های آماری هستند، از این دسته از روش ها می‌توان به PCA و LDA اشاره کرد. روش information gain هم که در در random forest استفاده می‌شود نیز روش مناسبی است تا تمیز پذیری را بفهمیم.

د)

ماتریس آشفتگی دارای اطلاعات زیادی است، برای مثال نه‌تنها می‌توان دقت را از آن بدست آورد بلکه می‌توان انواع خطا ها را از روی آن بدست آورد مانند (false positive و false negative). علاوه بر این با نرمال سازی های مختلفی که روی این ماتریس می‌توان انجام داد برداشت های مختلفی می‌توان از آن کرد. برای مثال اگر این ماتریس را به صورت سطری نرمال کنیم مقدار اشتباه شدن هر کلاس با دیگر کلاس ها بدست می‌آید. اگر ستونی نرمال کنیم داده ها نشان دهنده خطا در هر یک از تصمیم های طبقه بند هستند .

از ماتریس confusion می‌توان برای بدست آوردن ماتریس confidence استفاده کرد. در خیلی از طبقه بند ها معیاری برای confidence تصمیم وجود دارد، اگر بخواهیم که confidence ماتریس را مناسب (نرمال شده به تعداد انتخاب ها) بدست آوریم به ماتریس confusion نیاز خواهیم داشت.

ه)

نرمال کردن به معنی این است که داده را در رنج ۰ تا ۱ اسکیل کنیم.

استاندارد کردن به این معنی است که داده را منهی میانگین و تقسیم بر انحراف معیار کنیم که در این صورت میانگین برابر با ۰ و واریانس برابر با ۱ خواهد داشت.

اجرای کد

پیاده سازی سوال‌ها در نوتبوک های متفاوت با نام های NNDL_HW2_Qnumber.inpy هستند.
