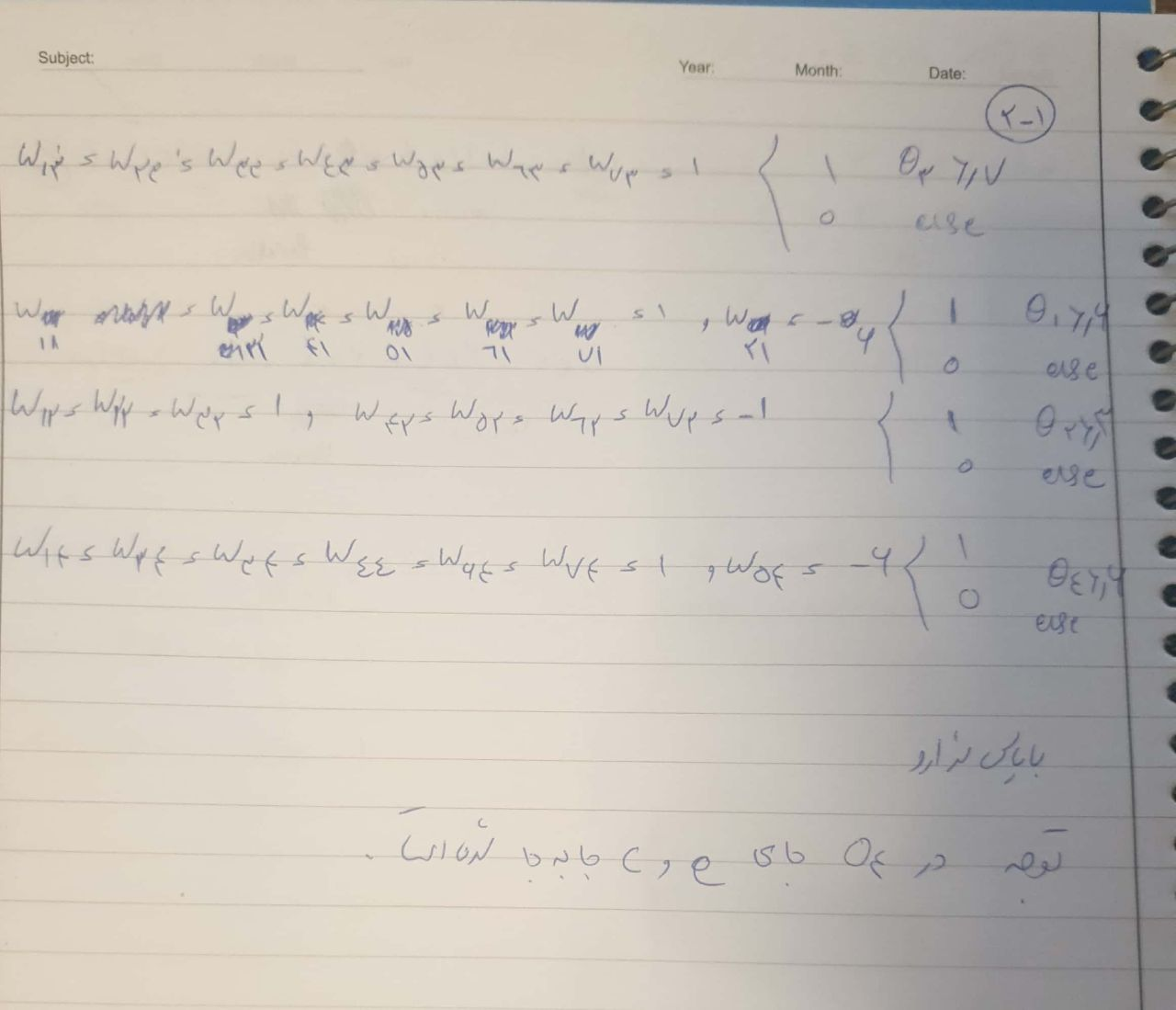
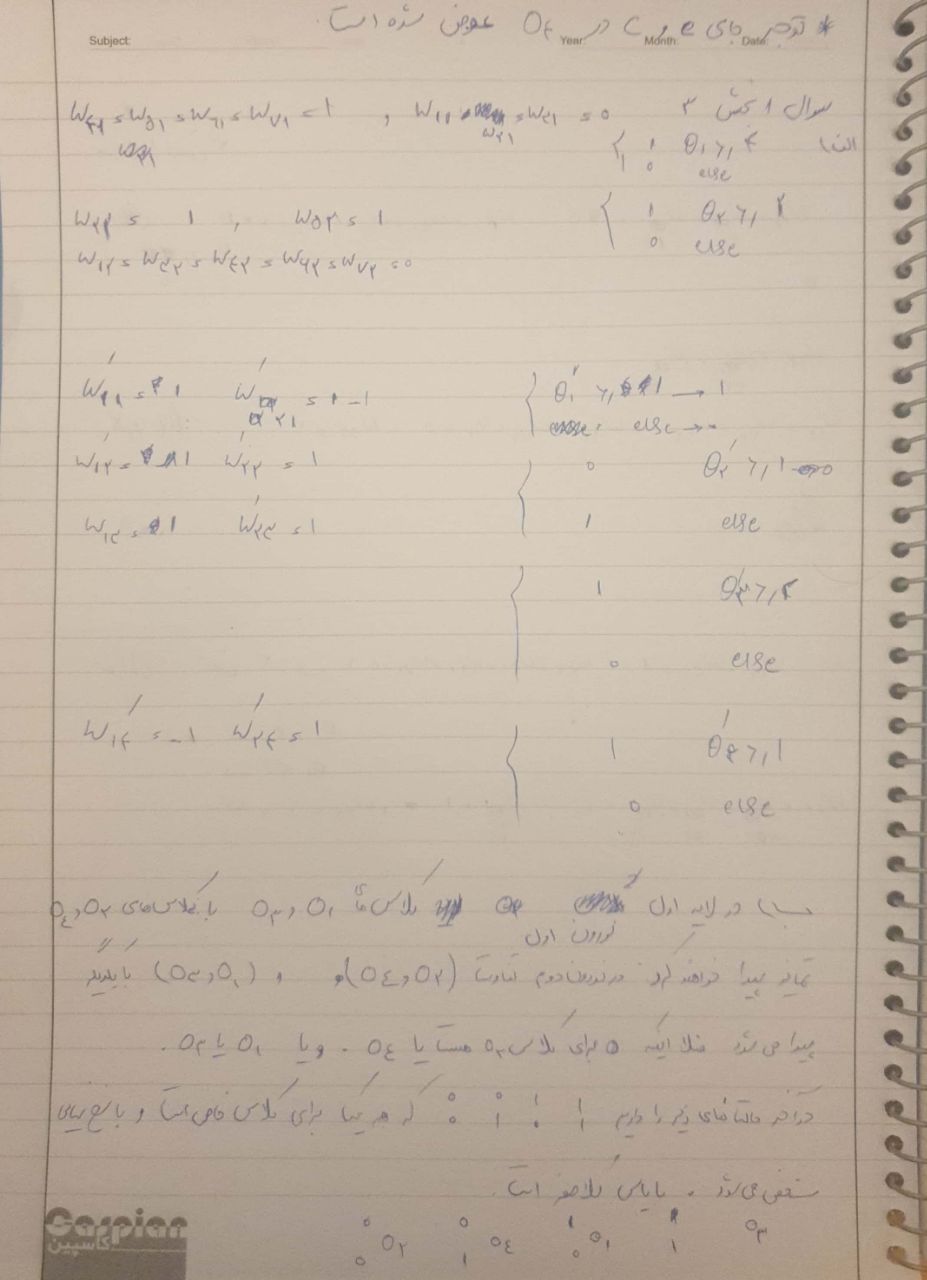
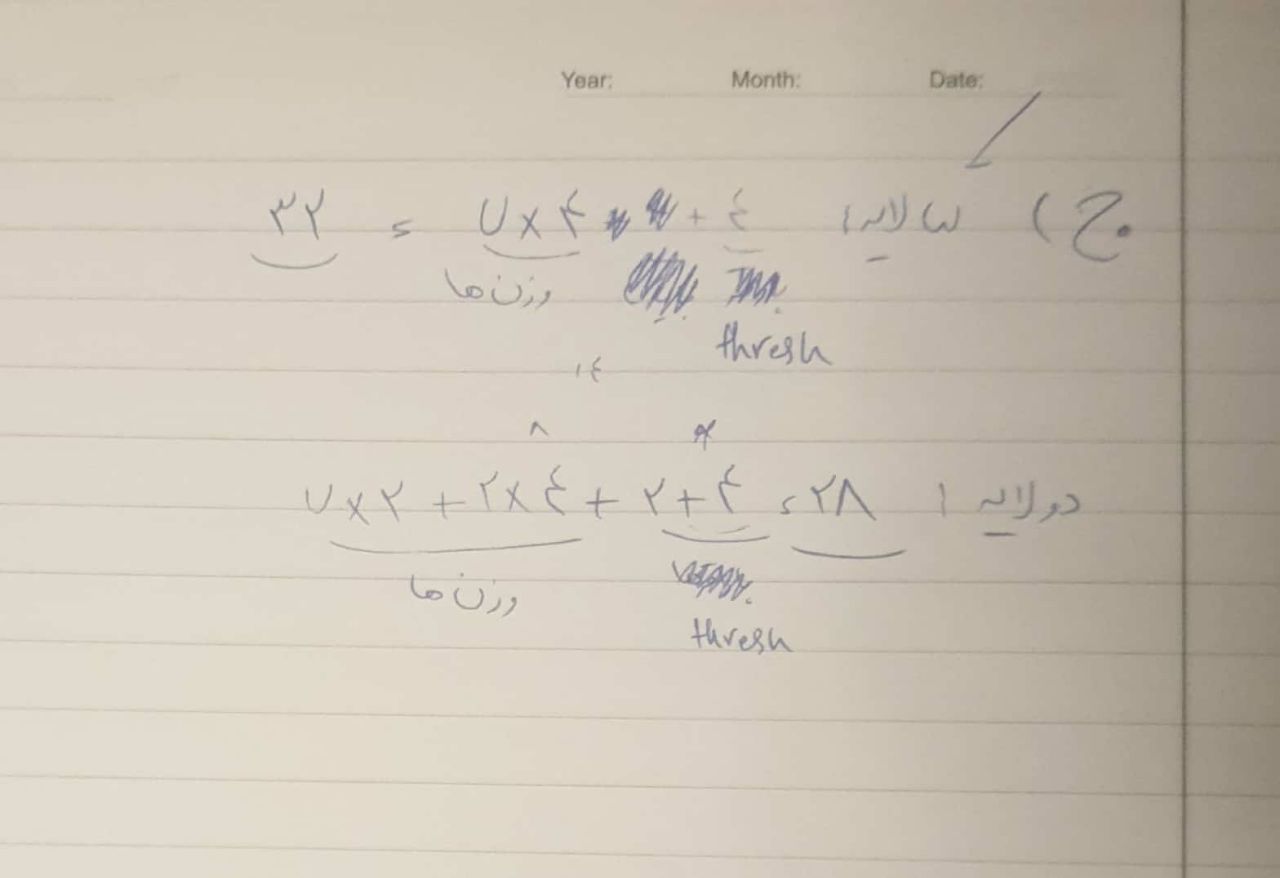
سوال 1 بخش 2

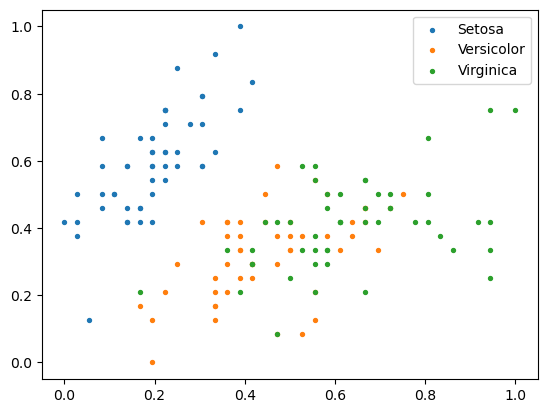


سوال 1 بخش 3 الف ب ج  




سوال 2 بخش 1 الف

ابتدا دیتاست را دانلود میکنیم. یک سری تحلیل‌ آماری انجام میدهیم. مانند تعداد کلاس های تارگت، پراکندگی داده مانند شکل 1-2 و تایپ و تعداد nan برای هر فیچر.



شکل 2-1

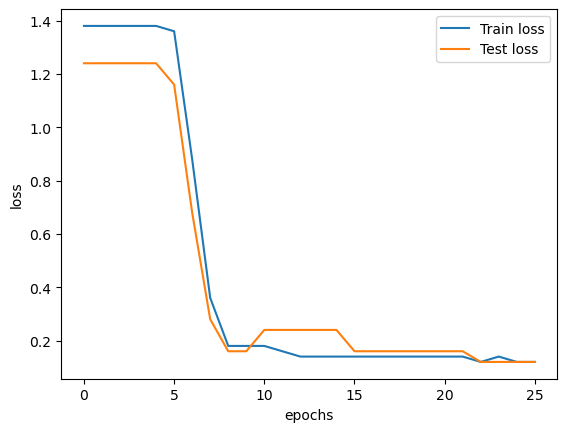
حال دیتاست را بر اساس کلاس SETOSA تقسیم بندی میکنیم به کلاس های Non-SETOSA و SETOSA

SETOSA = 1 و Non-SETOSA = -1. سپس دیتا را به نسبت 1 / 3 تقسیم میکنیم تا 2/3 آن برای آموزش باشد. مدل Adaline را مانند توضیحات کتاب پیاده سازی میکنیم. و با مشخصات زیر مدل را آموزش میدهیم.

Larning rate = 0.001, Thresh = 0.03, Epochs = 25

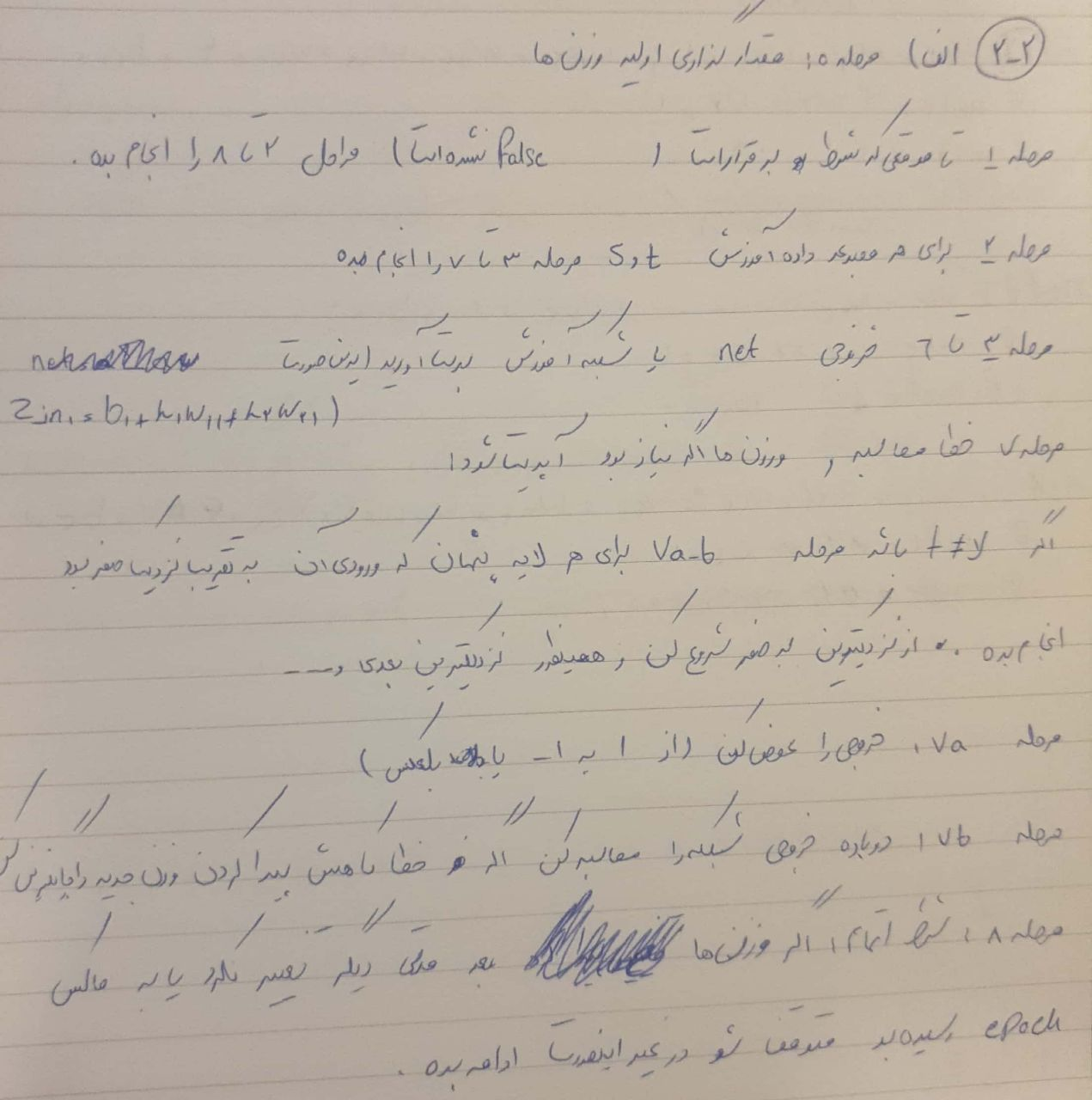
رابطه خطا نیز بدین صورت است:

سپس نمودار خطا را برای تست و آموزش میکشیم. شکل 2-2



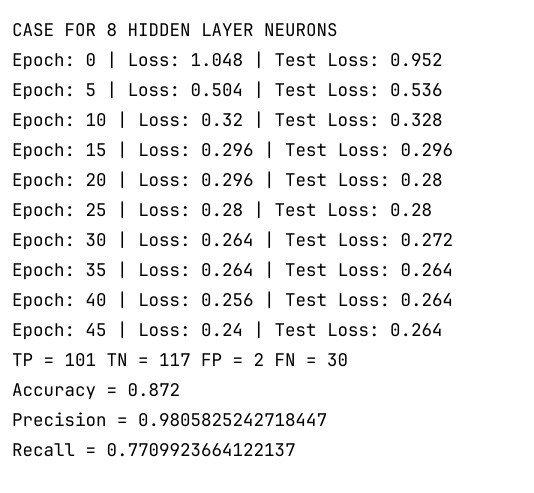
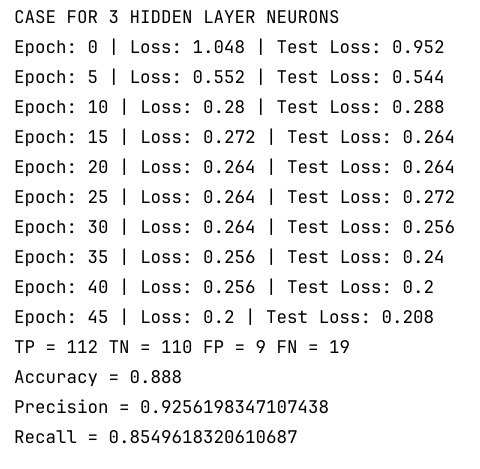
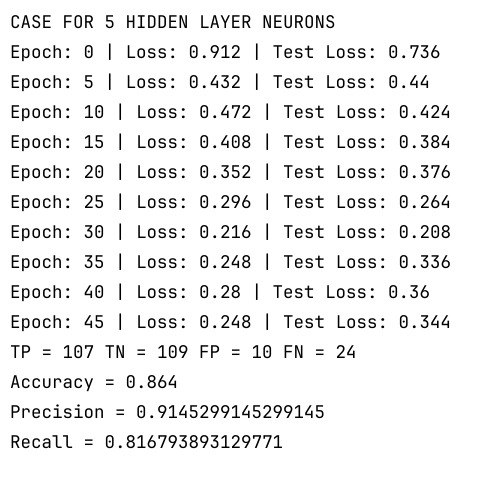
شکل 2-2

همانگونه که پیداست مدل بعد از 25 ایپاک به محل بهینه برای خطای 0.2 رسیده است.

سوال 2 بخش 2 الف

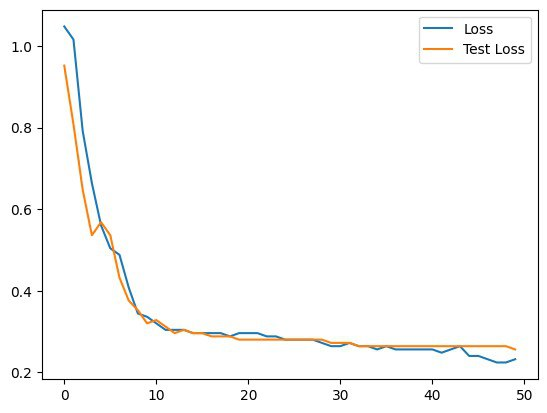
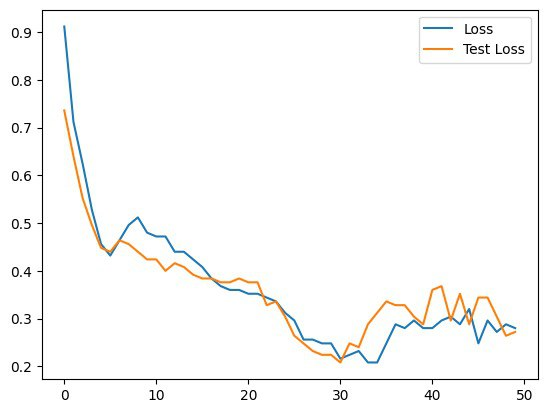
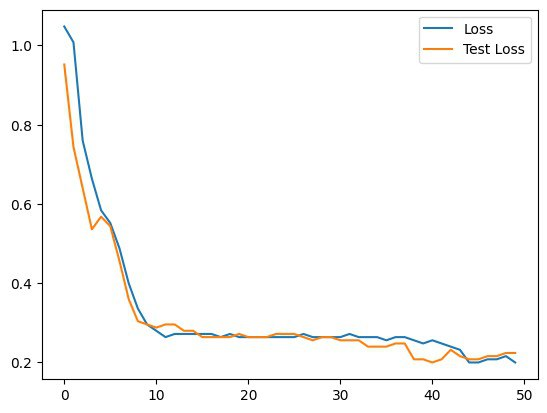
سوال 2 بخش 2 ج

شکل های زیر نمودار و خطا حاصل از نورون های 3، 5 و 8 میباشد.



الف) پنج نورون ب) سه نورون ج) هشت نورون

همانطور که از نتیج مشخص است به ازای اضافه شدن نورون‌ها ما شاهد افزایش خطا و همینطور کاهش دقت شده ایم. این بدین معنی است برای حل این مسئله به خصوص با مدل های خطی Madeline نیاز به تعداد نورون بالا نداریم و جواب شاید در تعداد نورون 3 تا 4 بهینه باشد. حال به سراغ نمودار خطا میرویم.

الف) هشت نورون ب) پنج نورون ج) سه نورون

تمام نکات از قبل گفته شده است. نکته جالب وجود نوسانات و ناپایداری در تعداد نورون 5 تایی است! از طرفی در برخی نواحی نیز خطای تست از ترین پایین تر است که میتواند به علت مقدار کم تعداد دیتا برای بخش تست باشد.

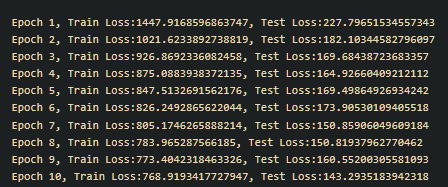
سوال 3 بخش 2

ابتدا مدل را پیاده‌سازی میکنیم. مدل دارای 2 بخش است decoder و encoder که هر کدام فانکشن خود را دارا هستند.

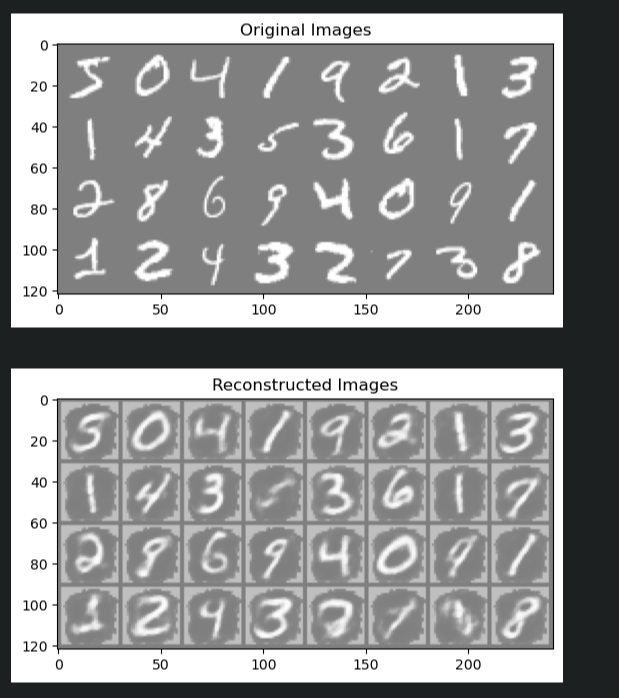
سپس دیتا رو دانلود کرده و به سراغ محاسبه w میرویم. برای اینکار ابتدا دیتا را بر اساس لیبل جدا میکنیم. سپس در هر لیبل دو جفت عدد انتخاب کرده و از طریق رابطه داخل مقاله اختلاف را حساب میکنیم. این کار را برای همه دو جفت انجام میدهیم. سپس میانگین میگیریم و بعد از پیدا کردن میانگین هر لیبل میانگین کلی یک پیکسل را حساب میکنیم. به این عدد inter میگوییم. حال همین کار را برای دیتا های بین دو لیبل ( کلاستر ) انجام میدهیم. پس از محاسبه با رابطه داخل مانند قبل برای هر لیبل میانگین را حساب کرده و میانگین کلی را حساب میکنیم. به این عدد intra میگوییم.

حال که w به دست آمده است. این وزن را در mse ضرب میکنیم. و با نورم وزن ها ضرب در b که در مقاله 0.00001 تعریف شده است جمع میکنیم. توجه کنید که قبل از تمامی این مراحل دیتا نرمالایز شده است. خطا بدین صورت میباشد. شکل 3-1

همچین عکس ها تولید شده به ازای ورودی به این صورت میباشد. شکل 3-2



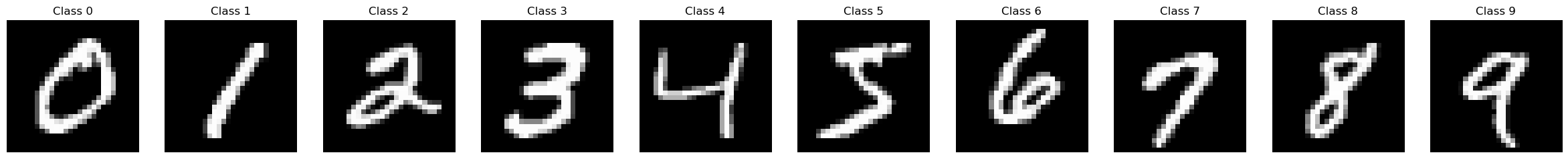
شکل 3-1



شکل 3-2

سوال 4 بخش 1 ب

ابتدا به مانند سوال قبل با پایتورچ دیتا را دانلود میکنیم و تبدیل به تنسور میکنیم. حال یک آرایه ایجاد میکنیم به طوری که تصویر هر کلاس در ایندکسش جا بگیرد. مثلا تصویر 0 در خانه 0 قرار بگیرد و.. سپس بااستفاده از subplot و پیمایش بر روی آرایه شکل زیر پدید می‌آید. شکل 4-1



شکل 4-1

سوال 4 بخش 1 د

به سادگی مانند قبل موقعی که در حال لود دوباره دیتا هستیم میتوان پارامتر transform را برابر با مقدار زیر قرار داد:  
transforms.Normalize(min\_value, max\_value)

که ماکس 1 و مین 0 هست. حال دیتایی که داریم بین 0 و1 است.

البته میتوان همان دیتا لود شده را بعد از لود شدن ترنسفورم انجام داد! ولی خب برای این سوال به این شکل اعمال شده است.

سوال 4 بخش 3

ابتدا seed میگذاریم تا جواب های بکسان بدست بیاوریم.

سپس دیتا رو دانلود میکنیم، تبدیل به تنسور کرده و split میکنیم. دو دیتالودر با بچ 32 برای قسمت تست و ترین ایجاد میکنیم. مدل را همانطور که خواسته شده بود ایجاد میکنیم. و سپس اپتیمایز SGD با لرنینگ ریت 0.01 و loss فانکشن CrossEntropyLoss را تعریف میکنیم.

حال مدل را با epoch = 10 ترین میکنیم. در هر مرحله یک بچ دریافت میکنیم به مدل میدهیم، loss را حساب میکنیم و با اپتیمایزر مدل را آپدیت میکنیم. سپس بعد از اتمام epoch خطا را نشان میدهیم و همان طور که از کد پیداست بعد از مرحله اول مدل تغییری نمیکند.

حال برای تعیین misclassifications این بار از دیتالودر تست استفاده میکنیم. با دستور with torch.no\_grad به مدل می‌فهمانیم که در مرحله feedforward هستمی و نیازی به گرادیان نیست

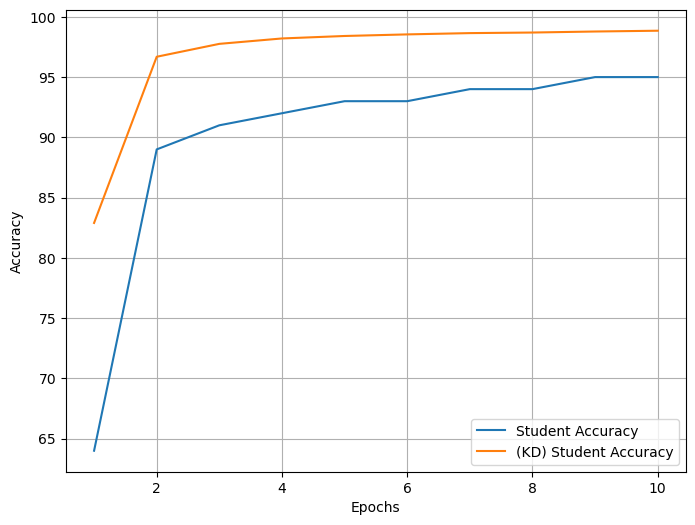
مانند مرحله ترین دیتا را به مدل میدهیم و با argmax محتمل ترین کلاس را به عنوان کلاس برتر انتخاب کرده و حال با مقدار واقعی مقایسه میکنیم. تعداد کلاس های غلط جمع میشیود. در آخر با مقدار 10684 از 12000 محسابه میشود.

سوال 4 بخش 4 ب

همانطور که از شکل زیر 4 – 2 معلوم است. به طور کل مدل در آخر عملکرد بهتری از خود نشان داده است.

مدل KDدر ایپاک اول نیز با اختلاف حدود 20 درصدی برتری داشته است؛ که نشان میدهد قابلیت یادگیری با ایپاک کم نیز وجود دارد. از طرفی مدل KD در ایپاک کمتری به نقطه بهینه رسیده است درحالی که مدل ساده یک سری نوسانات داشته است.

از همه مهم تر مدل KD با اینکه تعداد پارامتر های کمی داشته توانسته است از مدل teacher که تعداد پارامتر های بیشتری داشته است جلو بزند. که این خود یکی از ویژگی های یادگیری با روش KD است.



شکل 4-2