سوال یک

۱. الف) بیش برازش (Overfitting)

بیش برازش رخ می‌دهد وقتی شبکه عصبی به گونه‌ای آموزش ببیند که بیش از حد به داده‌های آموزش خود پاسخ دهد و قدرت تعمیم‌پذیری آن برای داده‌های جدید کاهش یابد. علت اصلی بیش برازش این است که مدل پیچیده شبکه قادر است روابط پیچیده‌تری را در داده‌های آموزش کشف کند که ممکن است برای داده‌های جدید و ناشناخته نامناسب باشد.

علامت‌های بیش برازش عبارتند از: دقت بالا در داده‌های آموزش و دقت پایین در داده‌های ارزیابی، اختلاف بزرگ بین خطاهای آموزش و ارزیابی، و شبکه‌ی عصبی حفظ جزئیات ناهمخوان و نویز داده‌های آموزش.

برای کاهش بیش برازش، می‌توان از روش‌های زیر استفاده کرد:

- جمع‌آوری داده بیشتر: افزایش تنوع و تعداد داده‌های آموزش می‌تواند به کاهش بیش برازش کمک کند.
- کاهش پیچیدگی مدل: استفاده از مدل‌های ساده‌تر، تعداد لایه‌ها و نورون‌ها را کاهش داده و قیدهای مناسبی روی پارامترها اعمال کرد.
- استفاده از روش‌های منظم‌سازی: مانندDropout ، L1و L2

۲. کم برازش (Underfitting)

کم برازش رخ می‌دهد وقتی شبکه عصبی به دلیل کمبود قدرت مدل‌سازی، قادر به یادگیری ساختارهای پیچیده در داده‌های آموزش نمی‌باشد. به عبارت دیگر، شبکه نتوانسته است الگوهای کلی داده‌ها را یاد بگیرد و در نتیجه در داده‌های آموزش و ارزیابی دقت پایینی داشته باشد.

علامت‌های کم برازش عبارتند از: دقت پایین در داده‌های آموزش و ارزیابی، اختلاف کم بین خطاهای آموزش و ارزیابی و عدم توانایی شبکه در تقلید الگوهای مهم موجود در داده‌های آموزش.

برای کاهش کم برازش، می‌توان از روش‌های زیر استفاده کرد:

- جمع‌آوری داده بیشتر: افزایش تنوع و تعداد داده‌های آموزش می‌تواند به بهبود کم برازش کمک کند.
- افزایش پیچیدگی مدل: استفاده از مدل‌های پیچیده‌تر با افزایش تعداد لایه‌ها و نورون‌ها، متغیرهای مخفی بیشتر و ظرفیت مدل بیشتر.

- استفاده از روش‌های افزایش داده: مانند افزایش نمونه‌های داده، تکنیک‌های Augmentation
- انتخاب معماری مناسب

ب) برای تشخیص بیش برازش در یک مدل که قبلاً آموزش دیده است، می‌توانید از روش‌های زیر استفاده کنیم:

۱. مقایسه خطا در داده‌های آموزش و ارزیابی: یکی از علائم بیش برازش، اختلاف بزرگ بین خطاهای داده‌های آموزش و ارزیابی است. اگر خطای داده‌های آموزش بسیار کم و خطای داده‌های ارزیابی بیشتر است، احتمالاً مدل دچار بیش برازش شده است.

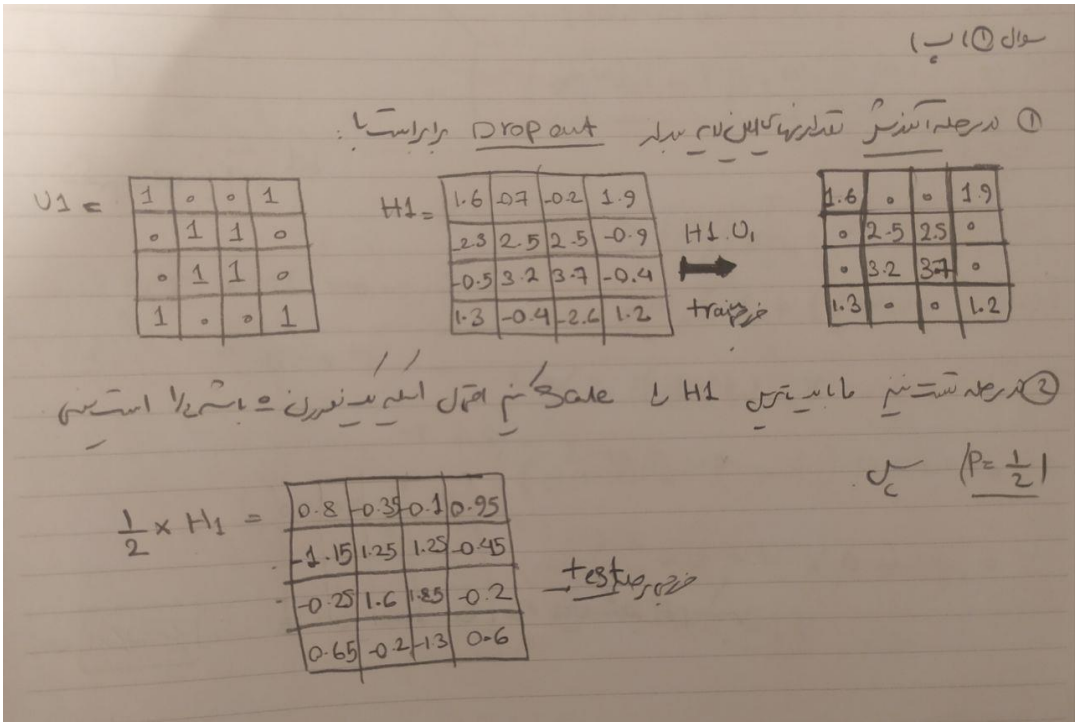
۲. تغییرات دقت در طول آموزش: بیش برازش ممکن است باعث کاهش دقت در داده‌های ارزیابی شود. در حالی که دقت در داده‌های آموزش ممکن است به طور مداوم افزایش یابد، دقت در داده‌های ارزیابی به طور ناپیوسته کاهش یافته و حتی ممکن است بهبود نیابد.

۳. استفاده از داده‌های ارزیابی جدید: اگر مدل را بر روی داده‌های ارزیابی جدیدی تست کنید و دقت آن در داده‌های ارزیابی جدید نسبت به داده‌های آموزش به طور قابل ملاحظه‌ای کاهش یابد، ممکن است مدل دچار بیش برازش شده باشد.

۴. استفاده از روش‌های ارزیابی متعدد: می‌توانید از روش‌های ارزیابی متعدد مانند اعتبارسنجی متقابل (cross-validation)استفاده کنید. این روش‌ها به شما کمک می‌کنند تا مدل را بر روی مجموعه‌های آموزش و ارزیابی مختلف آموزش دهید و عملکرد آن را مقایسه کنید. اگر مدل در مجموعه‌های ارزیابی مختلف نتایج نامناسبی داشته باشد، ممکن است بیش برازش داشته باشد.

پ)

از Dropout طبق اسلایدها استفاده کردم.



سوال دو

الف)

۱. بایاس (Bias) بایاس به میزان اختلاف بین مقدار پیش‌بینی شده توسط الگوریتم و مقدار واقعی برچسب نمونه‌ها اشاره دارد. با افزایش مقدارK ، تعمیم‌دهی بیشتری در الگوریتم ایجاد می‌شود و در نتیجه بایاس کاهش می‌یابد. زیرا به جای تصمیم‌گیری بر اساس تعداد کمی از همسایگان، اکثریت همسایگان برای تصمیم‌گیری مورد استفاده قرار می‌گیرند. این می‌تواند منجر به تصمیم‌گیری درست‌تر و کاهش بایاس شود.

۲. واریانس(Variance) میزان تغییرات پیش بینی مدل را نشان میدهد. با افزایش مقدارK ، تعداد بیشتری از همسایگان در محاسبات مورد استفاده قرار می‌گیرند و در نتیجه واریانس افزایش می‌یابد. زیرا این تصمیم‌ها بر اساس نمونه‌های بیشتری اتخاذ می‌شوند و در نتیجه پیش‌بینی‌ها می‌تواند بیشتر پراکنده شود.

به طور کلی، با افزایش مقدار K در الگوریتم نزدیک‌ترین همسایگی، بایاس کاهش می‌یابد و امکان تصمیم‌گیری درست‌تر و تعمیم‌پذیری بیشتر فراهم می‌شود. اما در عین حال، واریانس افزایش می‌یابد و پیش‌بینی‌ها ممکن است پراکنده‌تر شوند.

ب)

استفاده از منظم سازی، ممکن است باعث تضعیف عملکرد مدل شود:درست

منظم‌سازی یک روش استفاده می‌شود تا از بیش‌برازش (overfitting) در مدل‌های یادگیری ماشین جلوگیری کند. بیش‌برازش به وقوع می‌پیوندد وقتی که مدل به رو داده های اموزشی عملکرد خوبی دارد و نمی‌تواند به طور عمومی الگوها را تعمیم دهد اصطلاحا داده های اموزشی را حفظ کرده است. با استفاده از منظم سازی، وزن‌های مدل کنترل می‌شوند و از پیچیدگی زیاد مدل جلوگیری می‌شود. اما در برخی موارد، استفاده از منظم سازی می‌تواند باعث تضعیف عملکرد مدل شود، به خصوص زمانی که وزن‌ها به طور غیرمنطقی کاهش یابند و مدل قدرت تعمیم‌دهی خود را از دست دهد

اضافه کردن تعداد زیاد ویژگی‌های جدید، باعث جلوگیری از بیش‌برازش می‌شود:نادرست(البته بستگی دارد)

بسیار بستگی ارد چه نوع ویژگی هایی اضافه شود. اضافه کردن ویژگی‌هایی که اطلاعات معنی‌داری را ندارند یا ارتباط ضعیفی با مسئله دارند، ممکن است منجر به افزایش پیچیدگی مدل و بیش‌برازش شود. اگر یک ویژگی به تنهایی یا با ویژگی‌های دیگر قادر به تفکیک داده‌ها و تشخیص الگوهای مهم است، آنگاه افزودن آن می‌تواند بهبودی در عملکرد مدل داشته باشد. اما اگر ویژگی به تنهایی اطلاعات کمی ارائه می‌دهد یا با ویژگی‌های دیگر تداخل دارد، اضافه کردن آن ممکن است منجر به افزایش پیچیدگی زیاد و بیش‌برازش شود.

با زیاد کردن ضریب منظم‌سازی، احتمال بیش‌برازش بیشتر می‌شود: نادرست

ضریب منظم‌سازی در واقع یک عامل مهم در کنترل بیش‌برازش است. با افزایش ضریب منظم‌سازی، مدل تمایل به استفاده کمتر از ویژگی‌های پیچیده و تنظیمات پیچیده دارد و در نتیجه، به سمت یادگیری الگوهای ساده‌تر و کمتر پیچیده متمایل می‌شود. این باعث کاهش احتمال بیش‌برازش می‌شود و مدل قدرت تعمیم‌دهی خود را افزایش می‌دهد.

به عبارت دیگر، با افزایش ضریب منظم‌سازی، مدل به سمت یک حالت متوازن بین بیش‌برازش و کم‌برازش، به نام تعمیم‌پذیری مناسب (appropriate generalization) تمایل دارد. این به معنی این است که مدل قادر به تعمیم الگوهای مشترک و معنی‌دار در داده‌های آموزشی و داده‌های جدید است، به جای یادگیری جزئیات تصادفی و نویزهای موجود در داده‌های آموزشی می‌باشد.

ج)

Wexp1 = [0.26, 0.25, 0.25, 0.25]:

در اینجا از منظم سازی L2 استفاده شده است در منظم سازی L2 باعث میشود وزن ها به سمت صفر میل پیدا کنند ولی دقیقا صفر نمی شوند دلیل این است که وزن هایی که وزن بیشتری دارند بیشتر جریمه میشوند برعکس L1 که وزن ها بی توجه به مقدارشان جریمه میشوند.

W exp2 = [1, 0, 0, 0]:

در این آزمایش، مقدار وزن‌ها برای ویژگی اول برابر با ۱ و برای سایر ویژگی‌ها برابر با صفر است. این مقادیر نشان می‌دهد که احتمالاً از منظم‌سازی L1 استفاده شده است. منظم‌سازی L1 باعث کم شدن برخی وزن‌ها به صفر می‌شود و در نتیجه، برخی از ویژگی‌ها را حذف می‌کند و تعمیم‌پذیری مدل را افزایش می‌دهد.

Wexp3 = [13.3, 23.5, 53.2, 5.1]:

در این آزمایش، مقادیر وزن‌ها برای تمامی ویژگی‌ها بسیار بزرگ است. این مقادیر نشان می‌دهد که از هیچ کدام استفاده نشده است.

Wexp4 = [0.5, 1.2, 8.5, 0]:

در اینجا نیز از L1 استفاده شده است چون L2 تمایل دارد همه مقادیر کوچک کند ولی همه مقادیر کوجک نمی کند ولی در L1 بعضی از مقادیر را ۰ می کند.

سوال سه

الف)

فرایند تقطیر دانش (Knowledge Distillation) یک روش در یادگیری ماشین است که برای انتقال دانش از یک مدل پیچیده‌تر به یک مدل ساده‌تر استفاده می‌شود. هدف اصلی این روش، انتقال دانش و قابلیت‌های یک مدل پیچیده به یک مدل ساده‌تر و کوچکتر است.

در فرایند تقطیر دانش، دو مدل مشارکت دارند: مدل پیچیده معروف به مدل معلم مدل ساده معروف به مدل شاگرد مدل معلم معمولاً یک مدل عمیق و پیچیده‌تر است که دقت بالا و توانایی خوبی در تشخیص و وظایف مورد نظر دارد. اما مدل شاگرد کوچکتر و ساده‌تر است و تعداد پارامترهای کمتری دارد.

فرایند تقطیر دانش به این صورت انجام می‌شود که مدل معلم با استفاده از داده‌های آموزش، پاسخ‌ها و احتمالات خروجی را تولید می‌کند. سپس این احتمالات و خروجی‌ها به عنوان برچسب‌ها به مدل شاگرد داده می‌شوند. مدل

شاگرد با داشتن این برچسب‌ها، تلاش می‌کند تا بهترین تطابق را با خروجی‌های مدل معلم داشته باشد و دانش موجود در مدل معلم را تقلید کند.

استفاده از فرایند تقطیر دانش چندین مزیت دارد. اولاً، به دلیل استفاده از مدل معلم، می‌توان از دانش و قابلیت‌های مدل پیچیده برای آموزش مدل ساده استفاده کرد. دوماً، مدل ساده با تعداد کمتری پارامتر، سریعتر آموزش می‌بیند و در نتیجه زمان و منابع کمتری می‌برد. سوماً، مدل ساده ممکن است برای استفاده در سیستم‌های با منابع محدود مفید باشد.

ب)

۱. Teacher Modelمدل اولیه، که اغلب یک مدل بزرگ و پیچیده است و ما می‌خواهیم از آن دانش استخراج کنیم.

۲. Student Modelمدل جدید با تعداد کمتری پارامتر که می‌خواهیم با استفاده از تقطیر دانش آن را آموزش دهیم. هدف از این مدل، تقلید از رفتار مدل استاد است.

۳. Soft Labelsاحتمالات خروجی مدل استاد وقتی که softmax با دمایی بزرگتر از ۱ (T > 1) استفاده می‌شود .توزیع‌های احتمال بر روی کلاس‌ها هستند و به صورت بردارهای one-hot نیستند.

۴. Soft Predictionsاحتمالات خروجی مدل دانشجو وقتی که softmax با دمایی بزرگتر از ۱ (T > ۱) استفاده می‌شود. این احتمالات هدف دارند رفتار مدل استاد را تقلید کنند.

۵. Hard Labelsبرچسب‌های حقیقی به صورت بردارهای one-hot که نشان دهنده کلاس واقعی است.

۶. Hard Predictionsاحتمالات خروجی مدل دانشجو وقتی که softmax معمولی (T = 1) استفاده می‌شود Hard predictions .بردارهای one-hot هستند و نشان می‌دهند که مدل دانشجو کدام کلاس را پیش‌بینی کرده است.

در طول فرایند آموزش، مدل دانشجو به دست آوردن پیش‌بینی‌های نرم مدل استاد (soft labels) را به جای مستقیم پیروی از برچسب‌های سخت(hard labels) ، یاد می‌گیرد. این کار با کمینه کردن تفاوت بین پیش‌بینی‌های نرم مدل دانشجو و soft labels تولید شده توسط مدل استاد انجام می‌شود. پارامتر دما (T) در تابع softmax نرمی پیش‌بینی‌ها را کنترل می‌کند.

در فرایند آموزش مدل دانشجو، ما می‌خواهیم مدل دانشجو را طوری آموزش دهیم که با پیش‌بینی‌های نرم مدل استاد (برچسب‌های نرم) همخوانی داشته باشد. این به منظور کمینه کردن تفاوت بین پیش‌بینی‌های نرم مدل دانشجو و برچسب‌های نرم تولید شده توسط مدل استاد انجام می‌شود. پارامتر دما (T) در تابع softmax کنترل کننده نرمی پیش‌بینی‌ها است.

تکثیر دانش امکان می‌دهد مدل دانشجو از اطلاعات اضافی موجود در برچسب‌های نرم بهره‌برداری کند و از دانش مدل استاد که با برچسب‌های سخت تنها قابل دسترسی نیست بهره‌برداری کند. با استفاده از نرمی و دانش مدل استاد، مدل دانشجو می‌تواند عملکرد و کلیت بهتری را با تعداد کمتری پارامتر داشته باشد.

پ)

وزن‌های شبکه دانشجو student بر اساس distillation loss به‌روزرسانی می‌شوند. در الگوریتم تقطیر دانش، هدف اصلی آموزش شبکه دانشجو، تقلید از رفتار و دانش شبکه استاد (teacher) است. بنابراین، تابع ضرر تکثیر برای محاسبه خطا و به‌روزرسانی وزن‌های شبکه دانشجو استفاده می‌شود.

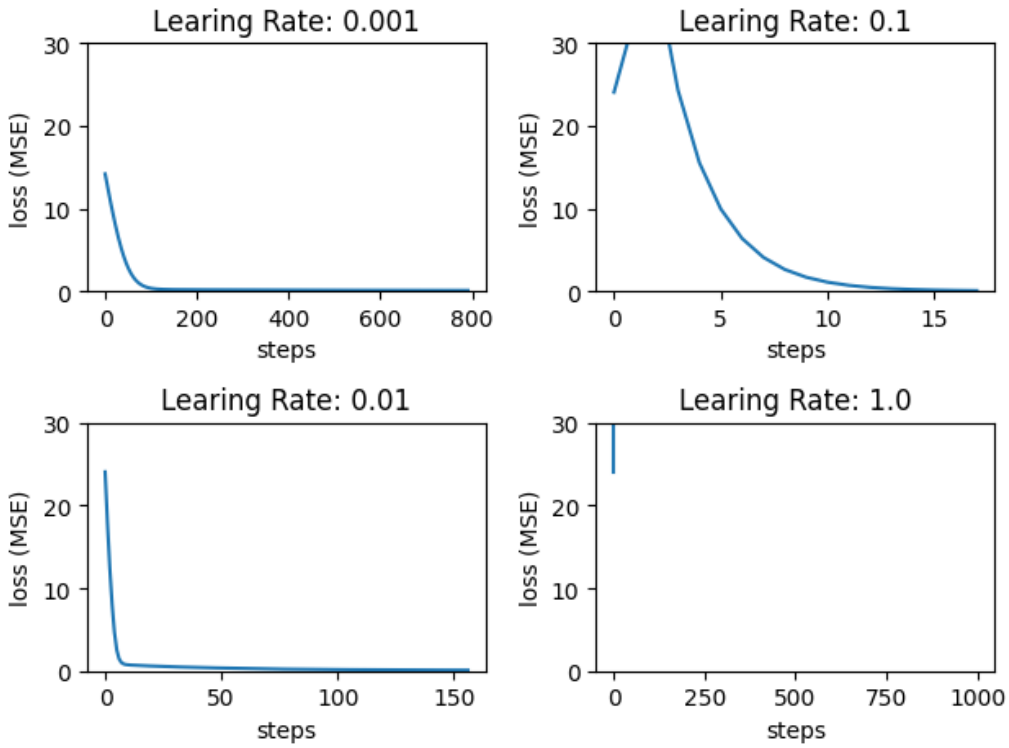
تابع ضرر distillation loss معمولاً بر اساس اختلاف بین پیش‌بینی‌های شبکه دانشجو و برچسب‌های شبکه استاد محاسبه می‌شود. این تابع ضرر، جهت هدایت شبکه دانشجو به سمت پیش‌بینی‌هایی که به شبکه استاد نزدیک‌تر هستند، استفاده می‌شود. با کمینه کردن تابع ضرر distillation loss، وزن‌های شبکه دانشجو به‌روزرسانی می‌شوند تا بهترین تطابق با رفتار شبکه استاد را داشته باشند.

به طور معمول، تابع ضرر دانشجو (student loss) نیز می‌تواند در آموزش شبکه دانشجو مورد استفاده قرار گیرد. این تابع ضرر، معمولاً بر اساس اختلاف بین پیش‌بینی‌های شبکه دانشجو و برچسب‌های واقعی (به صورت یکنواخت یا سخت) محاسبه می‌شود. اما در الگوریتم تقطیردانش، تمرکز اصلی بر روی تابع ضرر تکثیر است و تابع ضرر دانشجو معمولاً جهت اصلاح نقاط ضعف شبکه دانشجو به‌کار می‌رود.

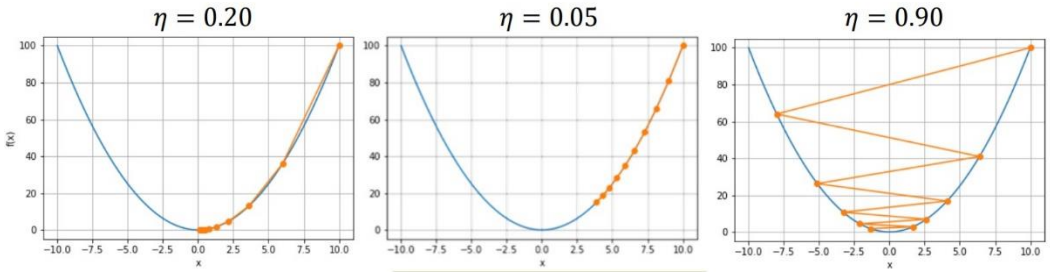
بنابراین، به‌طور خلاصه، وزن‌های شبکه دانشجو بر اساس تابع ضرر distillation loss به‌روزرسانی می‌شوند تا شبکه دانشجو بتواند دانش و رفتار شبکه استاد را تقلید کند.

سوال چهار

برای بهینه ساز SGD نتایج به صورت زیر بود:

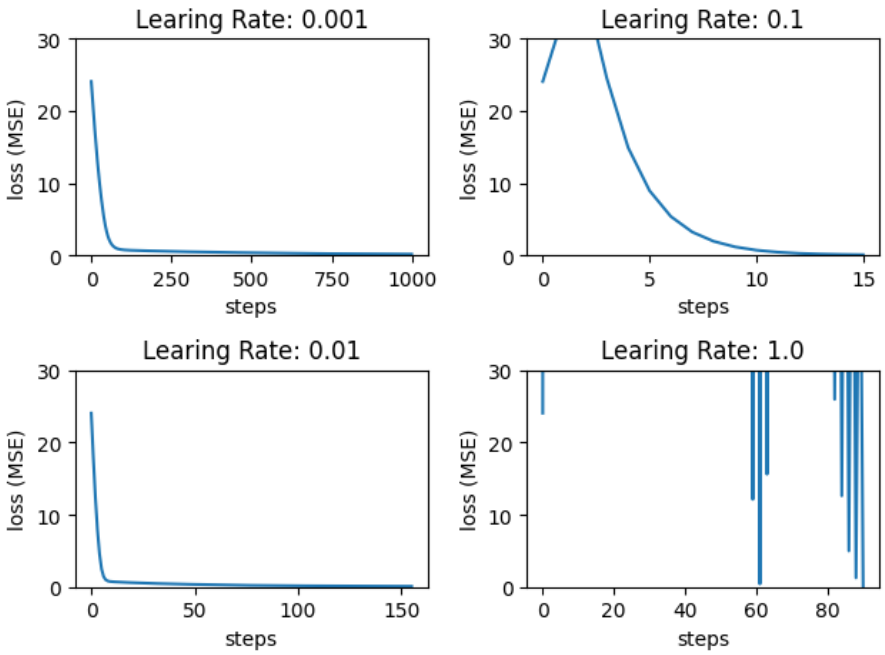


اگر دو حالن lr 0.001 و 0.01 در نظر بگیریم در ان صورت در 0.01 با شیب تندتری loss کاهش پیدا می کند چون وزن ها با سرعت بیشتری به نقطه مینیم نزدیک تر میشوند اما در حالت 0.1 به احتمال زیاد در ابتدا در ابتدا از نقطه مینیم عبور کردیم و در ادامه دوباره به سمت نقطه مینیم حرکت کردیم برای این سه lr دقیقا اتفاقات شبیه شکل زیر افتاده است:



ولی در lr که برابر با ۱ هست احتمالا چون با سرعت زیاد می پریم بعد از یک مدت در global local minimum گیر می کنیم و نمی توانیم از ان در بیایم .

برای حالت momentum نمودار به صورت زیر است:



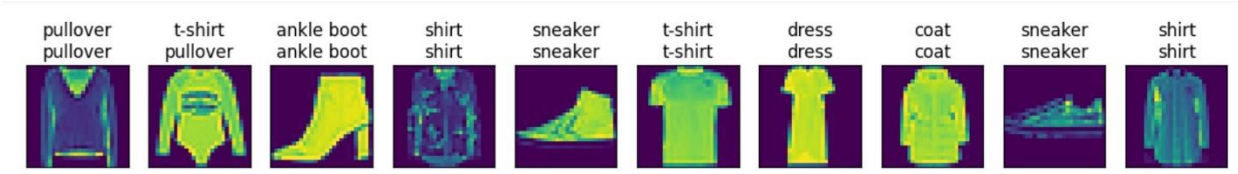
در momentum ما اطلاعات گذشته را استفاده می کنیم و مهم هست استفاده از مومنتوم می‌تواند باعث تسریع همگرایی به نقاط بهینه شود. با اعمال مومنتوم، الگوریتم بهینه‌سازی می‌تواند از سرعت قبلی حرکت کند و در جهت تغییرات گرادیان قبلی و جدید حرکت کند. این سبب می‌شود که بهینه‌سازی سریعتر به نقاط بهینه نزدیک شود.برای ارنینگ ریت 0.001,0.01,0.1 استدلال شبیه به SGD هست فقط با سرعت همگرایی بیشتر ولی برای لرنینگ ریت ۱ ما که SGD در local minium گیر کرده بودیم اینجا باعث میشود یعنی وقتی ار مومنتوم استفاده می کنیم بتوانیم ازlocal minimum بیایم بیرون.

سوال پنج

الف)در قسمت اول از یک لایه hidden به تعدادذ ۲۵۶ نورون استفاده کردم loss ان در ۱۰ تا اپک به صورت زیر شد:

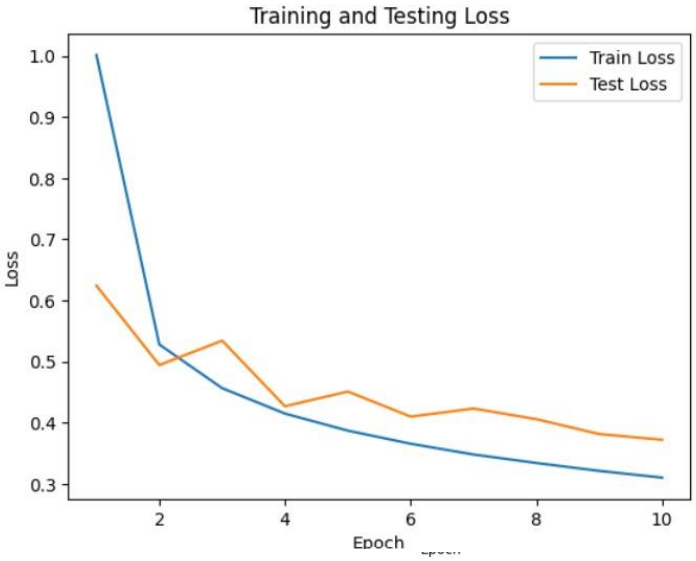
```
Training loss: 0.6906883086858273
Training loss: 0.47706854559465256
Training loss: 0.4346252785785112
Training loss: 0.4091472718824964
Training loss: 0.38897509018241216
Training loss: 0.3738068049428051
Training loss: 0.3608269055070145
Training loss: 0.3502158037921005
Training loss: 0.34052412738519183
Training loss: 0.3310425146016231
```

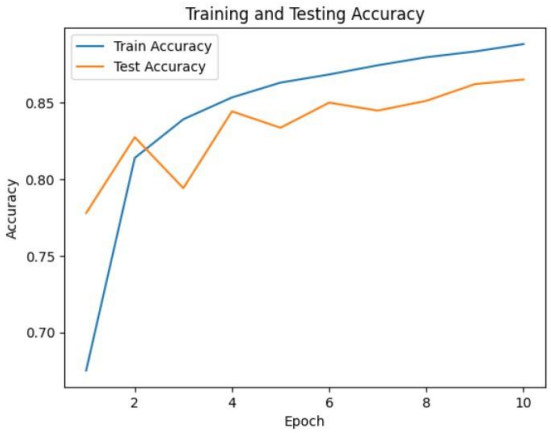
و برای چندعکس داده تست فقط یک خطا داشت:



سوال شش

ب)برای اینکه مدل overfit بشود تعداد پارامترها را افزایش دادم در واقع تعداد لایه های linear رابیشتر کردم که مدل داده های train را حفظ کند دو نمودار accuracy,loss به صورت زیر شد:

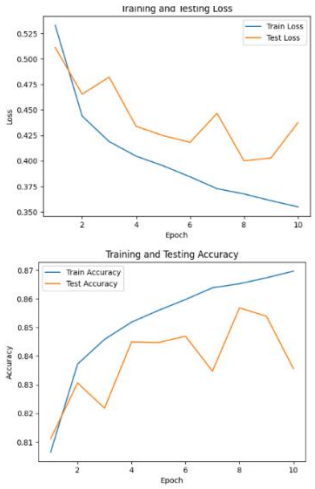




مشاهده می کنیم که مدل overfit شده است.

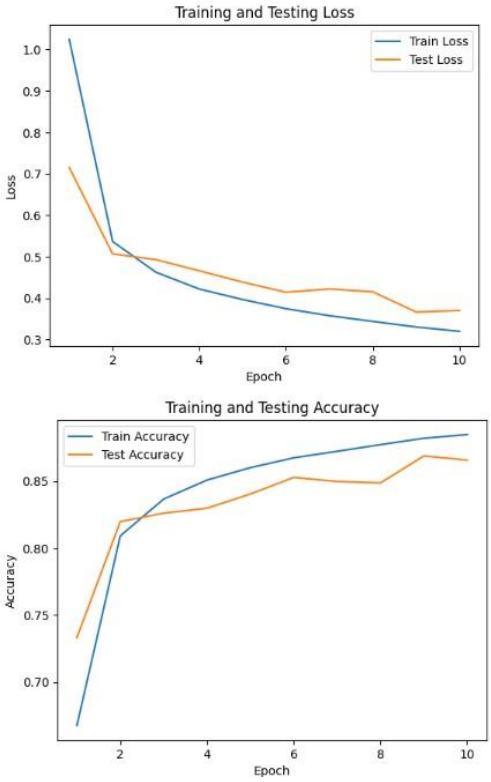
ج)نتایج کمی بهتر شد ولی هنوز overfitting داریم باد تعداد اپک بیشتری نسبت به ۱۰ بذاریم تا بتونیم نتیجه بهتری بگیریم من دوتا اگمنت flip و rotate را اضافه کردم

```
Epoch [1/10], Train Loss: 0.5330, Train Accuracy: 80.64%, Test Loss: 0.5113, Test Accuracy: 81.11%
Epoch [2/10], Train Loss: 0.4439, Train Accuracy: 83.72%, Test Loss: 0.4654, Test Accuracy: 83.06%
Epoch [3/10], Train Loss: 0.4189, Train Accuracy: 84.58%, Test Loss: 0.4822, Test Accuracy: 82.18%
Epoch [4/10], Train Loss: 0.4046, Train Accuracy: 85.18%, Test Loss: 0.4339, Test Accuracy: 84.49%
Epoch [5/10], Train Loss: 0.3952, Train Accuracy: 85.59%, Test Loss: 0.4247, Test Accuracy: 84.47%
Epoch [6/10], Train Loss: 0.3843, Train Accuracy: 85.97%, Test Loss: 0.4182, Test Accuracy: 84.69%
Epoch [7/10], Train Loss: 0.3727, Train Accuracy: 86.38%, Test Loss: 0.4466, Test Accuracy: 83.47%
Epoch [8/10], Train Loss: 0.3675, Train Accuracy: 86.52%, Test Loss: 0.4000, Test Accuracy: 85.68%
Epoch [9/10], Train Loss: 0.3609, Train Accuracy: 86.73%, Test Loss: 0.4028, Test Accuracy: 85.39%
Epoch [10/10], Train Loss: 0.3548, Train Accuracy: 86.96%, Test Loss: 0.4376, Test Accuracy: 83.56%
```



ت)من از منظم سازی L2 استفاده کردم و نتایج نسبت به حالت داده افزایی بهتر بود این منظم سازی از بزرگ شدن بیش از حد وزن ها جلوگیری میکند:

```
Epoch [1/10], Train Loss: 1.0242, Train Accuracy: 66.75%, Test Loss: 0.7150, Test Accuracy: 73.32%
Epoch [2/10], Train Loss: 0.5364, Train Accuracy: 80.91%, Test Loss: 0.5066, Test Accuracy: 81.97%
Epoch [3/10], Train Loss: 0.4622, Train Accuracy: 83.65%, Test Loss: 0.4927, Test Accuracy: 82.60%
Epoch [4/10], Train Loss: 0.4217, Train Accuracy: 85.07%, Test Loss: 0.4658, Test Accuracy: 82.97%
Epoch [5/10], Train Loss: 0.3963, Train Accuracy: 86.00%, Test Loss: 0.4384, Test Accuracy: 84.03%
Epoch [6/10], Train Loss: 0.3742, Train Accuracy: 86.73%, Test Loss: 0.4139, Test Accuracy: 85.27%
Epoch [7/10], Train Loss: 0.3573, Train Accuracy: 87.22%, Test Loss: 0.4219, Test Accuracy: 84.96%
Epoch [8/10], Train Loss: 0.3436, Train Accuracy: 87.71%, Test Loss: 0.4151, Test Accuracy: 84.86%
Epoch [9/10], Train Loss: 0.3301, Train Accuracy: 88.19%, Test Loss: 0.3663, Test Accuracy: 86.87%
Epoch [10/10], Train Loss: 0.3196, Train Accuracy: 88.47%, Test Loss: 0.3698, Test Accuracy: 86.56%
```



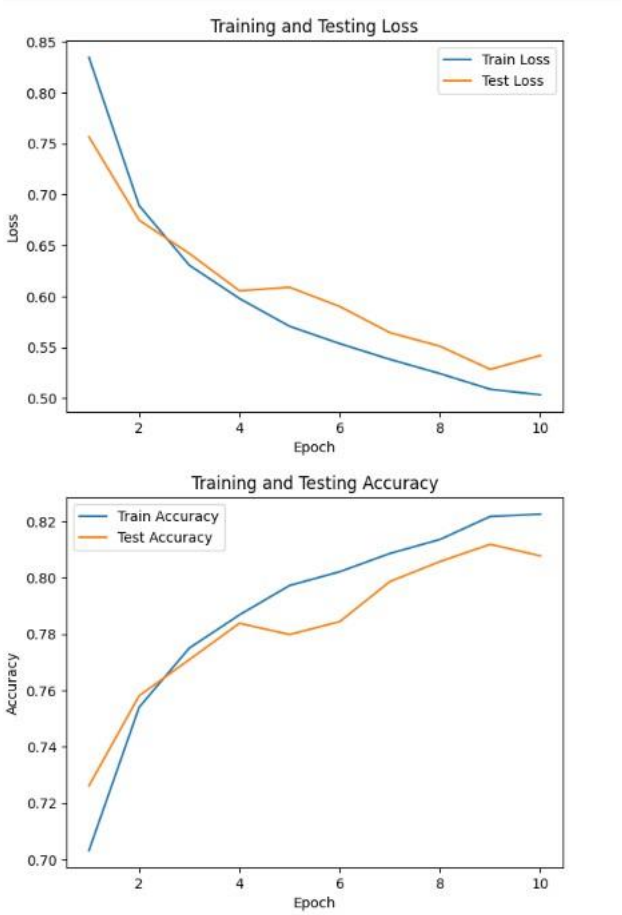
ث)

۱. داده‌افزایی (Data Augmentation) با استفاده از داده‌افزایی، می‌توانید تنوع بیشتری در داده‌های آموزشی ایجاد کنید. این به شبکه عصبی کمک می‌کند تا الگوهای عمومی‌تر را یاد بگیرد و از برازش زیاد داده‌های آموزشی به الگوهای خاص جلوگیری کند. بسته به نوع داده‌ها، شما می‌توانید از تکنیک‌هایی مانند تغییر اندازه، برش، چرخش، اعمال نویز و... استفاده کنید.

۲. منظم‌سازی (Regularization) منظم‌سازی به مدل کمک می‌کند تا از برازش زیاد به داده‌های آموزشی جلوگیری کند و باعث کاهش اورفیت (Overfitting) شود. یکی از روش‌های منظم‌سازی معروف، رگولاریزاسیون L1 و L2 است که در مقابل افزایش مقدار تابع هزینه، وزن‌ها را کاهش می‌دهد.

۳. Dropout :به شبکه عصبی کمک می‌کند تا اطلاعات را به طور تصادفی در هر مرحله آموزش حذف کند. این باعث می‌شود که هر یک از نورون‌ها به طور مجزا آموزش ببینند و وابستگی زیادی به نورون‌های دیگر پیدا نکنند. این به شبکه عصبی اجازه می‌دهد الگوهای عمومی‌تر را یاد بگیرد و از برازش زیاد به داده‌های آموزشی جلوگیری کند.

من ترکیب هر سه را استفاده کردم ونتایج به صورت زیر شد:




```
Epoch [1/10], Train Loss: 0.8344, Train Accuracy: 70.32%, Test Loss: 0.7565, Test Accuracy: 72.62%
Epoch [2/10], Train Loss: 0.6891, Train Accuracy: 75.40%, Test Loss: 0.6748, Test Accuracy: 75.81%
Epoch [3/10], Train Loss: 0.6306, Train Accuracy: 77.50%, Test Loss: 0.6421, Test Accuracy: 77.09%
Epoch [4/10], Train Loss: 0.5979, Train Accuracy: 78.68%, Test Loss: 0.6055, Test Accuracy: 78.38%
Epoch [5/10], Train Loss: 0.5707, Train Accuracy: 79.72%, Test Loss: 0.6089, Test Accuracy: 77.98%
Epoch [6/10], Train Loss: 0.5537, Train Accuracy: 80.21%, Test Loss: 0.5901, Test Accuracy: 78.44%
Epoch [7/10], Train Loss: 0.5382, Train Accuracy: 80.86%, Test Loss: 0.5645, Test Accuracy: 79.86%
Epoch [8/10], Train Loss: 0.5243, Train Accuracy: 81.35%, Test Loss: 0.5511, Test Accuracy: 80.57%
Epoch [9/10], Train Loss: 0.5089, Train Accuracy: 82.17%, Test Loss: 0.5284, Test Accuracy: 81.18%
Epoch [10/10], Train Loss: 0.5035, Train Accuracy: 82.25%, Test Loss: 0.5420, Test Accuracy: 80.77%
```