# تمرین اول شبکه عصبی

**محمدرضا صیدگر-401422215**

Exercise 1)

· Online Learning:

OL یک روش ML است که در آن داده ها به ترتیب در دسترس هستند و ما از آن برای پیش بینی داده های آینده در هر مرحله زمانی استفاده می کنیم. در مقایسه با راه‌حل‌های سنتی ML، یادگیری آنلاین رویکردی اساساً متفاوت است که این واقعیت را در بر می‌گیرد که محیط‌های یادگیری می‌توانند ثانیه به ثانیه تغییر کنند.علاوه بر این، در OL، ما پیش بینی را در زمان واقعی به روز می کنیم. به گفته شای شالو-شوارتز: «OL فرآیند پاسخ دادن به یک سری سؤالات داده شده از پاسخ های صحیح سؤالات قبلی و احتمالاً اطلاعات اضافی موجود است». خانواده OL شامل online convex optimization (که منجر به الگوریتم‌های کارآمد می‌شود)، یک مدل بازخورد محدود زمانی که سیستم مقدار loss را مشاهده می‌کند اما مقدار واقعی loss را مشاهده نمی‌کند، و موارد دیگر است .

· Progressive Learning:

ایده progressive neural networks انتقال دانش در یک سری وظایف به طور موثر است. از نظر مفهومی، شبکه های عصبی progressive دارای سه هدف عمده هستند:

الف) توانایی ترکیب دانش قبلی در هر لایه از سلسله مراتب feature

ب) توانایی استفاده مجدد از محاسبات قدیمی و یادگیری محاسبات جدید

ج) مصونیت در برابر فراموشی

برخلاف مدل‌های transfer learning که دانش قبلی را فقط در زمان اولیه ترکیب می‌کنند، شبکه‌های progressive مجموعه‌ای از مدل‌های از پیش آموزش‌دیده‌شده را در طول آموزش حفظ می‌کنند و اتصالات جانبی را از این مدل‌ها برای استخراج ویژگی‌های مفید برای کارهای جدید یاد می‌گیرند. رویکرد توسعه‌یافته برای یادگیری، ترکیب‌بندی غنی‌تری را به دست می‌آورد و اجازه می‌دهد تا دانش قبلی در هر لایه از سلسله مراتب feature یکپارچه شود.

· Meta-Learning:

Meta-learning در ml به الگوریتم های یادگیری اشاره دارد که از سایر الگوریتم های یادگیری یاد می گیرند.

معمولاً، این به معنای استفاده از الگوریتم‌های ml است که یاد می‌گیرند چگونه پیش‌بینی‌های سایر الگوریتم‌های ml را در زمینه یادگیری گروهی به بهترین شکل ترکیب کنند.

با این وجود، Meta-learning ممکن است به فرآیند دستی model selecting و algorithm tuning نیز اشاره داشته باشد که توسط یک متخصص در یک پروژه ml انجام می‌شود که الگوریتم‌های خودکار ml مدرن به دنبال خودکارسازی آن هستند. همچنین به یادگیری در چندین تکالیف مدل‌سازی پیش‌بینی‌کننده مرتبط، که multi-task learning نامیده می‌شود، اشاره دارد، که در آن الگوریتم‌های Meta-learning یاد می‌گیرند که چگونه یاد بگیرند.

· Self-supervised Learning:

یادگیری Self-supervised learning یک فرآیند ml است که در آن مدل خود را آموزش می دهد تا بخشی از ورودی را از قسمت دیگری از ورودی یاد بگیرد.
در این فرآیند، مشکل بدون نظارت با تولید خودکار برچسب ها به یک مشکل نظارت شده تبدیل می شود. برای استفاده از حجم عظیمی از داده های بدون برچسب، تعیین اهداف یادگیری مناسب برای نظارت از خود داده ها بسیار مهم است.
به عنوان مثال، در NLP، اگر چند کلمه داشته باشیم، با استفاده از یادگیری خود نظارتی می‌توانیم بقیه جمله را کامل کنیم. به طور مشابه، در یک ویدیو، می‌توانیم فریم‌های گذشته یا آینده را بر اساس داده‌های ویدیویی موجود پیش‌بینی کنیم. یادگیری خود نظارتی از ساختار داده‌ها برای استفاده از انواع سیگنال‌های نظارتی در مجموعه داده‌های بزرگ استفاده می‌کند.

· Federated Learning:

یادگیری Federated راهی برای آموزش مدل‌های هوش مصنوعی بدون دیدن یا لمس داده‌های شماست و راهی برای باز کردن قفل اطلاعات برای تغذیه برنامه‌های هوش مصنوعی جدید ارائه می‌دهد.

Exercise 2)

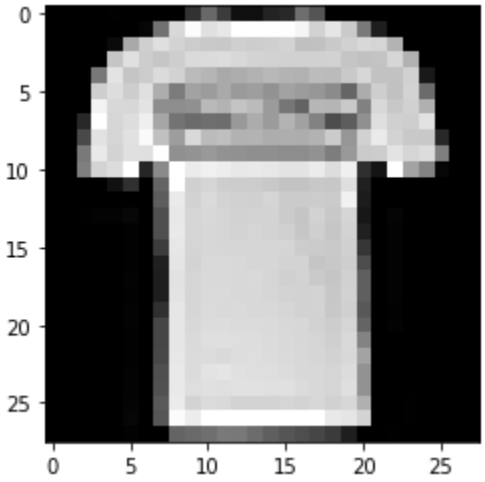
تفاوت اصلی این الگوریتم ها به این صورت است که FGD می اید کل داده های آموزشی را یک جا در نظر می گیرد و توسط همه آن ها کامل آموزش می بیند و در واقع loss ی که بدست میاید میانگین خطا روی کل داده ها است و مدل ما از این میانگین کلی گرادیان می گیرد و سعی میکند وزن ها را اپدیت کند که در این حالت احتمال افتادن در local optimum بیشتر می شود .اما در SGD به این صورت است داده ها را کامل شافل میکند و به صورت تصادفی یکی از داده ها را در هر مرحله می بیند و بر اساس خطای آن loss را محاسبه کرده و گرادیان میگیرد و وزن ها را اپدیت میکند که خب میدانیم اگر تعداد داده ها بسیار زیاد باشد طبیعتا این مدل بسیار سریعتر عمل خواهد کرد نسبت به FGD چون آن الگوریتم محاسبات را پیچیده و بسیار سخت تر می کند و از طرفی هم SGD ممکن است خطر افتادن در local optimum را هم کاهش دهد چون ممکن است یک داده ناگهان ما را از روی یک دره کوچک پرتاب کند و نزدیک کند به global optimum اما از طرفی هم این الگوریتم معمولا با خطای زیادی همراه است به همین دلیل حتی ممکن است در پیدا کردن global optimum نیز خوب نباشد. در اخر هم MBGD هم به این شکل است که داده ها را batch batch در نظر می گیرد و از میانگین خطای یک batch میاد loss را بدست می اورد چون گرادیان را برای تعدادی داده داریم محاسبه میکنیم که این تعداد نه مثل FGD زیاد است که محاسبات سخت شود و نه مثل SGD کم است که خطا زیاد باشد بنابراین امکان ایجاد نویز کمترمیشود و بدین ترتیب
باعث میشود تا بتوانیم global optimum را دقیق تر پیدا کنیم. یک جورایی می توان گفت این الگوریتم متوسط دو الگوریتم بالا است ولی نکته منفی این الگوریتم است که هایپرپارامتر جدیدی دارد که همان batch size است که مهم است چند در نظر گرفته می شود.
از نظر من برای آنلاین لرنینگ بهترین گزینه MNGD است چون یه تعدادی از داده ها را می ببیند و خطا بدست میاید و learn می شود.

Exercise 3)

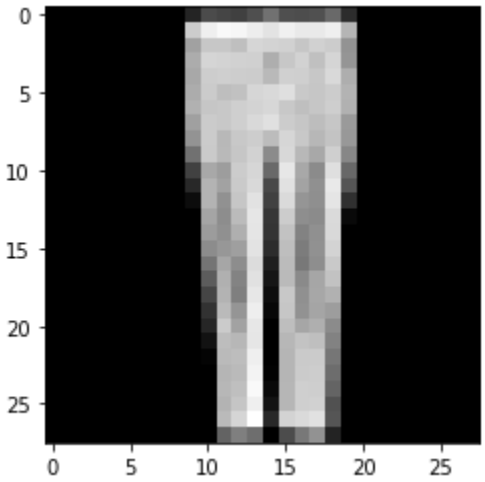
گزارش پروژه:

در این پروژه داده های fashion mnist را از کگل دریافت کرده و یک کلاس برای داده ها درست کرده تا ماتریس عکس را برای ما اماده و لیبل را هم جدا کند.

در ادامه چند تا از عکس ها را چاپ کرده و لیبل آن را بررسی کردیم:



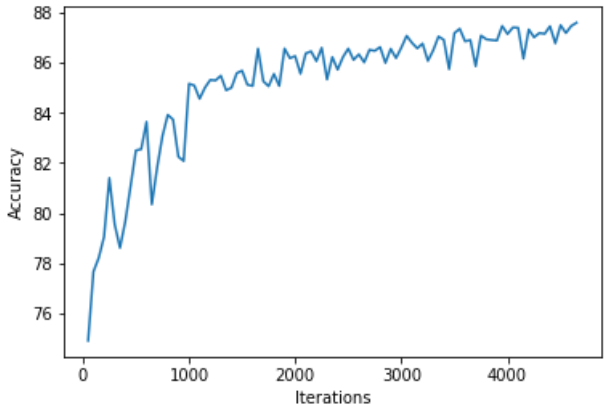
Label = T-shirt/Top

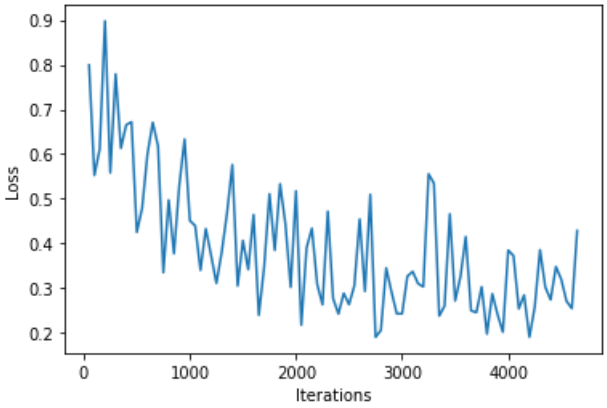


Label = Dress

labels:  Pullover, Ankle Boot, Shirt, T-shirt/Top, Dress, Coat, Coat, Sandal, Coat, Bag

در نهایت اولین مدل را ساختیم که شامل 3 لایه نورون است لایه اول ورودی 28*28 را گرفته به صورت vector ولایه بعدی که لایه hidden است دارای 512 نورون است و لایه اخر هم که لایه خروجی است دارای 10 نورون است زیرا که کلا 10 دسته بندی مختلف داریم.
نرخ یادگیری 0.001 در نظر گرفته و برای محاسبه loss هم crossentropy استفاده کردیم. 5 ایپاک در نظر گرفته و مدل را ترین کرده و روی داده های تست ارزیابی کردیم که نتیجه در نهایت loss = 0.22 و accuracy = 87.36. نمودار های زیر نتایج رو به خوبی در طول آموزش نشان می دهد.



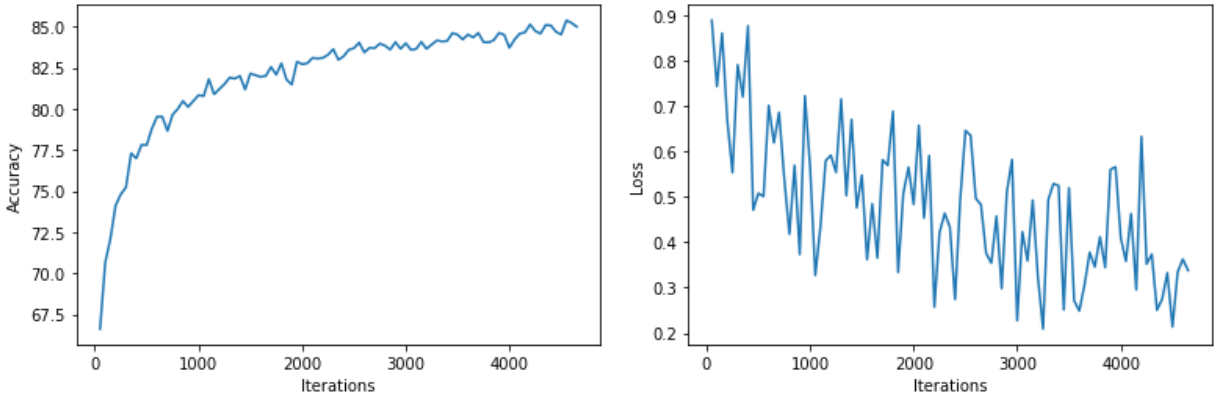

درآخر بررسی کردیم که مدل ما هر کلاس از داده ها را چطور پیشبینی می کند:

Accuracy of T-shirt/Top: 81.64%
Accuracy of Trouser: 98.44%
Accuracy of Pullover: 73.64%
Accuracy of Dress: 85.83%
Accuracy of Coat: 91.80%
Accuracy of Sandal: 93.00%
Accuracy of Shirt: 54.44%
Accuracy of Sneaker: 90.12%
Accuracy of Bag: 96.00%
Accuracy of Ankle Boot: 96.48%

همانطور که می بینیم مدل ما شلوار و کیف و نیم بوت را عالی تشخیص میدهد ولی از طرفی پیراهن را اصلا خوب تشخیص نمیدهد (تقریبا شانسی عمل میکند) و پلیور هم درصد

پایین تری دارد نسبت به بقیه و احتمالا چون این ها از نظر ظاهری شبیه هستن مدل ما هنوز خوب فرق این ها را تشخیص نمیدهد.

در مدل بعدی همان مدل قبلی استفاده کرده ولی چند تا لایه dropout را به آن اضافه میکنیم برای اینکه ببینیم چه اتفاقی در نتیجه رخ می دهد در نهایت Loss: 0.21 و Accuracy: 84.51 .که درصد درستی کمتر شد نسبت به قبل ولی loss فرق زیادی نکرد نمودار های زیر نتایج رو به خوبی در طول آموزش نشان می دهد.



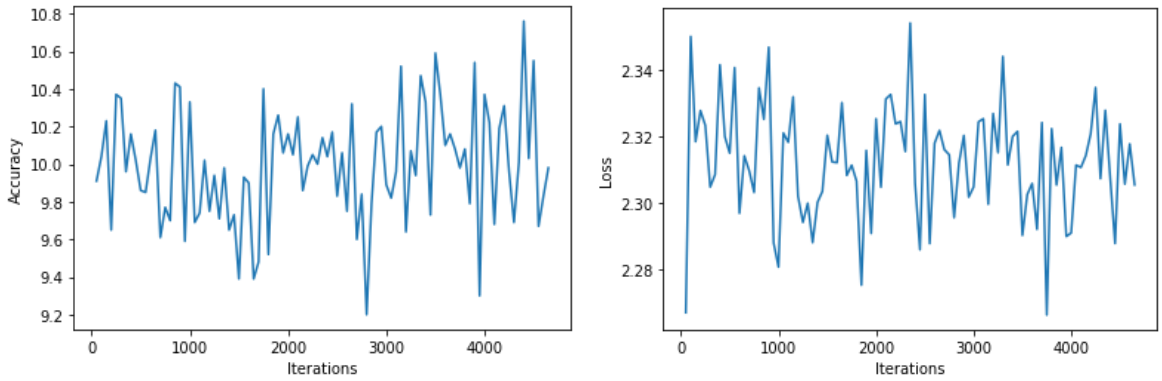

درآخر بررسی کردیم که مدل ما هر کلاس از داده ها را چطور پیشبینی می کند:

Accuracy of T-shirt/Top: 80.08%
Accuracy of Trouser: 96.46%
Accuracy of Pullover: 74.36%
Accuracy of Dress: 88.97%
Accuracy of Coat: 77.46%
Accuracy of Sandal: 93.03%
Accuracy of Shirt: 48.92%
Accuracy of Sneaker: 91.98%
Accuracy of Bag: 95.31%
Accuracy of Ankle Boot: 93.82%

در اینجا حتی مدل از شانس هم بدتر عمل میکند در تشخیص پیراهن.

در مدل بعدی هم یه سری تغییرات ایجاد کردیم در لایه ها و نورون ها که ببینیم چه تغییری حاصل می شود که این تغییرات به این شکل است که 4 لایه hidden داریم که لایه اول hidden دارای 512 نورون است و 2 لایه بعدی 480 نورون و لایه بعدیش هم 256

نورون دارد و مثل قبل لایه ورودی 28*28 نورون و لایه خروجی 10 نورون دارد. و بین هر لایه هم dropout قرار گرفته و تابع فعالیت به کار گرفته شده هم sigmoid می باشد.
نتایج این مدل بسیار بد بود و در اخرین مرحله Accuracy: 10.55 و Loss: 2.32 ثبت شد.



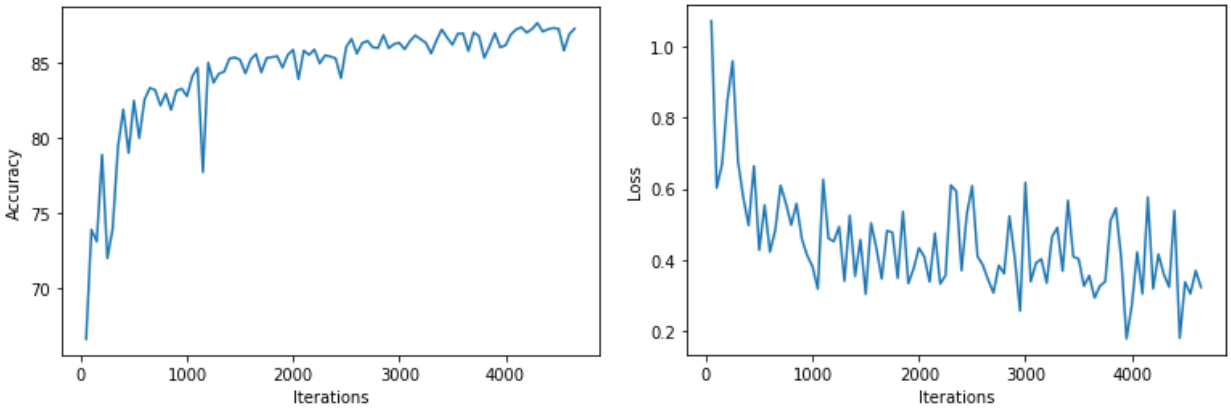

درآخر بررسی کردیم که مدل ما هر کلاس از داده ها را چطور پیشبینی می کند:

Accuracy of T-shirt/Top: 7.59%
Accuracy of Trouser: 8.70%
Accuracy of Pullover: 12.03%
Accuracy of Dress: 14.94%
Accuracy of Coat: 9.20%
Accuracy of Sandal: 10.73%
Accuracy of Shirt: 10.98%
Accuracy of Sneaker: 8.70%
Accuracy of Bag: 16.17%
Accuracy of Ankle Boot: 8.09%

که ایندفعه مدل ما در تشخیص شلوار و بوت از همه بدتر عمل کرد.

در مدل بعدی هم یه سری تغییرات ایجاد کردیم در لایه ها و نورون ها که ببینیم چه تغییری حاصل می شود که این تغییرات به این شکل است که 4 لایه hidden داریم که لایه اول hidden دارای 512 نورون است و 2 لایه بعدی 256 نورون و لایه بعدیش هم 128نورون دارد و مثل قبل لایه ورودی 28*28 نورون و لایه خروجی 10 نورون دارد.
و ایندفعه dropout قرار ندارد بین لایه چون نتیجه مثبتی را نداشت و تابع فعالیت به کار گرفته شده ایندفعه relu می باشد.

نتایج این مدل  بهتر شد و در اخرین مرحله Accuracy: 87.20 و Loss: 0.33 ثبت شد.

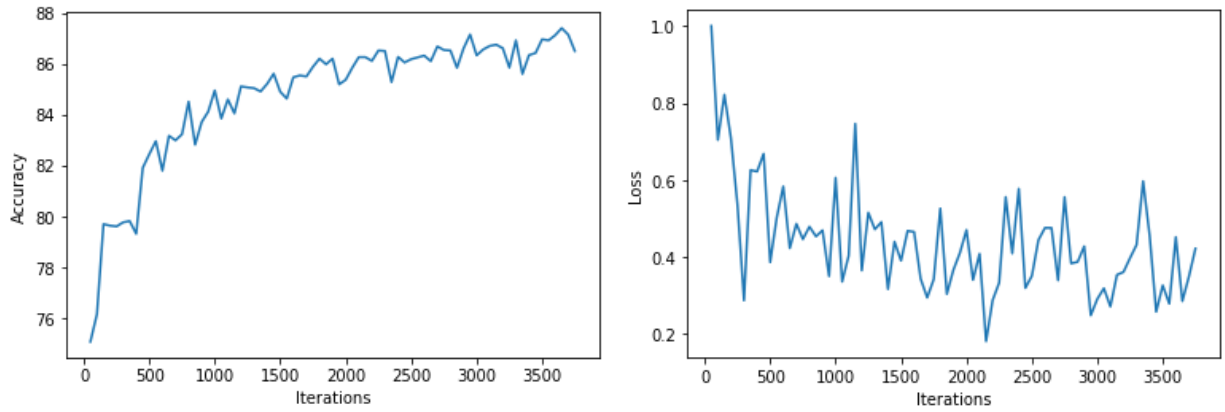



درآخر بررسی کردیم که مدل ما هر کلاس از داده ها را چطور پیشبینی می کند:

Accuracy of T-shirt/Top: 64.87%
Accuracy of Trouser: 98.44%
Accuracy of Pullover: 76.64%
Accuracy of Dress: 88.51%
Accuracy of Coat: 88.26%
Accuracy of Sandal: 90.94%
Accuracy of Shirt: 72.40%
Accuracy of Sneaker: 94.09%
Accuracy of Bag: 97.24%
Accuracy of Ankle Boot: 93.50%

ایندفعه تشخیص پیراهن بهتره شده ولی تشخیص تیشرت ضعیف تر شده است.

در مدل بعدی شبکه ما دوباره همان شبکه اولیه ای است که داشتیم فقط این دفعه در یادگیری یک پنالتی ترم برای loss در نظر گرفتیم که regularization را انجام میدهد که مشابه ridge regularization میباشد چون توان دوم را در نظر می گیرد.
نتایج این مدل  بد نبود و در اخرین مرحله Accuracy: 86.95 و Loss: 0.32 ثبت شد.

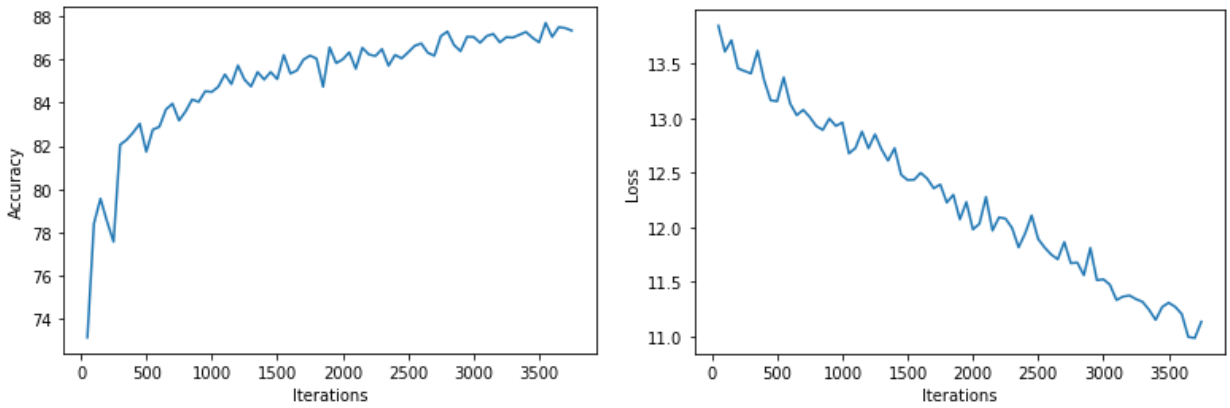



درآخر بررسی کردیم که مدل ما هر کلاس از داده ها را چطور پیشبینی می کند:

Accuracy of T-shirt/Top: 77.99%
Accuracy of Trouser: 96.97%
Accuracy of Pullover: 75.00%
Accuracy of Dress: 88.51%
Accuracy of Coat: 73.73%
Accuracy of Sandal: 97.14%
Accuracy of Shirt: 74.30%
Accuracy of Sneaker: 89.58%
Accuracy of Bag: 94.62%
Accuracy of Ankle Boot: 92.86%

با بررسی این درصد ها متوجه می شویم که ایندفعه مدل ما واقعا بهتر شده حتی با اینکه در درستی کلی پیشرفتی نداشته ولی الان هیچ کلاسی نیست که درستی کمتر از 70 درصد داشته باشد و این همان هدف ماست در این پروژه.

در مدل بعدی شبکه ما دوباره همان شبکه اولیه ای است که داشتیم فقط این دفعه در یادگیری L1 regularization را انجام میدهد.
نتایج این مدل هم بد نبود و در اخرین مرحله Accuracy: 86.79 و Loss: 11.31 ثبت شد که نشان میدهد پنالتی ترم اضافه شده خیلی تاثیر داشته در loss.

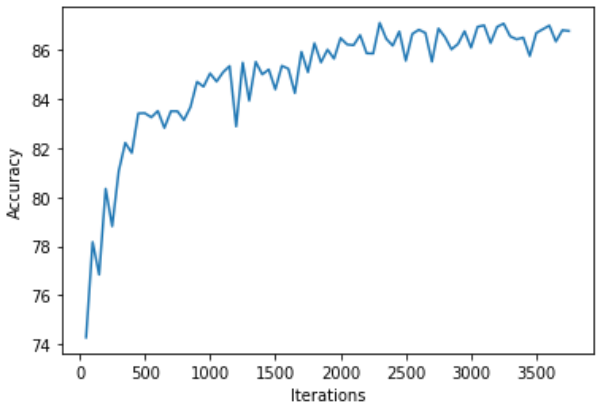


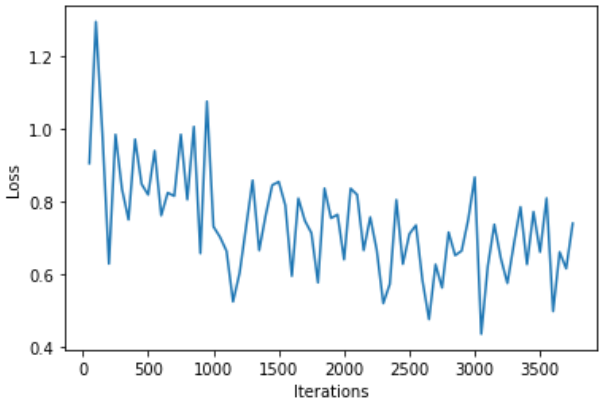

درآخر بررسی کردیم که مدل ما هر کلاس از داده ها را چطور پیشبینی می کند:

Accuracy of T-shirt/Top: 87.55%
Accuracy of Trouser: 95.69%
Accuracy of Pullover: 78.00%
Accuracy of Dress: 91.73%
Accuracy of Coat: 85.41%
Accuracy of Sandal: 90.55%
Accuracy of Shirt: 62.50%
Accuracy of Sneaker: 94.44%
Accuracy of Bag: 95.55%
Accuracy of Ankle Boot: 93.78%

در این مدل بازهم پیراهن درصد خوبی را ندارد.

در مدل بعدی شبکه ما دوباره همان شبکه اولیه ای است که داشتیم فقط این دفعه در یادگیری L2 regularization را انجام میدهد.
نتایج این مدل هم بد نبود و در اخرین مرحله Accuracy: 86.69 و Loss: 0.66 ثبت شد.



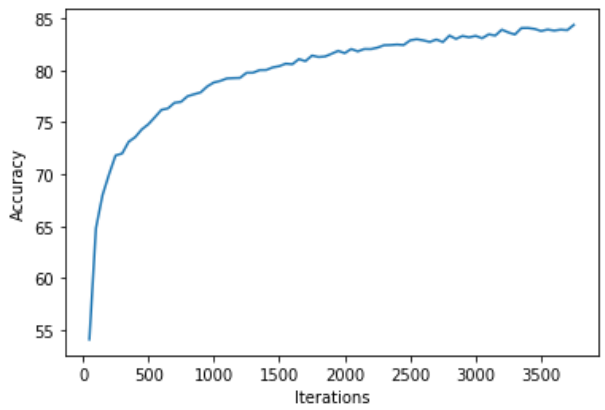



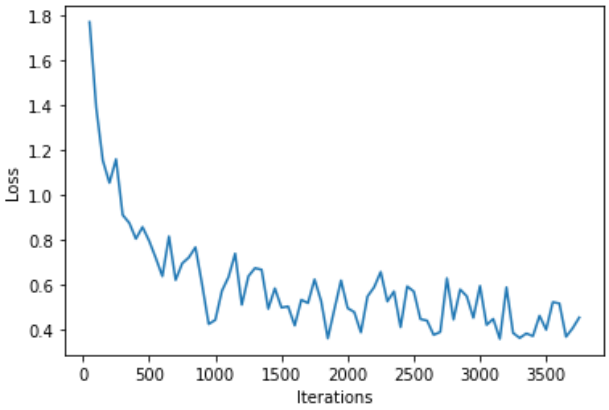

درآخر بررسی کردیم که مدل ما هر کلاس از داده ها را چطور پیشبینی می کند:

Accuracy of T-shirt/Top: 74.79%
Accuracy of Trouser: 97.53%
Accuracy of Pullover: 86.89%
Accuracy of Dress: 93.42%
Accuracy of Coat: 82.12%
Accuracy of Sandal: 95.67%
Accuracy of Shirt: 65.59%
Accuracy of Sneaker: 92.24%
Accuracy of Bag: 92.62%
Accuracy of Ankle Boot: 97.23%

باز هم این مدل تغییر خاصی در تشخیص پیراهن نداشت.

در اخرین مدل هم شبکه ما دارای لایه های batch normalization است. نرمال سازی دسته ای (BN) برای بهبود عملکرد مدل، کاهش تغییر internal covariate و اعمال یک اثر regularization کوچک شناخته شده است.
نتایج این مدل اما کمی ضعیف تر بود و در اخرین مرحله Accuracy: 83.76 و Loss: 0.39 ثبت شد.




درآخر بررسی کردیم که مدل ما هر کلاس از داده ها را چطور پیشبینی می کند:

Accuracy of T-shirt/Top: 78.57%
Accuracy of Trouser: 95.63%
Accuracy of Pullover: 74.74%
Accuracy of Dress: 84.43%
Accuracy of Coat: 81.86%
Accuracy of Sandal: 89.58%
Accuracy of Shirt: 54.22%
Accuracy of Sneaker: 90.04%
Accuracy of Bag: 94.14%
Accuracy of Ankle Boot: 91.06%

باز هم می بینیم که مدل ما در تشخیص پیراهن مشکل دارد

و درکل بهترین مدل ما همان مدل با ساختار شبکه اولیه و regularization که داخل سوال گفته شد بود که هم درستی خوبی داشت و هم همه کلاس ها با درصد خوبی پیشبینی میشدند.