به نام خدا

دانشگاه صنعتی امیرکبیر

(پلی‌تکنیک تهران)

درس یادگیری ماشین

استاد ناظرفرد

تمرین اول

علیرضا مازوچی

۴۰۰۱۳۱۰۷۵

بخش اول: پرسش‌های تشریحی

سوال ۱

الف) نادرست؛ چنین گزاره‌ای همواره برقرار نیست. به عنوان مثال اگر مدل دچار بیش‌برازش شده باشد و باز پیچیدگی مدل را زیاد کنیم، خطای مدل در هنگام آموزش کم‌تر هم می‌شود ولی در موقع تست ممکن است خطا بیشتر شود. برای درک بهتر می‌توانید به تصویر سوال ۲ مراجعه کنید.

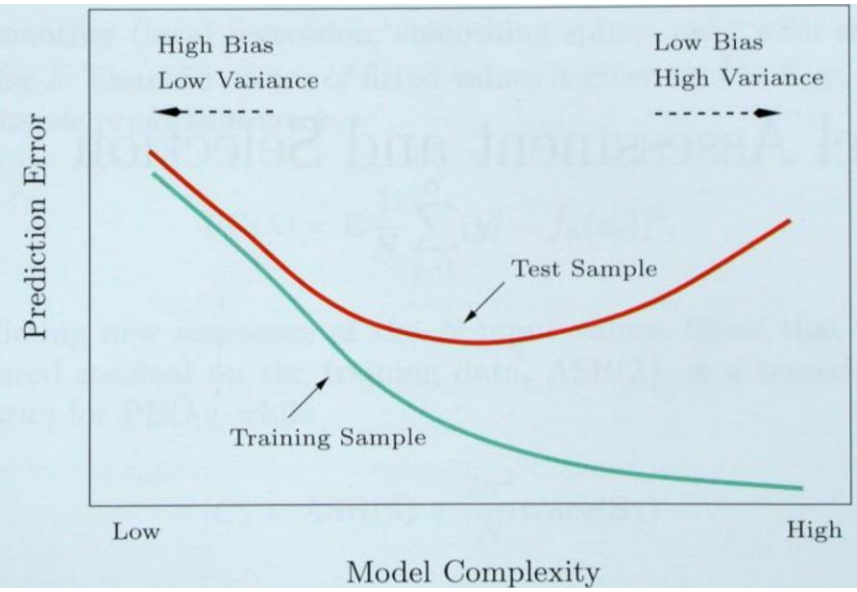
ب) درست اگر پیچیدگی مدل را ثابت فرض کنیم؛ پیچیدگی مدل و پارامترهای آن باید متناسب با تعداد داده و شرایط مسئله باشند. طبیعی است که در این شرایط و با فرض ثابت ماندن پیچیدگی مدل اگر داده‌های آموزش کمتر شود، استعداد مدل برای بیش‌برازش زیادتر می‌شود. اگر هم بتوان مدل را ساده‌تر کرد که این مشکل جلوگیری است.

ج) نادرست؛ الزاما خیر! به عنوان مثال اگر مدل دچار کم‌برازش باشد، پیچیده‌تر کردن به اندازه مدل باعث کاهش هم خطای آموزش و هم خطای تست خواهد شد. برای درک بهتر می‌توانید به تصویر سوال ۲ مراجعه کنید.

د) درست؛ به طور کلی معیار MAE در برابر نویز مقاوم‌تر است. داده‌های نویز داده‌هایی هستند که از الگوی واقعی داده‌های مسئله فاصله زیادی دارند. در معیار MSE چون از توان دو برای فاصله استفاده می‌کند، مدل را برای داده‌های با فاصله زیاد از جمله داده‌های نویز بیشتر جریمه می‌کند درحالی که در MAE این‌چنین نیست. در معیار RMSE هم اگرچه از یک رادیکال استفاده می‌شود، اما در زیر رادیکال باز داده‌های نویز می‌توانند بخش مهم‌تری از خطا را تشکیل دهند.

سوال ۲

این اتفاق زمانی رخ می‌دهد که مدل مانند تصویر زیر دچار بیش‌برازش شده باشد. به این علت که مدل در زمان آموزش به نوعی داده‌ها را به جای یادگیری، حفظ کرده است و روی آن می‌تواند خطای پایینی داشته باشد. اما در هنگام اعتبارسنجی (یا تست) و با داده‌های جدید مدل می‌تواند خطای بالایی داشته باشد چون الگویی که توسط مدل یادگرفته شده است یک الگوی شدیدا وابسته به نمونه‌های آموزش است.



برای حل این مشکل می‌توان راهکارهای زیر را پیشنهاد داد:

۱. کاهش پیچیدگی مدل: منطقی‌ترین و متداول‌ترین راه این است که پیچیدگی مدل را کاهش دهیم تا بیش‌برازش از میان برود.
۲. افزایش نمونه‌های آموزش: اگر نخواهیم پیچیدگی مدل را کاهش دهیم، می‌توان حجم داده‌های آموزش را افزایش داد تا تناسب این دو برقرار شود.
۳. افزودن جمله منظم‌ساز: می‌توان به تابع خطای مسئله یک جمله منظم‌ساز اضافه کرد تا در هنگام آموزش مدل سعی کند با حداقل پیچیدگی به دقت مناسب برسد.

سوال ۳

الف)

$$J(w) = \sum_{i=1}^{n} \left(y^i - w^T x_i\right)^2 = \left||y - xw|\right|^2 = (xw - y)^T (xw - y)$$
$$= ((xw)^T - y^T)(xw - y) = (xw)^T xw - 2(xw)^T y + y^T y$$
$$\frac{\partial J}{\partial w} = 2x^T xw - 2\, x^T y = 0 \rightarrow w = (x^T x)^{-1} x^T y$$

ب) می‌توان به مشکلات زیر اشاره کرد:

۱. وارون‌پذیر نبودن $x^T x$: اگر $x^T x$ وارون‌پذیر نباشد، از رابطه ارائه‌شده نمی‌توان استفاده کرد. برای آنکه همچنان این رابطه قابل استفاده باشد می‌توان از رابطه Moore-Penrose[1] برای یک تخمین مناسب از وارون $x^T x$ کمک گرفت.

۲. محدودیت حافظه: یکی دیگر از محدودیت‌های رابطه پیشنهادی بحث نگهداری ماتریس‌ها است. چنانچه مسئله داده‌های زیادی داشته باشد ماتریس x و x^T فضای زیادی از رم را اشغال می‌کنند و حاصل ضرب و وارون‌گیری آن‌ها هم سخت خواهد شد. برای غلبه به این مشکل می‌توان ماتریس‌ها را حافظه جانبی نگه داشت و موقع ضرب یک سطر در یک ستون تنها این دو را به حافظه اصلی آورد. سایر محاسبات نظیر وارون‌گیری هم تنها روی بخشی از داده‌ها در هر لحظه اعمال می‌شود. لذا با این روش می‌توان مشکل حافظه‌ی اصلی را حل کرد اما با این حال باید توجه داشت روش پیشنهادی مشکل زمان اجرای الگوریتم (که یکی دیگر از مشکلات آن است!) را تشدید خواهد کرد.

[1] https://en.wikipedia.org/wiki/Moore%E2%80%93Penrose_inverse

ج)

$$J(w) = \left|\left|y - xw\right|\right|^2 + \lambda \left|\left|w\right|\right|^2 = (y - xw)^T(y - xw) + \lambda w^T w$$
$$= ((xw)^T - y^T)(y - xw) + \lambda w^T w$$
$$= (xw)^T xw - 2(xw)^T y + y^T y + \lambda w^T w$$

$$\frac{\partial J(w)}{\partial w} = 2x^T xw - 2x^T y + 2\lambda w = 0 \rightarrow (x^T x + \lambda I)w = x^T y \rightarrow w$$
$$= (x^T x + \lambda I)^{-1}(x^T y)$$

د) هنگامی که از یک جمله منظم‌ساز استفاده می‌کنیم در تابع خطا اندازه پارامترهای مدل هم درنظر گرفته می‌شود یعنی آنکه مدلی که پارامترهای زیاد و با مقادیر زیاد داشته باشد جریمه می‌شود. نتیجه چه می‌شود؟ در این حالت الگوریتم یادگیری مجبور می‌شود تا از تعداد پارامتر کمتر و با مقادیر کوچک استفاده کند و بدین شکل مدل ساده‌تر خواهد شد و دچار بیش‌برازش نمی‌شود. به بیان دیگر، اگر این جمله را استفاده نمی‌کردیم و پارامترهای مدل را زیاد در نظر می‌گرفتیم مدل برای کاهش خطا در مجموعه داده آموزشی ممکن بود الگوهای بسیار پیچیده را پیشنهاد دهد درحالی که در مجموعه تست احتمالا خطا بیش‌تر هم می‌شود ولی با وجود این جمله مدل علاوه بر کاهش خطای آموزش سعی در ساده‌بودن نسبی مدل هم دارد.

سوال ۴[۳۲]

الف) الگوریتم معادله نرمال مناسب نیست زیرا به تعداد دفعه بالا باید برای هر داده ماتریس وارون را محاسبه کرد. دو روش Stochastic GD و Batch GD به دلیل پردازش دسته‌ای مشکل زمان را ندارند. از نظر حافظه هم مشکلی ندارند چراکه کافی است تا تنها بخشی از داده در حافظه باشد. روش Batch GD هم مشکل زمان را ندارند ولی ممکن است از نظر حافظه‌ای مشکل‌ساز شود. اگر حافظه به اندازه کافی داشته باشیم این روش را هم می‌توان پذیرفت درغیر این صورت خیر.

---

[2] https://gist.github.com/byelipk/345ee92c42f579a9dc1938b0bb86be2e

[3] https://www.atoti.io/when-to-perform-a-feature-scaling/

ب) زمانی که از الگوریتم مهادله نرمال استفاده می‌شود مقیاس‌های متفاوت ویژگی‌ها موثر نیست و مشکلی ایجاد نمی‌کند اما برای الگریتم‌های مبتنی بر گرادیان نزولی این مسئله تاثیر منفی می‌گذارد. در backpropagation و برای ویژگی‌ها با مقیاس بزرگ نیاز است که از گام‌های بلند استفاده شود اما برای سایر ویژگی‌ها باید از گام کوچک‌تر استفاده کرد. در نتیجه یا دقت مناسب را به سبب لطمه خوردن به ویژگی‌ها با مقیاس کم آسیب خواهد دید و یا آنکه مجبور به صرف زمان زیاد خواهیم بود تا تمام ویژگی‌ها به مقادیر مناسب خود برسند. برای حل این مشکل می‌توان از روش Feature Scaling استفاده کرد. با روش Feature Scaling تمامی ویژگی‌ها در یک مقیاس تقریبا یکسان قرار خواهند گرفت و سرعت همگرایی دسته الگوریتم‌های مبتنی بر گرادیان نزولی تسریع پیدا می‌کند. برای Feature Scaling روش‌های متعددی هست مثلا می‌توان از روش نرمال‌سازی min-max با فرمول زیر بهره جست. در این رابطه f' نرمال‌شده ویژگی f خواهد بود.

$$f' = \frac{f - \min(f)}{\max(f) - \min(f)}$$

پس از اعمال نرمال‌سازی مذکور مقدار تمام ویژگی‌ها در رنج ۰ تا ۱ قرار خواهد گرفت.

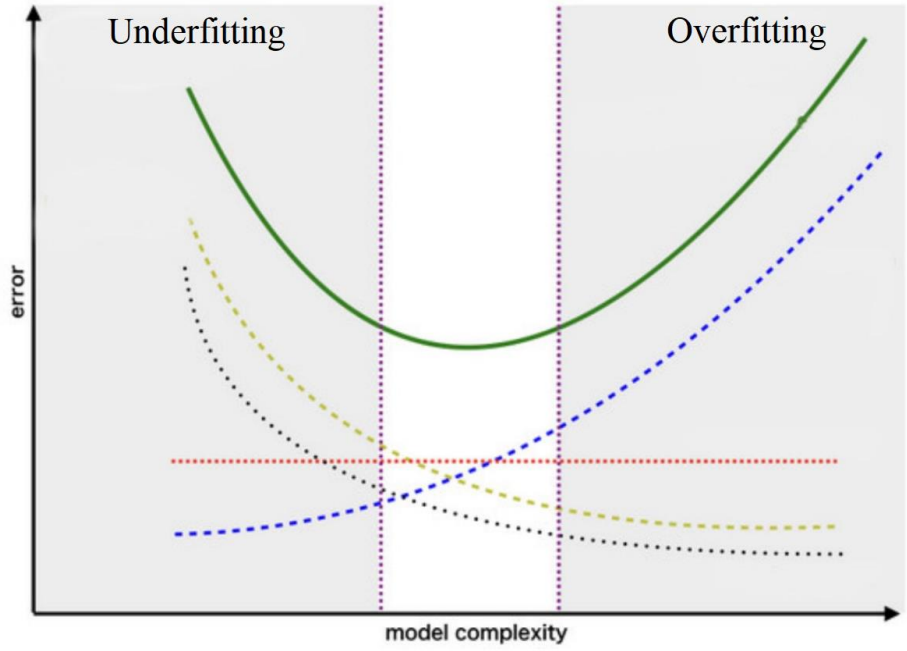
سوال ۵

به طور کلی، با افزودن داده‌های جدید به مدلی که واریانس بالایی دارد، مشکلات ناشی از واریانس بالا کمتر می‌شود. قبل از افزودن داده‌های جدید مدل توجه زیادی به نمونه‌های موجود در مجموعه آموزشی دارد و در مواردی به جای توجه به الگوی‌کلی درگیر نمونه‌ها می‌شود. پس از افزودن داده‌های بیشتر، مدل بیشتر کنترل می‌شود و یک داده نویز به تنهایی نمی‌تواند مانند سابق در نظر مدل اهمیت داشته باشد.

به طور کلی، با افزودن داده‌های جدید به مدلی که بایاس بالایی دارد، تاثیر چندان مثبتی برای مدل رخ نخواهد داد. توجه کنید که پیش از افزودن داده به مدل، مدل اهمیت کافی به داده‌های موجود نمی‌دهد و الگوی داده‌ها را ساده‌تر از چیزی که باید

باشد تصور می‌کند. بدیهی است که در این ظرایط افزودن داده‌های بیشتر در چیزی که مدل یادگرفته اســت تاثیر جدی‌ای نخواهد داشــت و مدل همچنان الگوی ســابق را پیشنهاد می‌دهد. در این شرایط نه تنها مشکل بایاس حل نشده است بلکه بیشتر هم شده است!

سوال ۶[4]



به نکات زیر توجه کنید:

- هرچه مدل ســاده‌تر باشــد احتما کم‌برازش و اهمیت ندادن به داده‌ها بیشــتر می‌شــود و هر چه مدل پیچیده‌تر باشــد احتمالا بیش‌برازش بیشــتر می‌شــود. (محدوده‌های خاکستری)
- هرچه تعداد پارامتر مدل بیشتر شود مدل توانایی بیشتری برای کاهش خطای آموزش دارد. در مورد خطای تســت، تا زمانی که مدل در کم‌برازش اســت هم

[4] https://towardsdatascience.com/the-bias-variance-tradeoff-8818f41e39e9

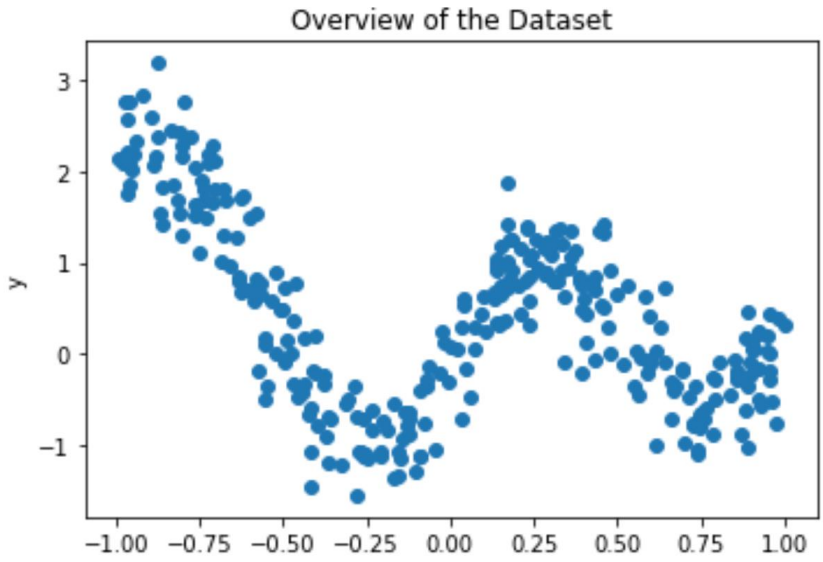
خطای آموزش و هم خطای تست زیاد است اما وقتی مدل از محدوده برازش مناسب خارج شود دیگر خطای تست کاهشی نخواهد ماند. (منحنی سبز و سیاه)

- زمانی که مدل بسیار ساده است با کمی پیچیده‌کردن مدل، آموزش بسیار کاهش پیدا می‌کند اما وقتی مدل به اندازه کافی پیچیده شد، پیچیده‌کردن مدل تاثیر کمتری بر کاهش خطای آموزش مدل دارد. (منحنی زرد و سیاه)
- مدل‌های ساده بایاس بیشتر و واریانس کمتر دارند و بالعکس. این نکته در مورد مربع واریانس هم صادق خواهد بود. (منحنی آبی و زرد)
- آخرین منحنی طبیعتا برای خطای ناشی از نویز باقی می‌ماند. (منحنی قرمز) مدل همواره مقداری خطا خواهد داشت و آن به دلیل وجود این داده‌های نویز است. با این دید شاید بتوان آن را ثابت درنظر گرفت و ادعا کرد این مقدار همواره وجود دارد و اگر تلاشی هم برای حذف آن‌ها گرفته شود باعث به وجود آمدن خطای بیشتری می‌شود. هرچند که می‌توانستیم برای منحنی خطای ناشی از نویز نموداری را درنظر بگیریم که در قسمت بیش‌برازش مقادیر بیشتری را به دلیل گمراه کردن مدل، اتخاذ کند.

بخش دوم: پیاده‌سازی

سوال ۱

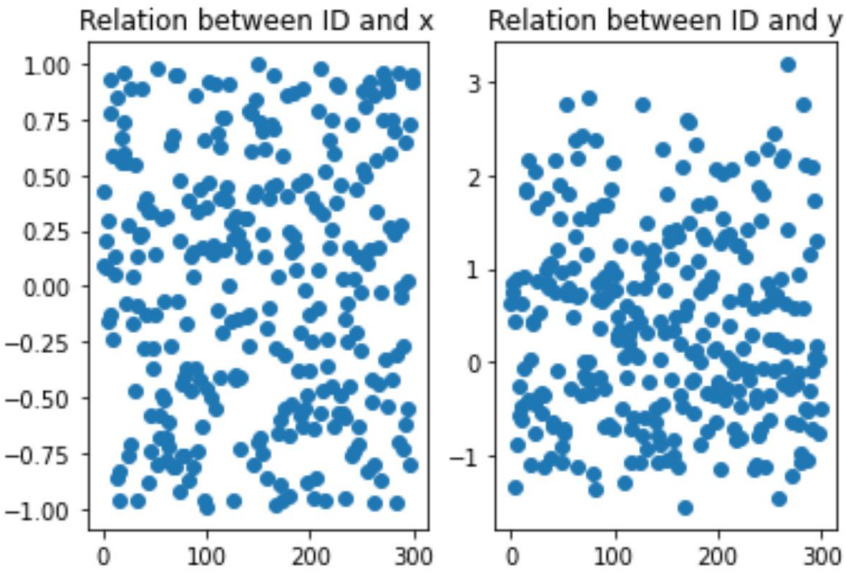
الف)



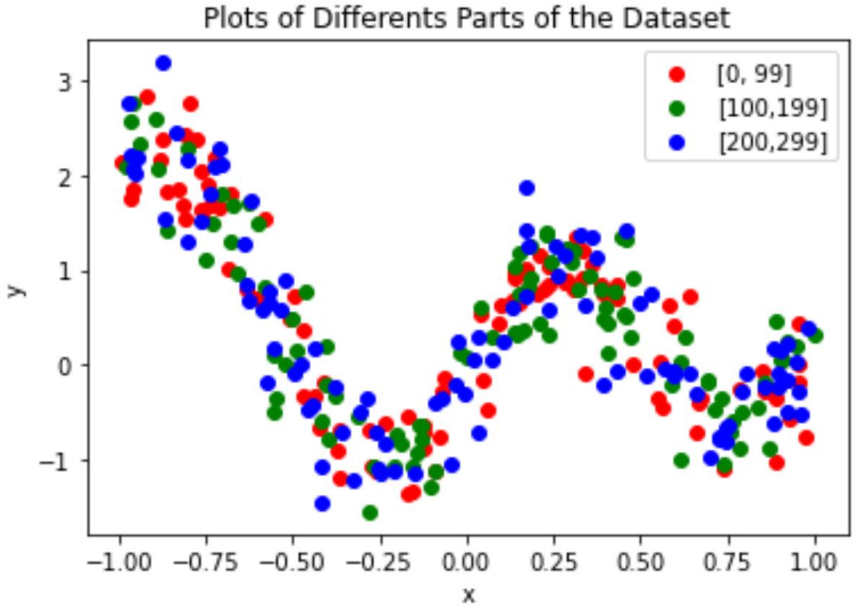
ب) در برخی از مجموعه‌های داده ممکن است ترتیب اولیه داده‌ها بامعنی باشد. مثلا مقدار یکی از ویژگی‌ها به صورت افزایشی مرتب شده باشد یا داده‌های مربوط به یک کلاس در ابتدا بیاید و داده‌های کلاس دیگر در انتها! چنین چیزی می‌تواند مشکلاتی را به وجود بیاورد. مثلا موقع تقسیم داده‌ها به مجموعه آموزشی و تست، چون عمدتا تکه اول داده‌ها برای آموزش و تکه دوم برای تست برداشته می‌شوند، دو مجموعه از همدیگر متفاوت خواهند شد و مدل در زمان تست در شرایط جدیدی قرار خواهد گرفت. همچنین در هنگام آموزش هم اگر قرار باشد داده‌ها به صورت دسته‌ای به مدل داده شود بهتر است مجموعه آموزشی شافل شده باشد تا هر دسته شرایط نسبتا مشابهی با کل داده‌ها داشته باشد.

برای بررسی اینکه آیا نیاز به شافل کردن وجود دارد یا خیر من دو آزمایش را انجام داده‌ام. ابتدا بررسی کرده‌ام که آیا مقدار یک ویژگی با افزایش شماره آیدی دارای الگوی

خاصی است یا نه؟ برای دو ویژگی x و y نمودارهای زیر حاصل شد که نشان می‌دهد ارتباطی میان آیدی و این دو ویژگی وجود ندارد:



نهایتا بررسی کردم که آیا قسمت‌های مختلف مجموعه‌داده الگویی مشابه هم دارند یا خیر؟ برای این کار مجموعه‌داده را به سه قسمت تقسیم کردم و آن‌ها را ترسیم کردم. به نظر می‌رسد الگوی هر سه قسمت مشابه هم دیگر است و مجموعه‌داده نیازی به شافل شدن ندارد.

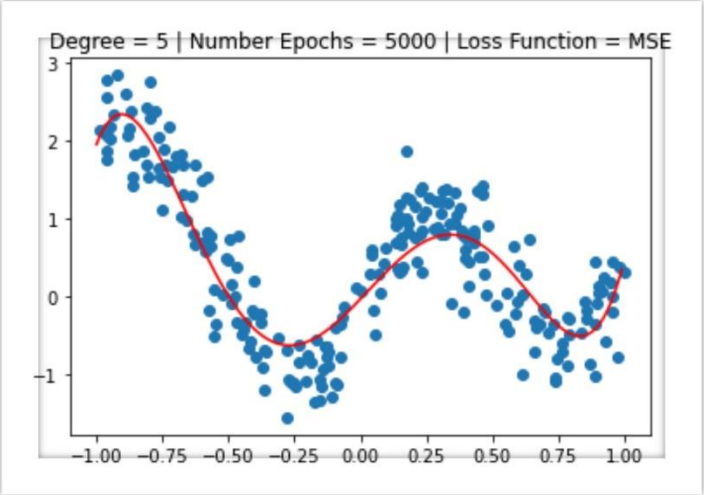


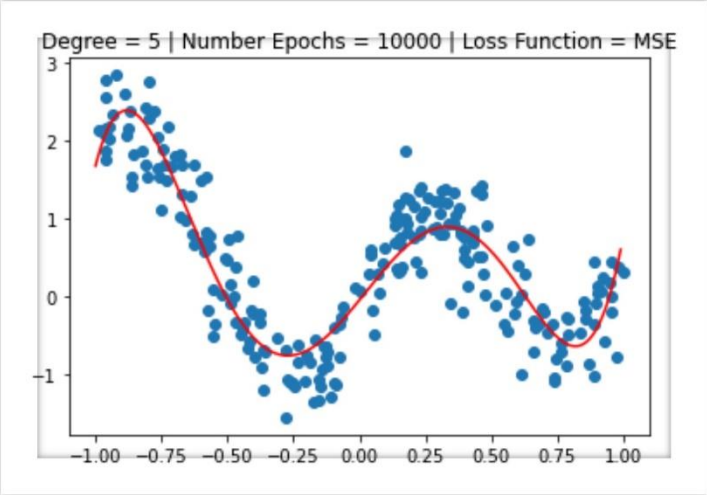
ج) پیش از هر چیز باید اشـــاره کنم که ۸۰٪ داده‌ها برای آموزش و ۲۰٪ برای تســـت انتخاب شـده اسـت. در هر صـفحه نمودارهای مربوط به یکی از سـه درجه ۵، ۸ و ۱۰ ترسیم شده است. در بالای هر نمودار توضیحات مربوط به آن نوشته شده است.

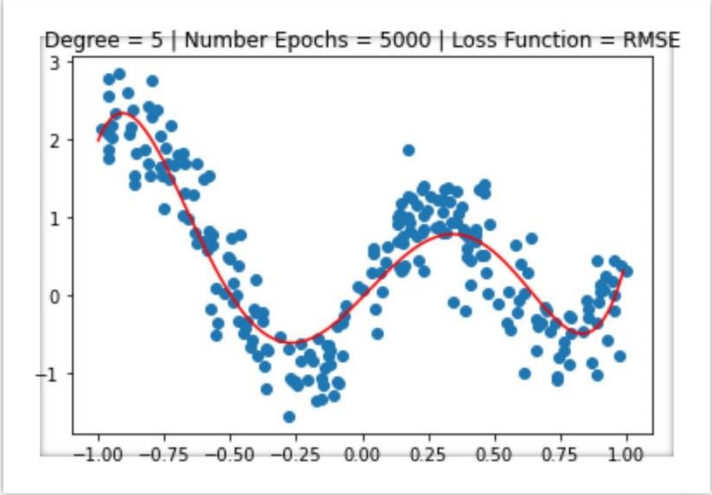
در مورد تحلیل این نمودارها، تفاوت معناداری را نمی‌توان میان این نمودارها یافت و همه این‌ها توانسـته‌اند به یک حالت مناسـب ختم شـوند. یکی از عللی که نمودارهای درجه بالاتر از مسـیر خود منحرف نشـده اسـت این اسـت که در پیاده‌سـازی من، مقدار پارامترهای اولیه برابر صـفر در نظر گرفته شـده اسـت، نرخ یادگیری مقادیر معقولی را دارد، حجم داده‌های آموزش قابل قبول و درصد داده‌های نویز کم است.

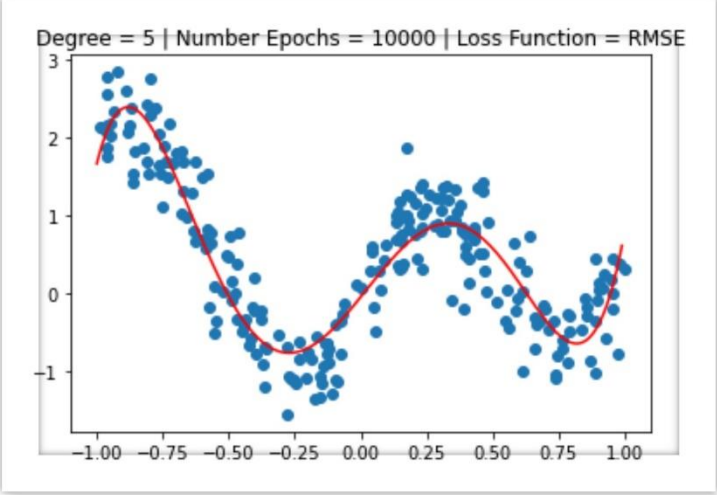
بحث بعدی، بحث بیش‌برازش اسـت. ظاهر نمودارها این مورد را نشـان نمی‌دهد ولی برای بررسی دقیق‌تر باید نمودارهای خطای آموزش و تست را بررسی کرد که در قسمت بعد آورده شده است ولی در همینجا بررسی می‌شود. با نگاه به آن‌ها می‌توان دید که همگرایی به سـرعت رخ می‌دهد و بیش‌برازشـی رخ نداده اسـت. شـاید متفاوت‌ترین نمودارها مربوط به به معیار خطای MAE و در حالت درجه ۵ باشـد. موقعی که تعداد گام برابر با ۵۰۰۰ اسـت، در میانه راه خطای تسـت به کمترین حالت رسـیده اسـت و سـپس افزایش پیدا کرده اسـت که به نظر یک بیش‌برازش کمی به وجود آمده اسـت، ولی با بررسی نمودار ۱۰۰۰۰ به نظر می‌آید که این یک مینیمم محلی است.

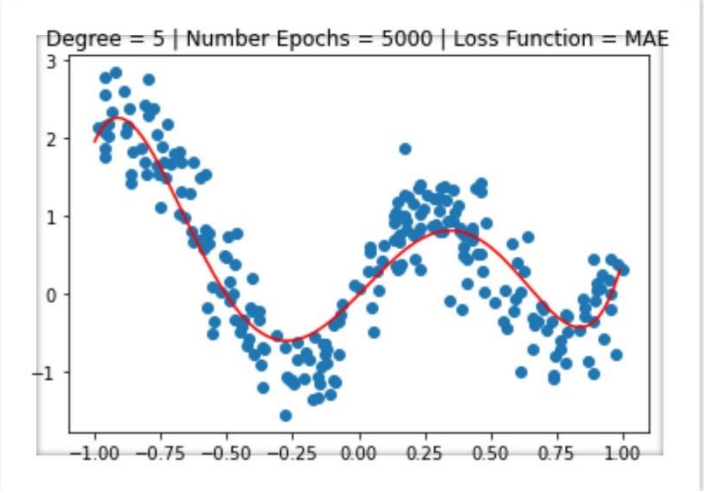
نمودارهای درجه ۵

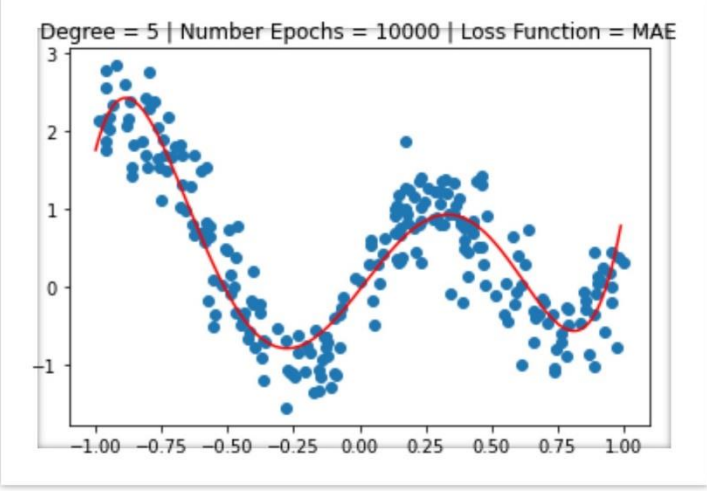




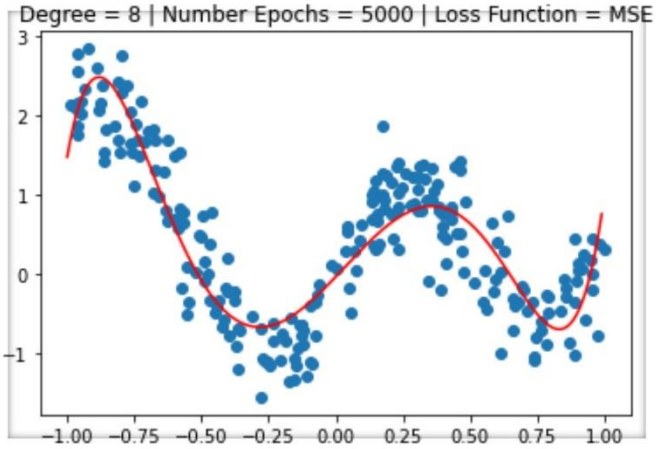


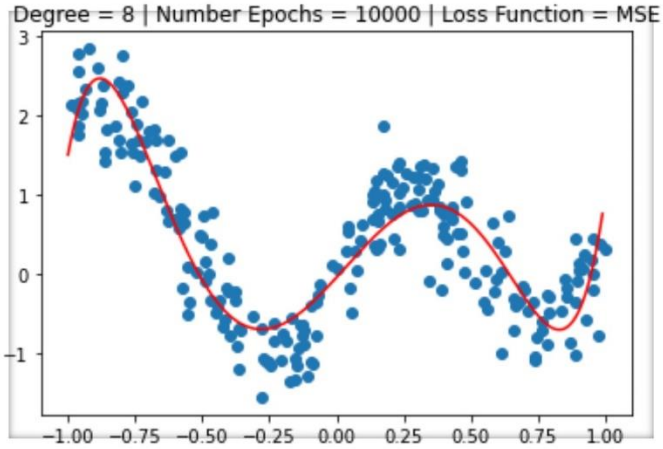


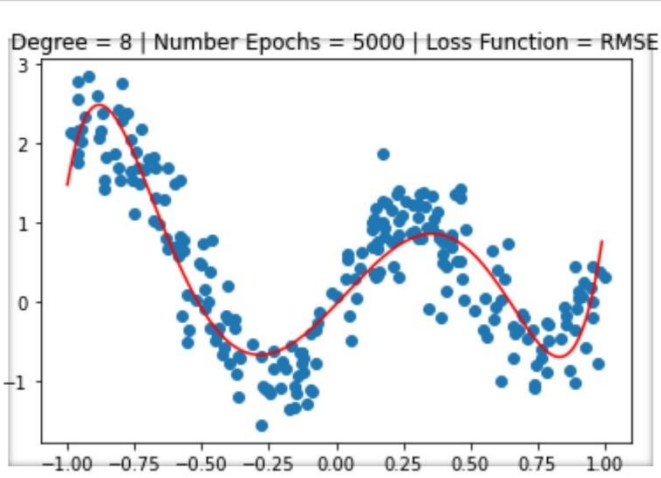


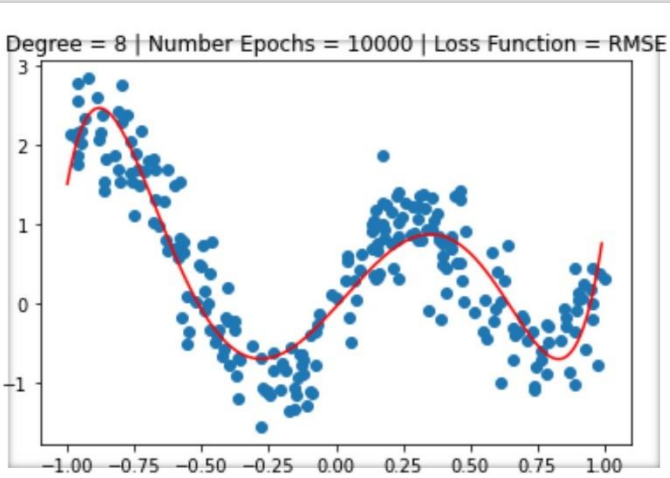


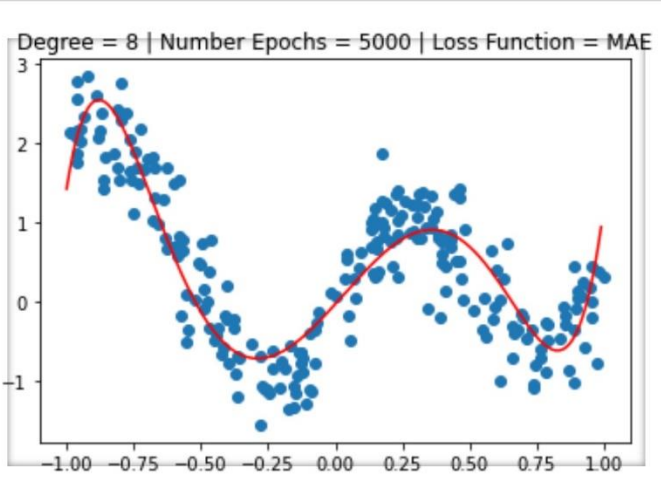
نمودارهای درجه ۸

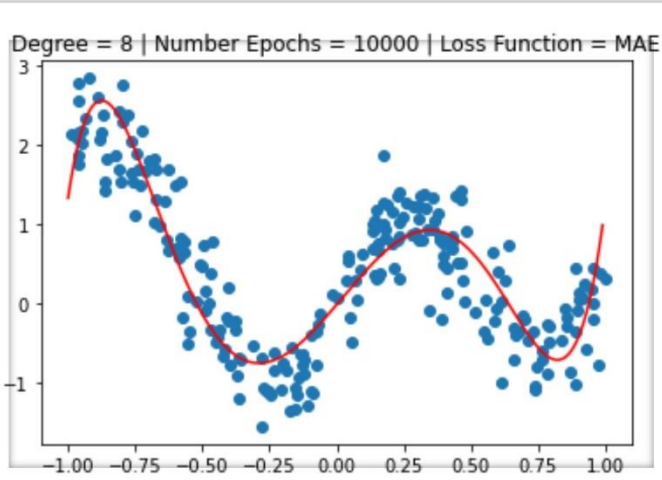




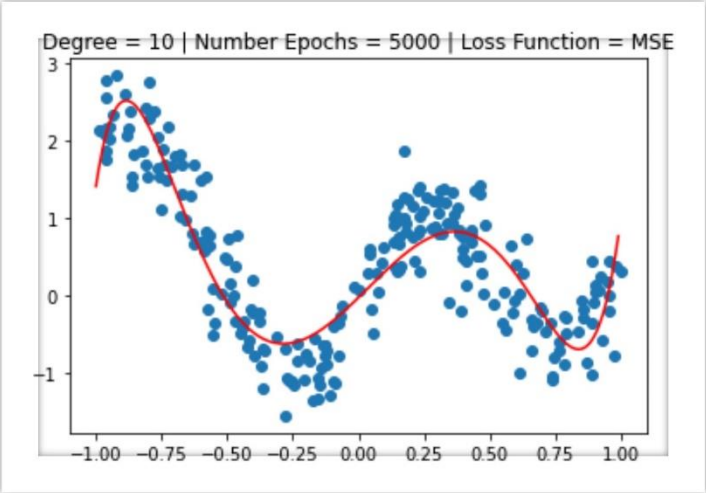


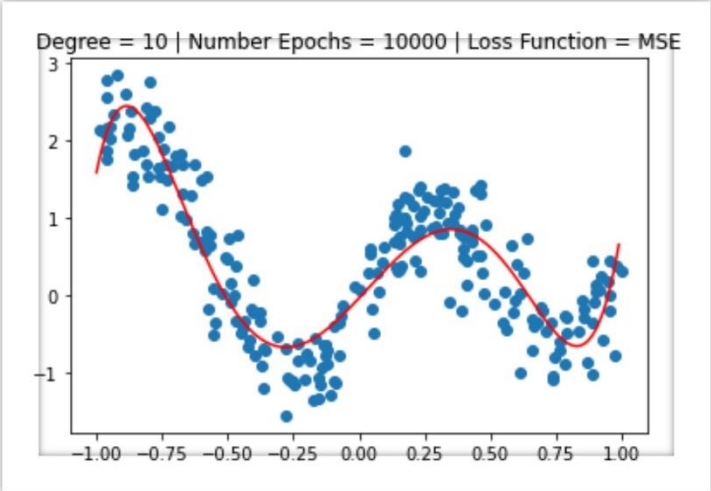


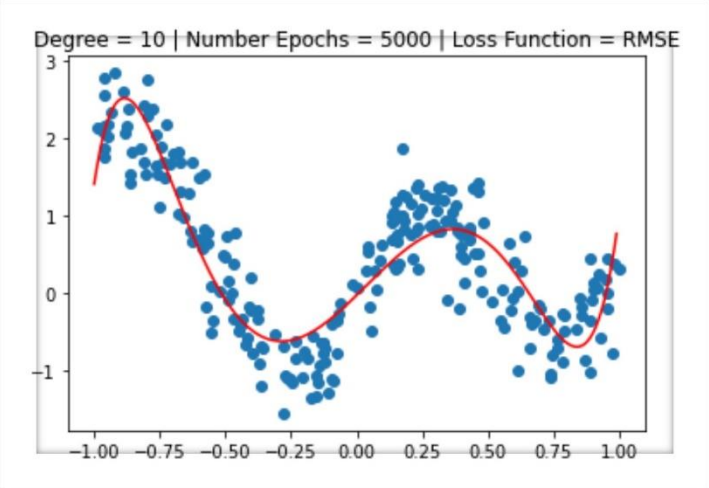


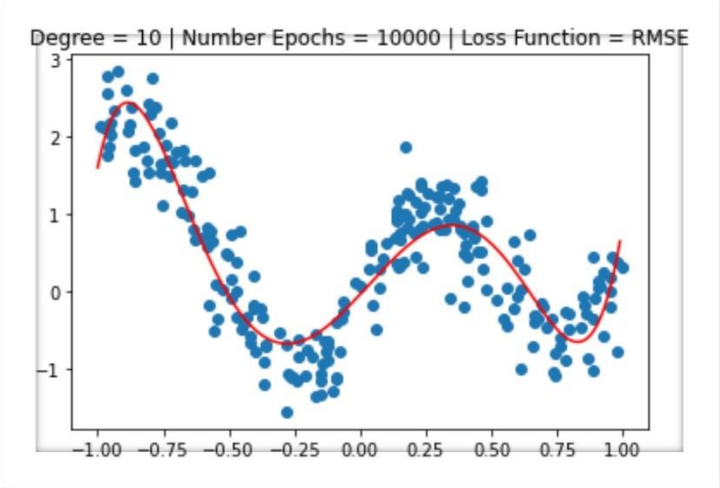


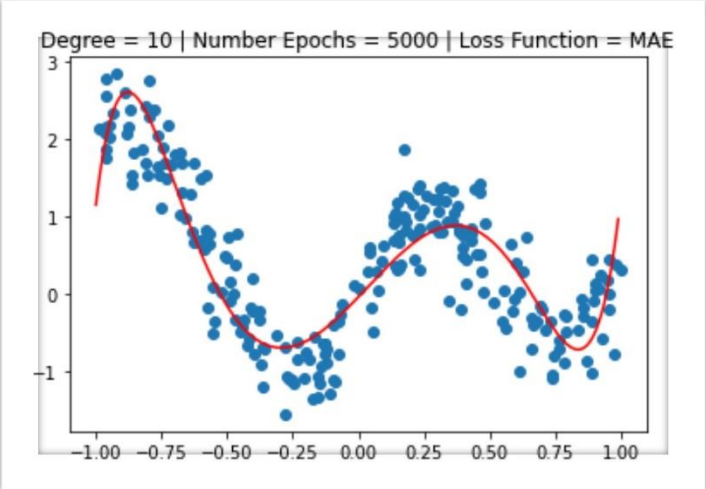
نمودارهای درجه ۱۰

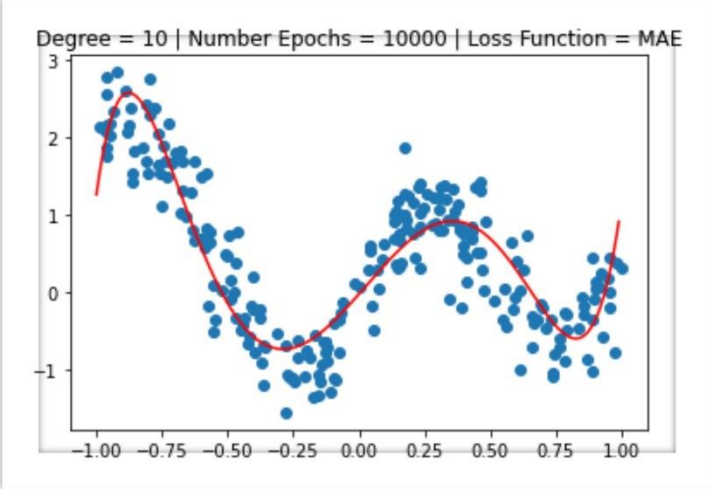






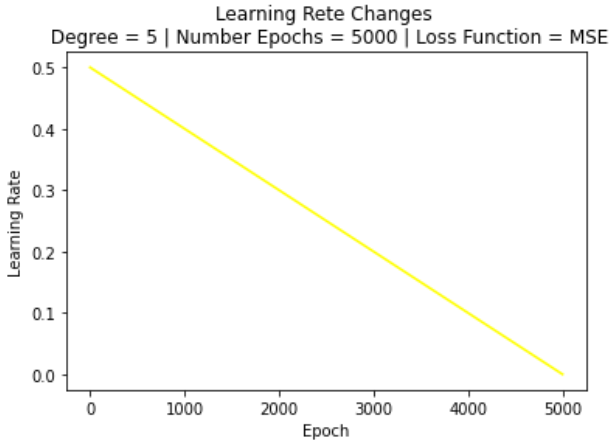


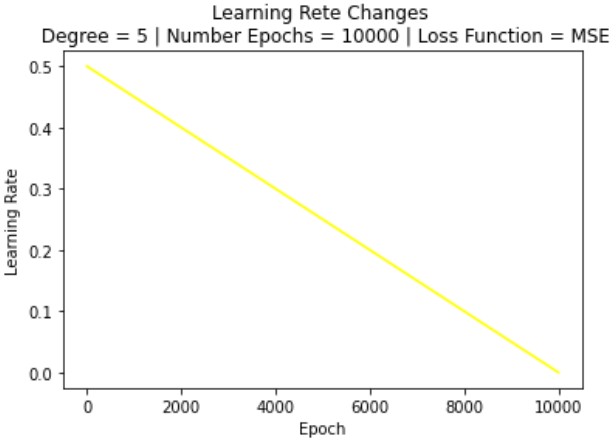




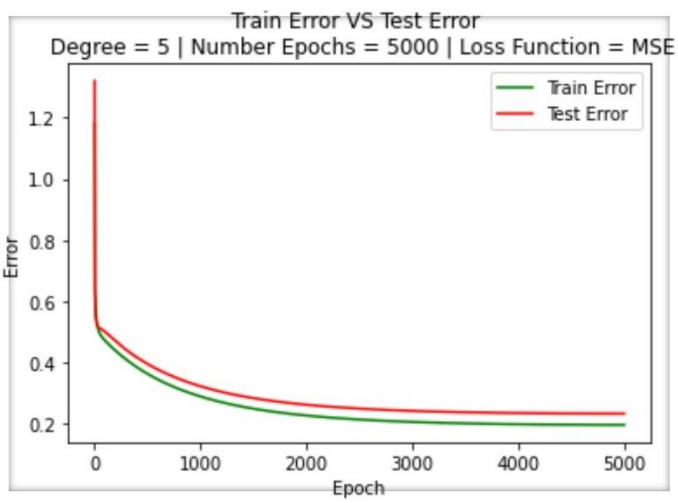
د) ابتدا نمودار اندازه قدم را ارائه می‌دهیم و سپس مشابه قسمت قبل برای هر درجه نمودارهای خطای آموزش و تست آورده می‌شود. مجددا یادآور می‌شوم تحلیل مربوط به خطای آموزش و تست در قسمت قبل آورده شده است.

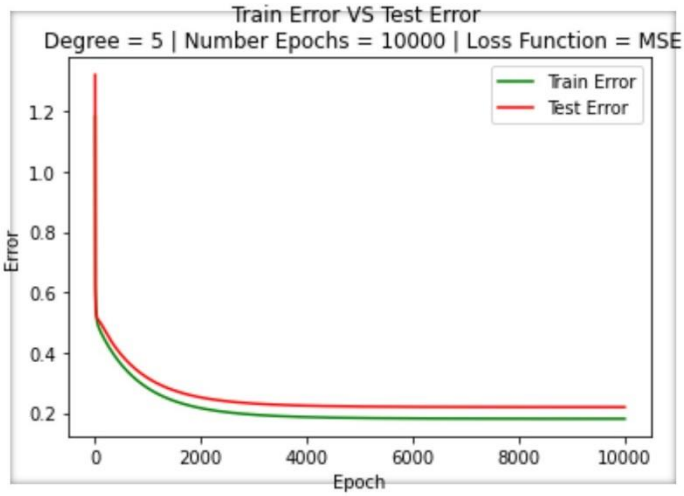
اندازه قدم یا نرخ یادگیری برای تمام حالات ابتدا برابر با ۰/۵ درنظر گرفته شده است و این مقدار به طور ثابت کاهش پیدا می‌کند تا در گام آخر به صفر برسد. لذا نمودارهای اندازه قدم برای تعداد تکرار ۵۰۰۰ تماما یکســـان خواهد بود. برای تعداد تکرار ۱۰۰۰۰ هم همگی مشـــابه خواهند بود. به همین دلیل از هر کدام تنها یک نمونه را در اینجا می‌آوریم.

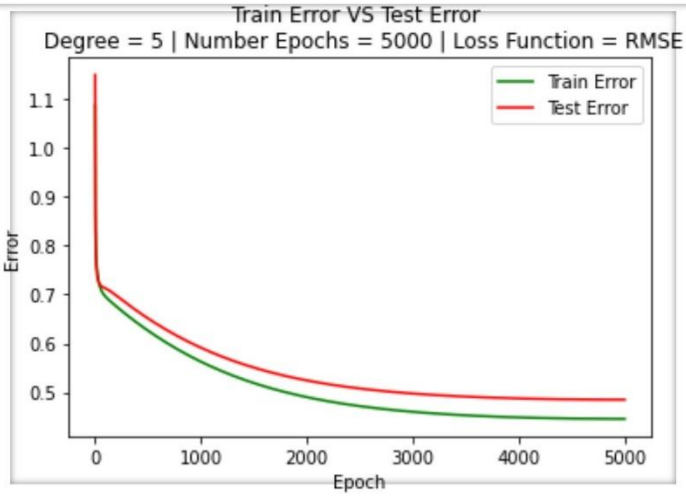


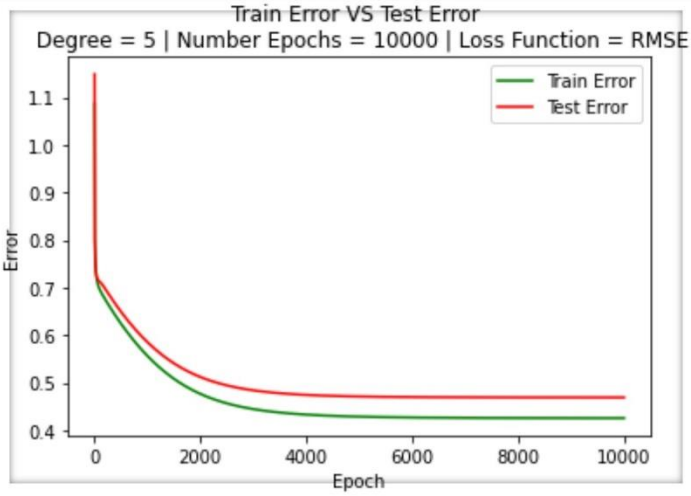


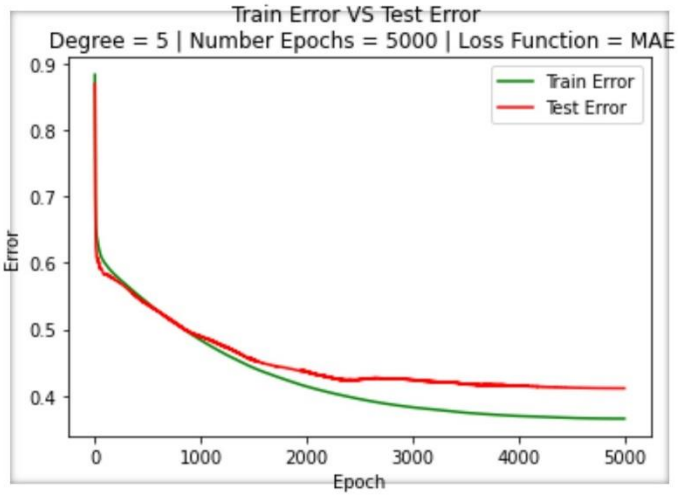
درجه ۵

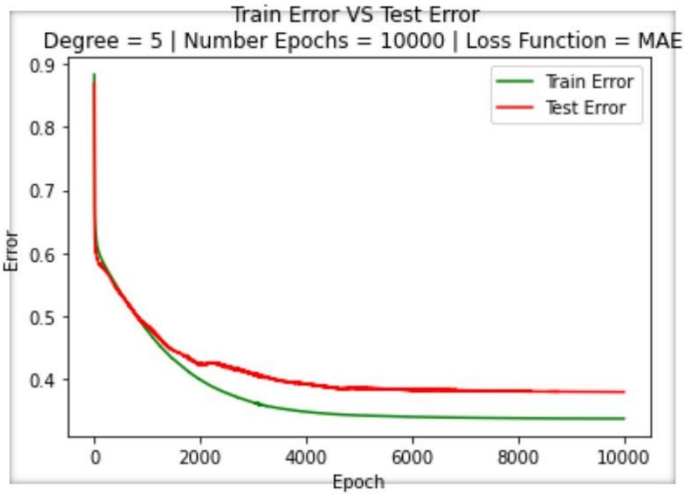

Train Error VS Test Error
Degree = 5 | Number Epochs = 5000 | Loss Function = MSE
Train Error
Test Error
Error
Epoch


Train Error VS Test Error
Degree = 5 | Number Epochs = 10000 | Loss Function = MSE
Train Error
Test Error
Error
Epoch

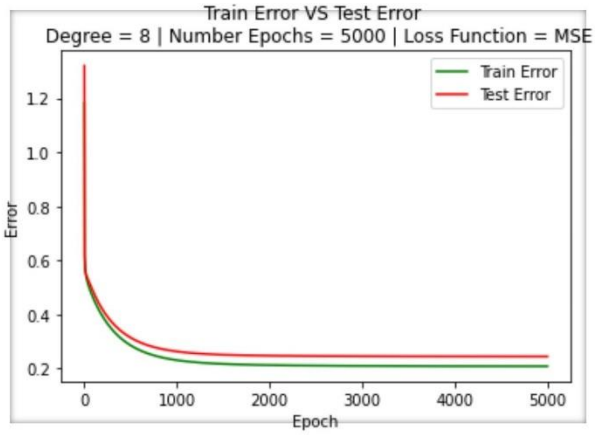

Train Error VS Test Error
Degree = 5 | Number Epochs = 5000 | Loss Function = RMSE
Train Error
Test Error
Error
Epoch

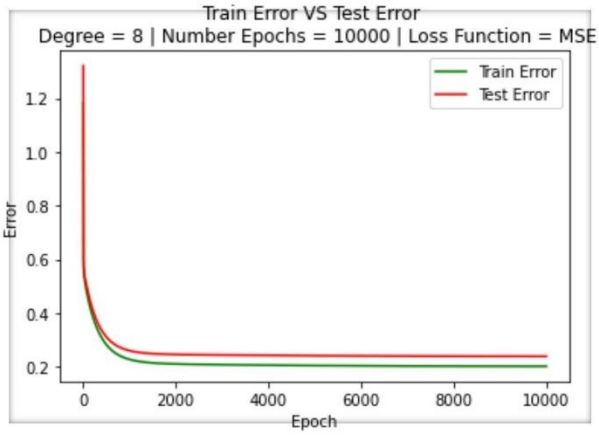

Train Error VS Test Error
Degree = 5 | Number Epochs = 10000 | Loss Function = RMSE
Train Error
Test Error
Error
Epoch

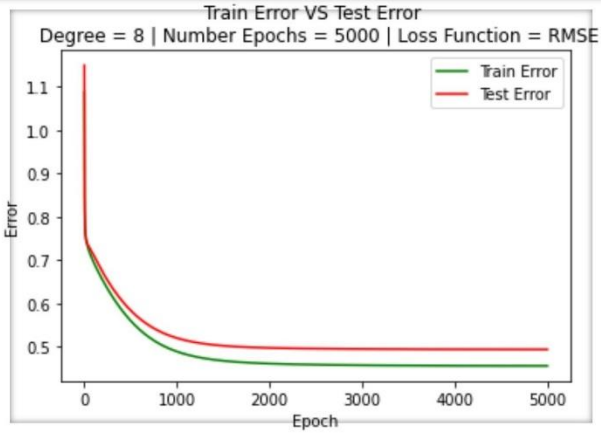

Train Error VS Test Error
Degree = 5 | Number Epochs = 5000 | Loss Function = MAE
Train Error
Test Error
Error
Epoch

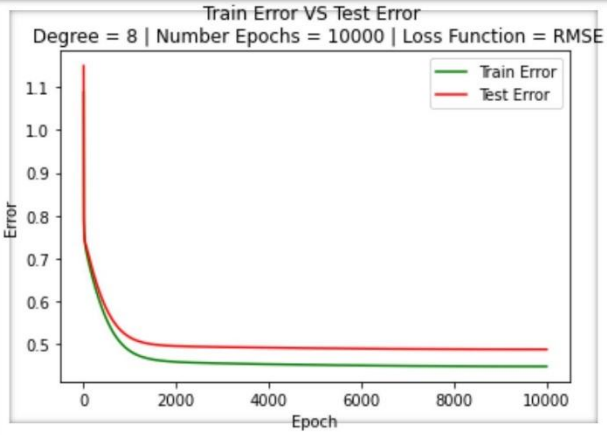

Train Error VS Test Error
Degree = 5 | Number Epochs = 10000 | Loss Function = MAE
Train Error
Test Error
Error
Epoch

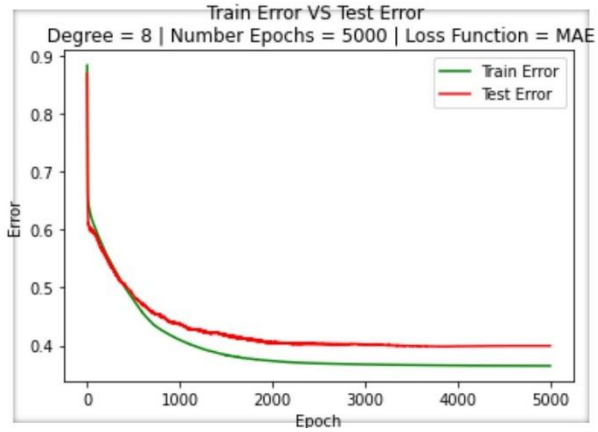
درجه ۸

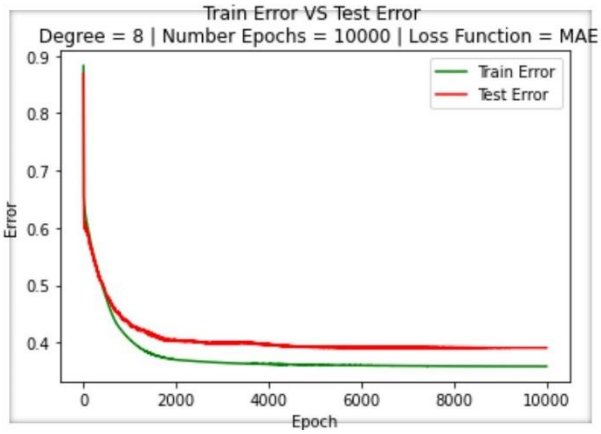

Train Error VS Test Error
Degree = 8 | Number Epochs = 5000 | Loss Function = MSE
Train Error
Test Error
Error
Epoch


Train Error VS Test Error
Degree = 8 | Number Epochs = 10000 | Loss Function = MSE
Train Error
Test Error
Error
Epoch

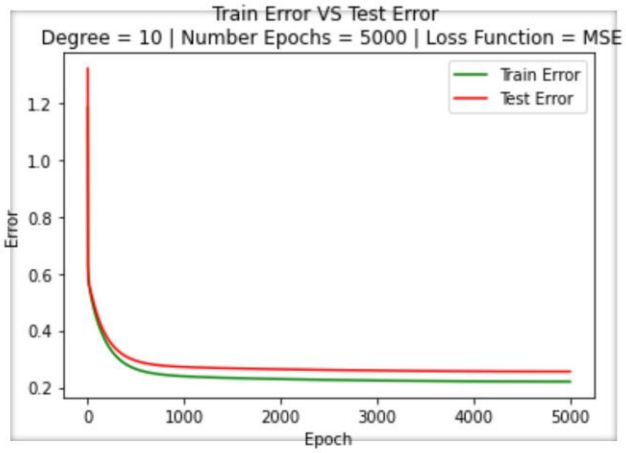

Train Error VS Test Error
Degree = 8 | Number Epochs = 5000 | Loss Function = RMSE
Train Error
Test Error
Error
Epoch

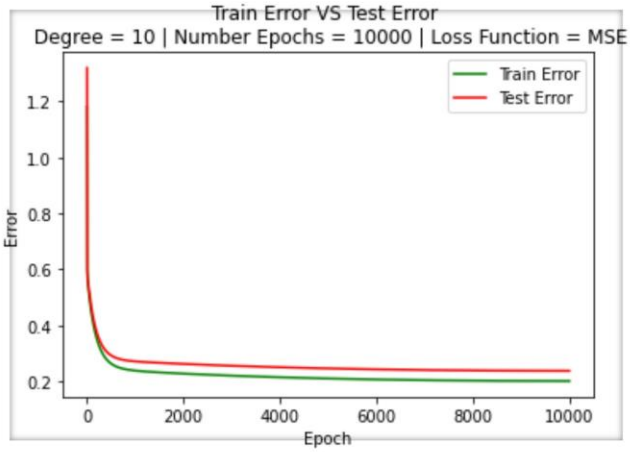

Train Error VS Test Error
Degree = 8 | Number Epochs = 10000 | Loss Function = RMSE
Train Error
Test Error
Error
Epoch

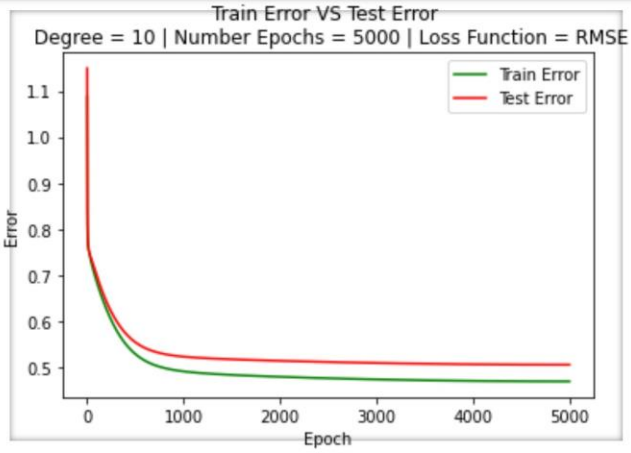

Train Error VS Test Error
Degree = 8 | Number Epochs = 5000 | Loss Function = MAE
Train Error
Test Error
Error
Epoch

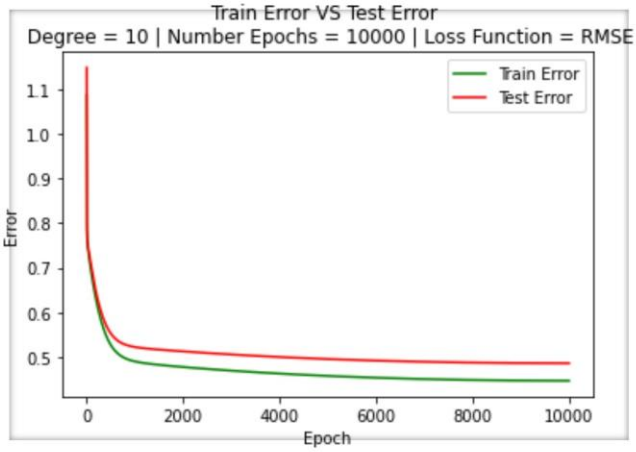

Train Error VS Test Error
Degree = 8 | Number Epochs = 10000 | Loss Function = MAE
Train Error
Test Error
Error
Epoch

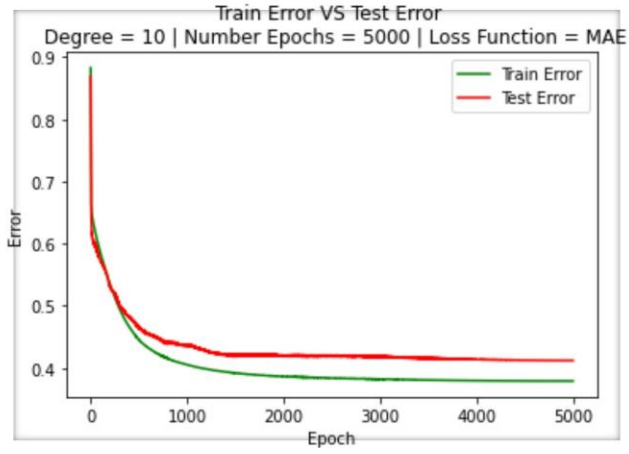
درجه ۱۰

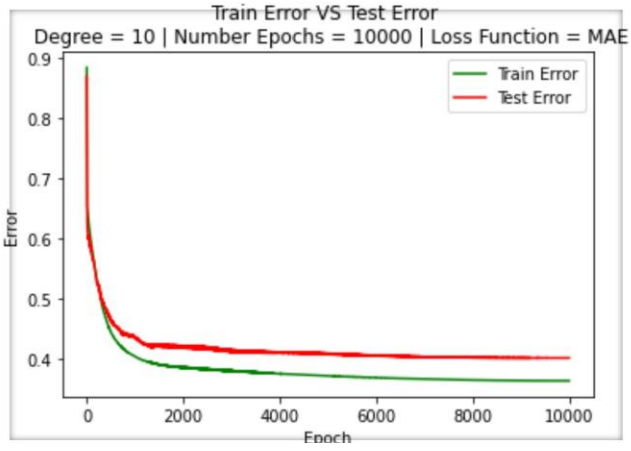

Train Error VS Test Error
Degree = 10 | Number Epochs = 5000 | Loss Function = MSE
Train Error
Test Error
Error
Epoch


Train Error VS Test Error
Degree = 10 | Number Epochs = 10000 | Loss Function = MSE
Train Error
Test Error
Error
Epoch

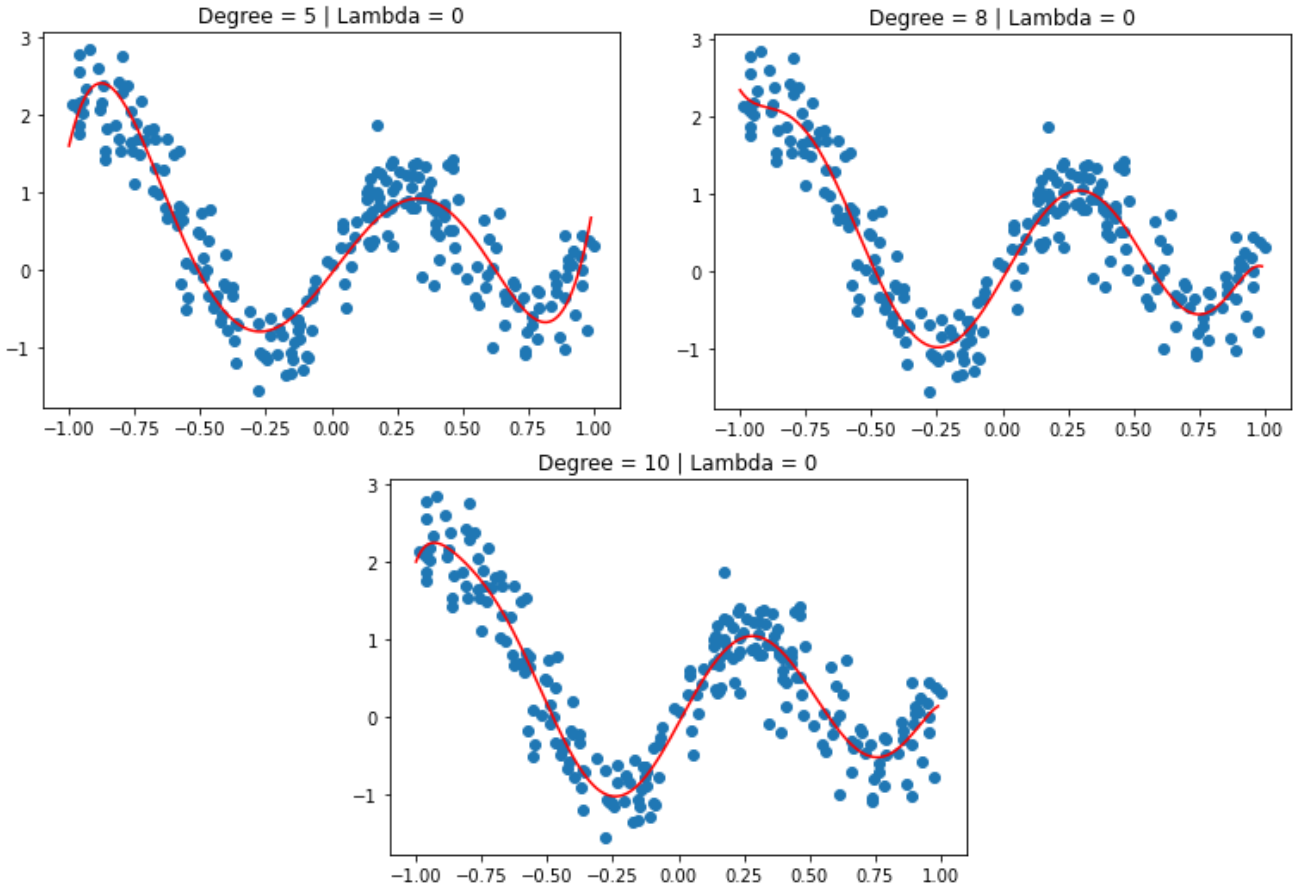

Train Error VS Test Error
Degree = 10 | Number Epochs = 5000 | Loss Function = RMSE
Train Error
Test Error
Error
Epoch


Train Error VS Test Error
Degree = 10 | Number Epochs = 10000 | Loss Function = RMSE
Train Error
Test Error
Error
Epoch


Train Error VS Test Error
Degree = 10 | Number Epochs = 5000 | Loss Function = MAE
Train Error
Test Error
Error
Epoch


Train Error VS Test Error
Degree = 10 | Number Epochs = 10000 | Loss Function = MAE
Train Error
Test Error
Error
Epoch

ه) در نمودارهای زیر، نمودار برای سه درجه ۵ و ۸ و ۱۰ با مقدار پارامتر لاندا ۰ ترسیم شده است. از نظر ظاهر تفاوت‌هایی با گرادیان نزولی دیده می‌شود. در حول مقدار x برابر با ۰/۲۵- و ۰/۲۵ تمام نمودارهای بدست آمده از معادله نرمال از میانه داده‌های این دو قسمت گذشته است درحالی که در نمودارهای گرادیان نزولی این عبور از سمت حاشیه بوده است. تفاوت دیگر در نمودار درجه ۸ و تاحدی در درجه ۱۰ وجود دارد و آن این است که در ابتدای نمودار و در مقدار x برابر با ۱- سر نمودار برخلاف قسمت قبل، به سمت بالاست.



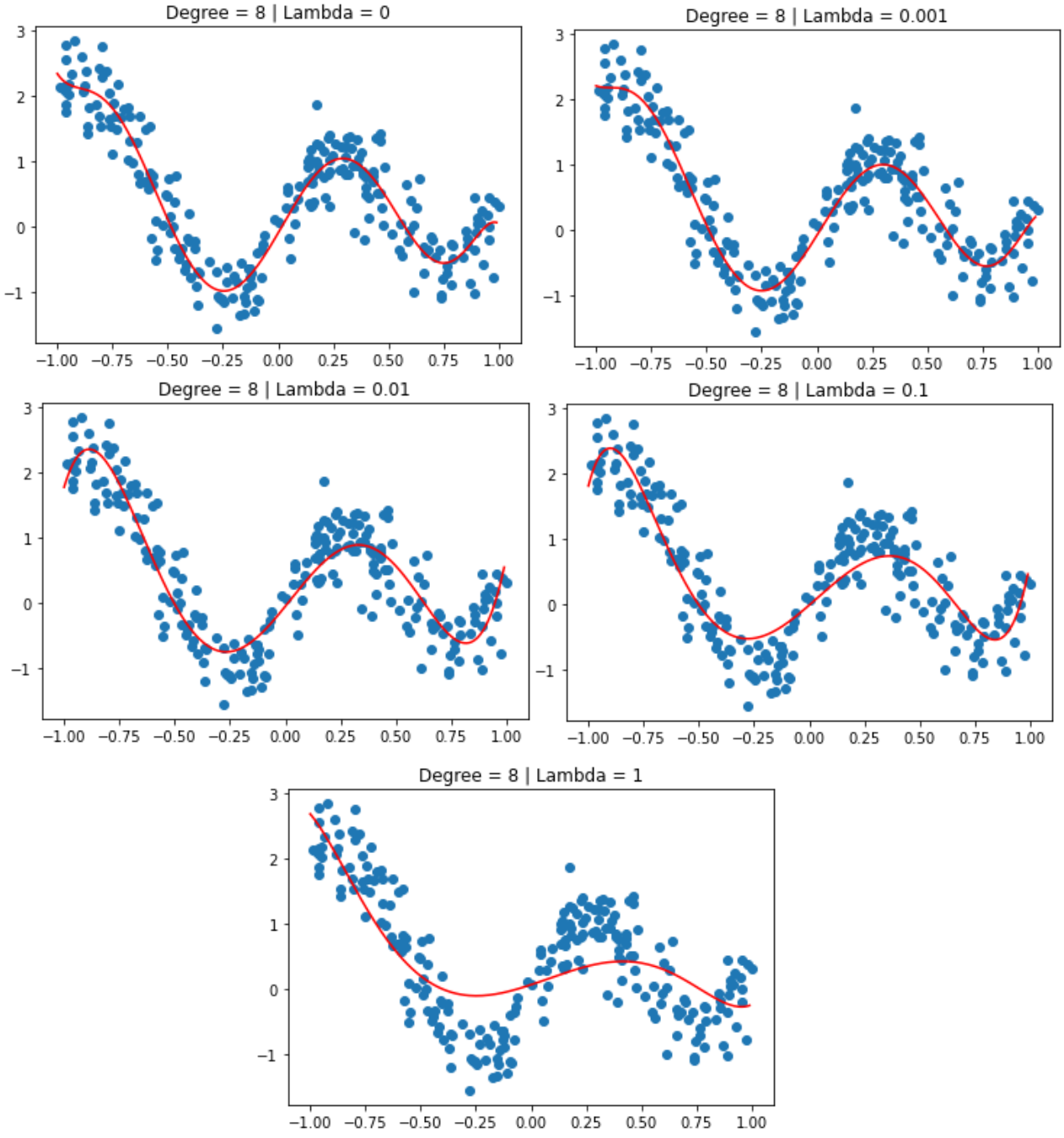
برای درک و مقایسه بهتر می‌توان میزان خطای RMSE این سه نمودار و نمودارهای قسمت گرادیان نزولی را بررسی کرد. مقادیر مربوط به آن را در جدول زیر آورده‌ایم. نکته قابل تعمل در جدول زیر آن است که مدل‌های گرادیان نزولی برای درجه ۵ بهترین دقت آموزشی و تست را داشته است و برای درجه‌های بالاتر مدل پس‌رفت کرده است. همچنین تعداد گام بیشتر منجر به آموزش بهتر شده است. برای معادله نرمال درجه

۸ بهترین دقت تست را داشته است و بعد از آن مدل دچار بیش‌برازش شده است. در همین حال مطابق انتظار می‌بینیم که معادله نرمال همواره خطای مناسب‌تری نسبت به گرادیان نزولی داشته است.

		درجه ۵	درجه ۸	درجه ۱۰
گرادیان نزولی ۵۰۰۰ گام	خطای آموزشی	۰/۴۴۵	۰/۴۵۵	۰/۴۶۹
	خطای تست	۰/۴۸۴	۰/۴۹۳	۰/۵۰۵
گرادیان نرولی ۱۰۰۰۰ گام	خطای آموزشی	۰/۴۲۵	۰۴۴۸	۰/۴۴۶
	خطای تست	۰/۴۶۸	۰/۴۸۸	۰/۴۸۵
معاله نرمال	خطای آموزشی	۰/۴۲۴	۰/۳۸۴	۰/۳۸۲
	خطای تست	۰/۴۶۸	۰/۴۳۴	۰/۴۳۸

و) برای این قسمت مقادیر ۰/۰۰۱، ۰/۰۱، ۰/۱ و ۱ را ترسیم کرده‌ایم. همچنین نمودار درجه ۸ قسمت قبل را هم برای قیاس بهتر آورده‌ایم. نهایتا یک جدول از مقادیر خطا را هم تهیه می‌کنیم تا به صورت دقیق بتوان نتایج را مقایسه کرد. با مقایسه نمودارها در می‌یابیم که مطابق انتظار افزایش شدید لاندا باعث می‌شود تا نمودار به نرمی تغییراتی را داشته باشد. با بررسی جدول هم می‌بینیم که مقدار صفر برای لاندا از حالات دیگر بهتر است! این یعنی نمودار در حالت عادی اصلا دچار بیش‌برازش نشده است که نیازی به منظم‌سازی وجود داشته باشد. چیزی که در قسمت قبل هم مشاهده شد. شاید اگر این سوال برای درجه ۱۰ مطرح می‌شد، استفاده از مقادیر پایین لاندا نسبت به لاندا صفر می‌توانست منجر به نتایج بهتری شود.

	خطای آموزش	خطای تست
$\lambda = 0$	۰/۳۸۴	۰/۴۳۴
$\lambda = 0.001$	۰/۳۸۷	۰/۴۳۷
$\lambda = 0.01$	۰/۴۲۱	۰/۴۶۶
$\lambda = 0.1$	۰/۴۷۱	۰/۵۰۵
$\lambda = 1$	۰/۶۲۲	۰/۶۴۱


Degree = 8 | Lambda = 0
Degree = 8 | Lambda = 0.001
Degree = 8 | Lambda = 0.01
Degree = 8 | Lambda = 0.1
Degree = 8 | Lambda = 1

سوال ۲

الف) ابتدا باید بررسی کنیم که کدام ستون‌ها دارای مقادیر گم‌شده هستند و تعدادشان چقدر است. در جدول زیر می‌توانید ستون‌های شامل مقادیر گم‌شده را ببینید. برای پرکردن مقادیر گم‌شده از میانگین مقادیر یک ستون استفاده کردیم.

نام ستون	تعداد داده گم‌شده
Budget	۱
Screens	۱۰
Aggregate Followers	۳۵

برای پیش‌بردازش هم از نرمال‌سازی مطابق با فرمول زیر برای هر ستون استفاده کردیم تا تمامی مقادیر فارغ از مفهوم ستون در بازه صفر تا یک قرار بگیرد و برای یادگیری مدل در قسمت‌های بعد هم مشکلی پیش نیاید.

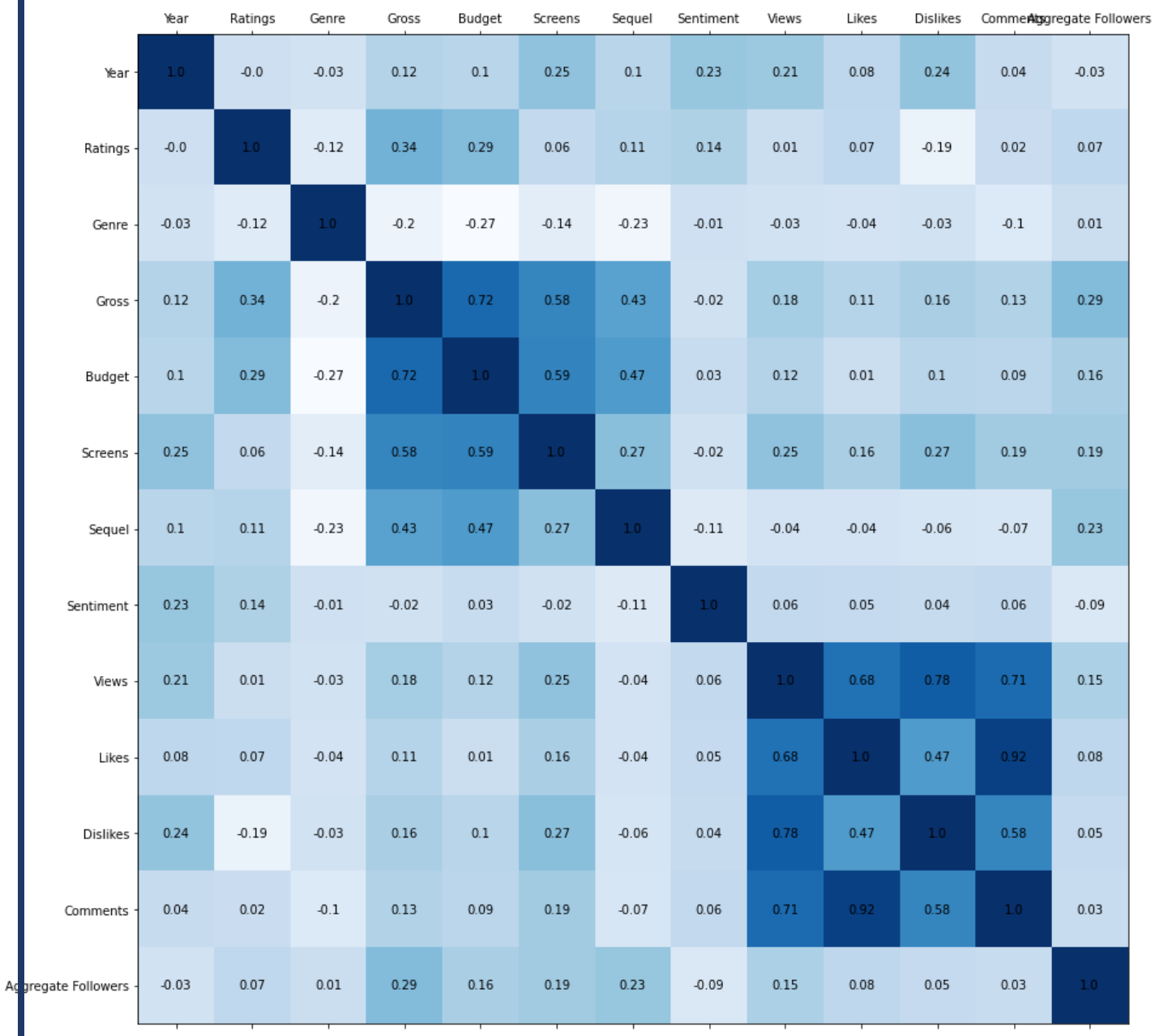
$$x_{normal} = \frac{x - \min x_i}{\max x_i - \min x_i}$$

همچنین ستون Movie که شامل نام فیلم‌ها بود و به نوعی نقش آیدی داشت را حذف کردیم. در ابتدا به نظر می‌رسید حذف ستون Genre هم نتایج را بهتر کند ولی در عمل اینطور نشد و این ستون را حذف نکردیم.

ب) در ماتریس color map زیر همبستگی بین ویژگی مشخص است. با بررسی مقادیر بالای ۰/۷ در نمودار می‌توان برخی از ویژگی‌ها را حذف کرد. پس از بررسی‌هایی که انجام دادم به نظر آمد حذف تنها ویژگی Comments به دلیل همبستگی بالای آن با Likes معقول باشد.

در این حال به حذف ویژگی‌های دیگر هم فکر شد. مثلا در آزمایشی ویژگی Budget را به دلیل همبستگی نسبتا بالا با ویژگی Gross حذف کردم (چون Gross همبستگی خوبی با ویژگی هدف داشت حذف نشد و Budget حذف شد.) در آزمایشی دیگر هم

تصـــمیم گرفتم بین چهـار ویژگی Views، Likes، Dislikes و Comments فقط دو ویژگی Likes و Dislikes را نگه‌دارم چراکه به نظر می‌آمد داشـتن این دو ویژگی از این مجموعه می‌تواند کافی باشد.

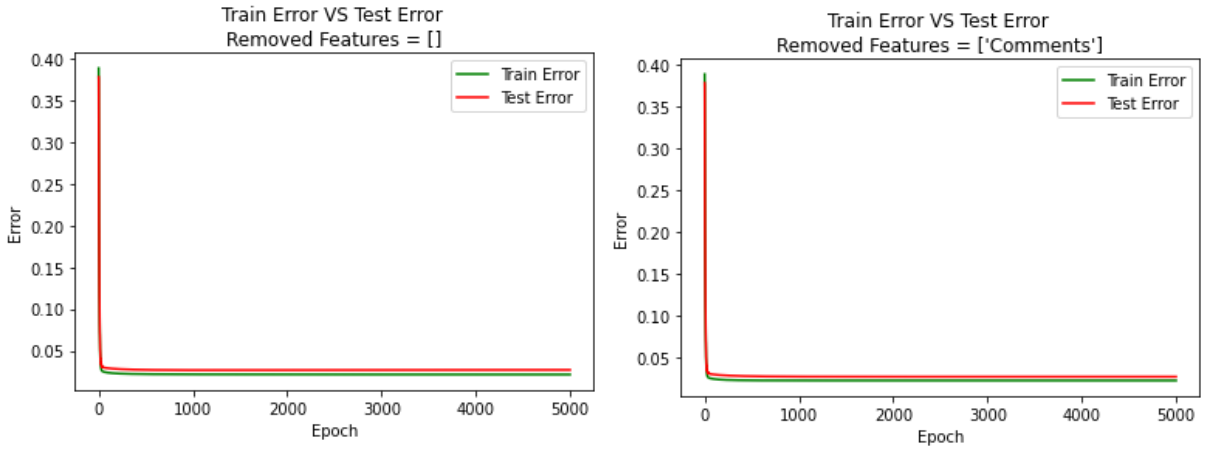


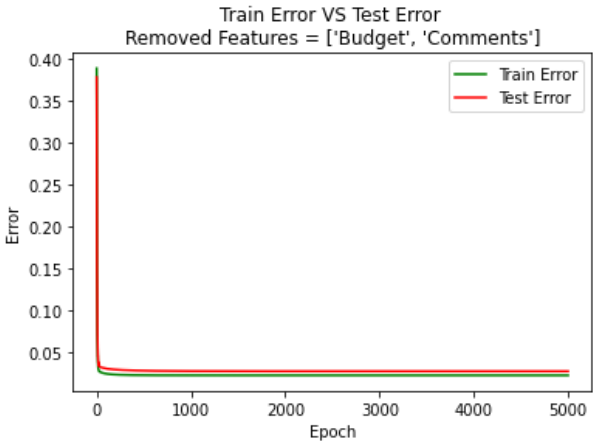
د) باتوجه به به ابعاد بالای مسئله امکان ترسیم گرافیکی مدل وجود ندارد ولی می‌توان معادله بدست آمده را ترسیم کرد. برای حالتی که تنها ویژگی Comments حذف شده باشد چنین معادله‌ای حاصل شد:

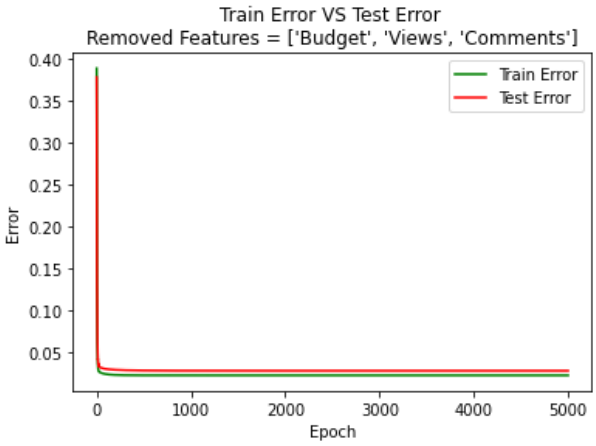
Ratings = -0.03 × Year + 0.02 × Genre + 0.48 × Gross + 0.12 × Budget - 0.10 × Screens - 0.10 × Sequel + 0.25 × Sentiment + 0.33 × Views + 0.17 × Likes - 0.90 × Dislikes + 0.02 × Aggregate Followers + 0.43

این معادله هم حداقل در ظاهر مناسب به نظر می‌رسد چراکه می‌بینیم ویژگی مانند Dislike تاثیر منفی بسیار زیادی را داشته است و از طرفی ویژگی‌هایی مانند Views و Gross مطابق انتظار تاثیر مثبت زیادی گذاشته است. (باتوجه به نرمال‌سازی انجام شده مقادیر هر ویژگی بین ۰ تا ۱ است و عددی با مقدار 0.90 زیاد محسوب می‌شود.)

از تابع خطا MSE و ۵۰۰۰ گام برای آموزش مدل استفاده کرده‌ایم. نمودار خطای آموزش و تست برای حالت‌های بررسی‌شده و همچنین جدولی از مقادیر نهایی خطا در ادامه آورده شده است:



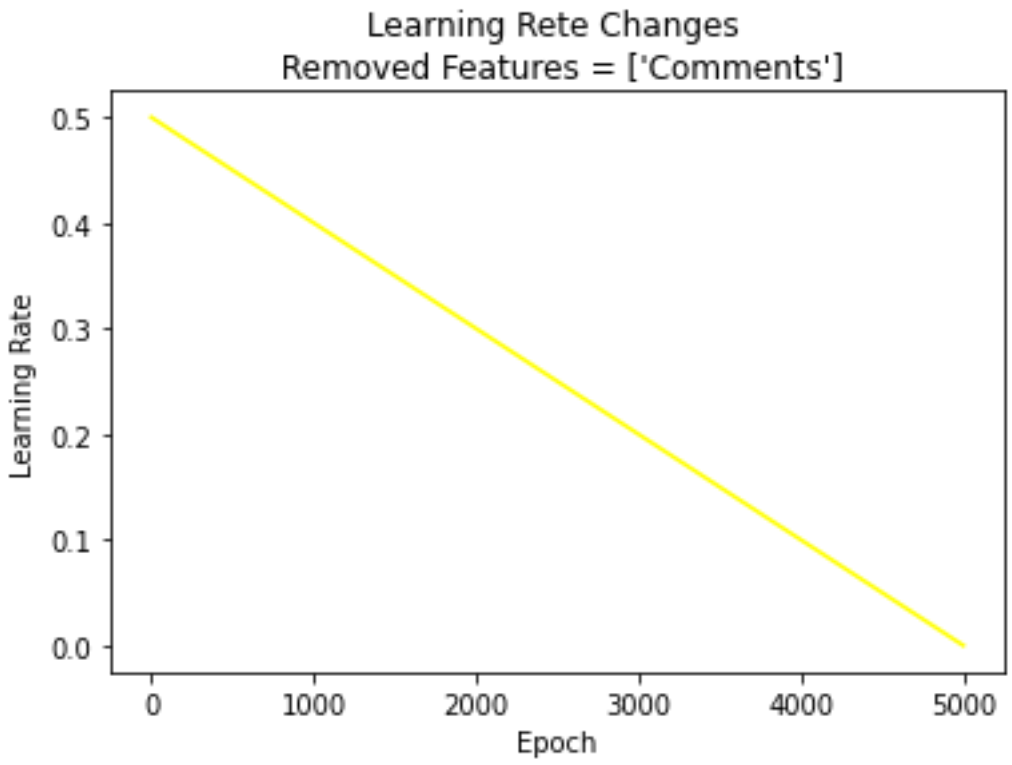




	خطای آموزش	خطای تست
تمام ویژگی‌ها	۰/۰۲۱۹	۰/۰۲۶۸
حذف ویژگی Comments	۰/۰۲۱۷	۰/۰۲۶۲
حذف ویژگی‌های Budget و Comments	۰/۰۲۱۹	۰/۰۲۶۸
حذف ویژگی‌های Budget، View و Comments	۰/۰۲۲۵	۰/۰۲۸۰

با توجه به جدول و نمودارها متوجه می‌شـــویم که خطا و پیش‌بینی مدل چندان تفاوتی بین حالات مختلف ندارد و اگر در شرایطی بودیم که محدودیت حافظه و زمان داشـتیم می‌توان سـه ویژگی را بدون افت جدی دقت حذف کرد ولی اگر دقت برای ما اهمیت بیشـتری داشـته باشـد حذف تنها ویژگی Comments ما را به بهترین نتایج می‌رساند. نکته جالب آن است که حذف این ویژگی از نگه‌داشتن آن و حالتی که تمام ویژگی‌ها را داریم بهتر است!

نهایتا نوبت به نمودار تغییرات نرخ یادگیری اســـت. نرخ یادگیری مانند ســـوال قبل کاهش پیدا می‌کند و نکته جدیدی ندارد. در تمام آزمایش‌های این سوال هم تغییرات آن یکسان است. در شکل زیر می‌توانید نمودار تغییرات نرخ یادگیری برای یک حالت را مشاهده کنید:


Learning Rete Changes
Removed Features = ['Comments']
Learning Rate
0.5
0.4
0.3
0.2
0.1
0.0
0
1000
2000
3000
4000
5000
Epoch

سوال ۳

الف)[5] دو روش مذکور برای تبدیل یک متغیر رشته‌ای که تعداد محدودی حالت داشته باشد به یک متغیر صحیح انجام می‌شود. مثلا برای یک ویژگی که شامل مقطع تحصیلات است قابل استفاده است. بدین شکل در ادامه می‌توان با ویژگی‌های عددی کار کرد که راحت‌تر است.

- در روش integer encoding هر یک از مقادیر رشته‌ای را با یک عدد صحیح متناظر می‌کنند.
- در روش one hot encoding یک بردار باینری درنظر گرفته می‌شود که تعداد ابعاد آن برابر با حالات مختلف مقدار برای ویژگی مدنظر است. هر بعد را متناظر یا یک مقدار درنظر می‌گیرند. سپس هر مقدار رشته‌ای را تبدیل به یک بردار تمام صفر می‌کنند که تنها همان بعد مرتبط یک خواهد بود.

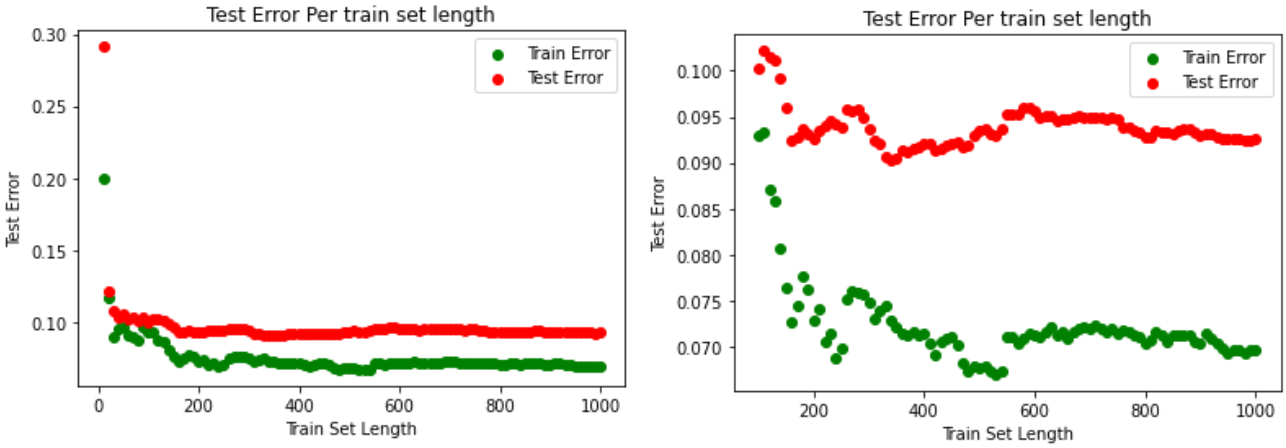
ج) برای متغیرهای رشته‌ای که تنها دو مقدار دارند استفاده از integer encoding معقول است ولی وقتی تعداد حالات ممکن بیشتر از دو باشد ممکن است به دلیل آنکه فضای عددی دارای ترتیب و سایر عملیات عددی است یک مفهوم غلط در مجموعه‌داده به وجود بیاید. به ویژگی region توجه کنید. دارای چهار مقدار است که ترتیب بین آن نمی‌توان درنظر گرفت حال اگر اعداد ۰ تا ۳ را به آن نسبت می‌دادیم دارای یک ترتیبی می‌شدند و این باور به وجود می‌آمد که مقدار ۳ سه برابر مقدار ۱ در این ویژگی است.

د) قبل از هرچیز لازم است اشاره کنم پیش از اجرای الگوریتم‌های این سوال، از نرمال‌سازی ویژگی‌ها مطابق با سوال قبل استفاده شده است. همچنین در این سوال ویژگی region را به چهار ویژگی تقسیم کرده‌ام. همچنین برای جلوگیری از خطای‌های احتمالی مقدار بسیار کمی برای لاندا (۰/۰۰۰۱) برای این قسمت و قسمت بعد درنظر

[5] https://machinelearningmastery.com/how-to-prepare-categorical-data-for-deep-learning-in-python/

گرفتم. مقدار خطای آموزش برابر با ۰/۰۶۹ و مقدار خطای تســـت برابر با ۰/۰۹۲ بوده است.

ه) خطای تســـت و آموزش به ازای اندازه‌های مختلف مجموعه آموزش به شـــرح زیر اســت. برای درک بهتر نمودار یک نمودار مشـابه از ۱۰۰ داده به بعد هم ترسـیم شـده است:



از این نمودارها می‌توان برداشـــت کرد که باید مجموعه‌داده آموزشـــی به اندازه کافی بزرگ باشـــد تا به دقت مناســـب تســـت برســـیم و از یک نقطه به بعد مجموعه‌داده آموزشی بزرگ‌تر تاثیر کمتری خواهد داشت.

و) با اســـتفاده از ۵۰ هزار گام و معیار MSE نمودار زیر برای خطای آموزش و تســـت بدســـت آمد. مقدار نهایی خطای تســـت برابر با ۰.۰۲۳ و مقدار نهایی خطای آموزش برابر با ۰/۰۰۸ شد. باتوجه به اینکه مقدار این دو خطا کمتر از خطای معادله نرمال‌شده اسـت به نظر می‌رسـد که پیاده‌سـازی یکی از دو قسـمت این سـوال دارای خطا اسـت! همچنین نمودار تغییرات خطای آموزش و تسـت در قسـمت بعد آورده شـده اسـت. تفاوتی که این نمودار با نمودارهای گرادیان نزولی ســـوال قبل دارد در این اســـت که نمودار خطای تست بسیار مواج است. علت این امر این است که در گرادیان این سوال

به ازای هر داده پارامترهای مدل بروز می‌شود که این بروزرسانی نسبت به بروزرسانی با کل مجموعه‌داده خشن‌تر است.