بسمه تعالی

مقدمه‌ای بر یادگیری ماشین

نیمسال اول ۹۸-۹۷

تمرین عملی سری سوم

مدرس: صابر صالح

- داده‌ای این تمرین را از این لینک می‌توانید دریافت کنید.
- برای این که مطمئن شوید داده‌ها بدون مشکل دانلود شده است، می‌توانید در ترمینال سیستم دستور زیر را اجرا کنید. خروجی باید به شکل زیر باشد:

```
md5sum MNIST_png.tar.gz

1b8a0a35ea6d3cd16d10d9eecd7e9570  MNIST_png.tar.gz
```

- این تمرین در تاریخ 18 آبان 1397 بارگذاری شده و به مدت 21 روز برای آن فرصت دارید. موعد تحویل تمرین تا 9 **آذر** 1397 **ساعت** $23 : 59' : 59''$ می‌باشد.
- نمره‌ی تمرین بر اساس دقت شما، کدی که برای تمرین آماده کرده‌اید و نیز بر اساس گزارش ارائه شده محاسبه خواهد شد.
- در این تمرین شما باید مفاهیمی از یادگیری ماشین را مطالعه کنید و چکیده‌ای از آن را در گزارشتان بیاورید. برای مطالعه می‌توانید از منابعی که در اینترنت وجود دارد استفاده کنید.
- در مورد تقلب و کپی‌برداری از کار دیگران، در هر بخش از تمرین مطابق سیاست‌های درس رفتار خواهد شد. می‌توانید از منابع اینترنتی برای حل سؤال کمک بگیرید ولی به هیچ عنوان کدی را کپی نکنید. سعی کنید که منظور کد را متوجه شوید. سپس مجدداً، آن را پیاده سازی کنید.

۱ مقدمه

هدف از این تمرین آشنایی با حالاتی است که الگوریتم یادگیری مطابق آنچه که انتظار دارید عمل نمی‌کند. در ادامه چند دلیل که باعث می‌شود این اتفاق رخ بدهد را بررسی می‌کنیم. سپس در تمرین خود در نظر داشته باشید که این اتفاقات رخ ندهد.

۲ دلایلی که ممکن است الگوریتم یادگیری مطابق انتظار عمل نکند

اتفاقات متعددی ممکن است منجر به عملکرد پایین در الگوریتم یادگیری شود.

۱.۲ Noise

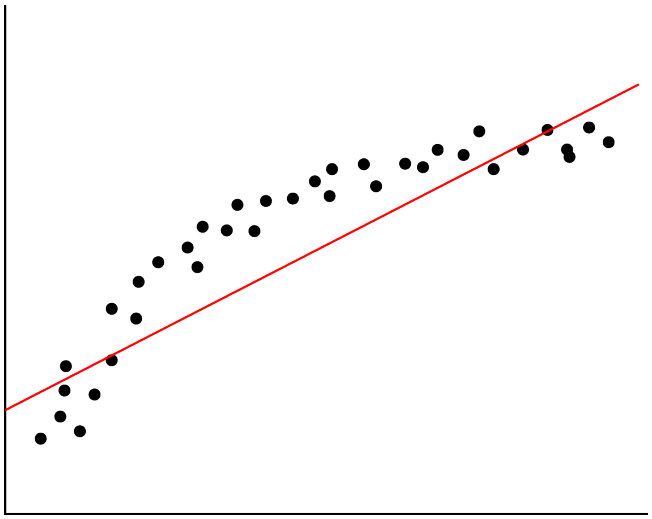
بطور طبیعی داده‌های اندازه گیری شده هیچگاه بدون نویز نمی‌باشند. زیرا هیچگاه وسایل به کار گرفته شده برای اندازه گیری داده‌ها ایده‌آل نمی‌باشد و همین طور ممکن است که عوامل محیطی بر روی داده‌ها در یک بازه تاثیر گذاشته باشند. در نتیجه، ممکن است نویز موجود در داده‌ها باعث خطا در الگوریتم یادگیری شود. به این دلیل در بسیاری از مورد توصیه می‌شود که در ابتدا نویز را از روی داده‌ها حذف کنید. یکی از راه‌های حذف نویز استفاده از فیلترها می‌باشد.

۲.۲ under-fitting

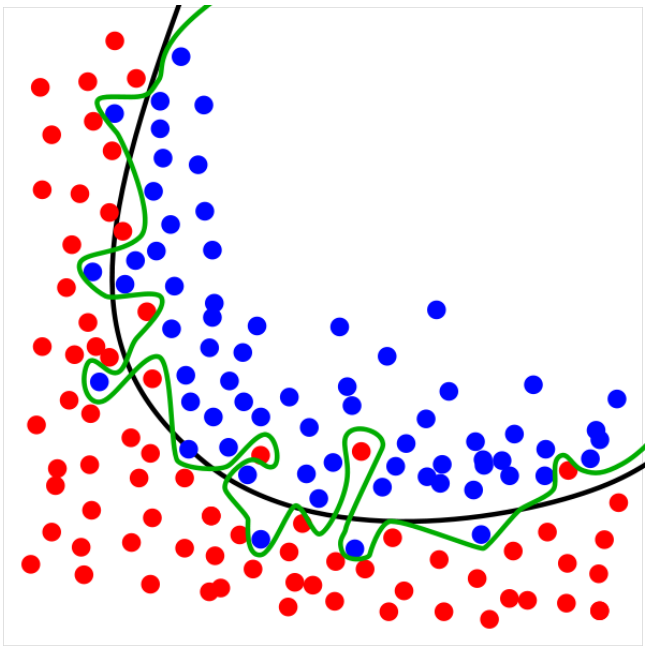
در بعضی از موارد، مدلی که learn شده است بسیار نادقیق و نیز ساده می‌باشد. در این مواقع می‌گویند که under-fitting رخ داده است. یعنی مدل learn شده یک تابع بسیار ساده می‌باشد. به عنوان مثال در شکل ۱ یک سری داده نمایش داده شده است که توسط یک تابع ساده تخمین زده شده‌اند. هنگامی که این اتفاق رخ می‌دهد خطا بر روی داده train و نیز test به شدت بالا است. برای جلوگیری از این اتفاق می‌توان از الگوریتم های یادگیری با قابلیت نمایش توابع پیچیده تر استفاده کرد.

۳.۲ over-fitting

در موارد دیگری داده به اندازی کافی در اختیار داریم اما ما می‌خواهیم که خطا بر روی داده‌های train را کم کنیم. لذا مدل تخمین زده را پیچیده‌تر می‌کنیم (می‌توان اینگونه مثال زد که درجه تابع را بالا می‌بریم تا همه نقاط در داده‌های train به صورت دقیق label بخورند.) به این حالت over-fitting می‌گویند.
انجام این‌کار باعث می‌شود که خطا بر روی داده‌های train بشدت کم شود، اما خطا بر روی داه‌های validation و نیز test بالا است. به عنوان مثال در شکل ۲ یک سری داده نمایش داده شده است که توسط یک تابع پیچیده (منحنی سبز رنگ) تخمین زده شده است در صورتی که منحنی سیاه یک تابع بهتر با پیچیدگی کمتر را نشان می‌دهد.



شکل ۱ : under-fitting



شکل ۲ : over-fitting

برای جلوگیری از over-fitting می‌توان کارهای زیر را انجام داد:

- جمع آوری بیشتر داده
- استفاده از *data augmentation*
- استفاده از *regularization* همانند $L1$ یا $L2$
- کاهش درجه تابع مدنظر

۴.۲ عدم در دست داشتن داده مناسب

استفاده از داده نامناسب باعث می‌شود که یک prediction بد داشته باشیم، حتی اگر مدل خود را به خوبی انتخاب کرده باشیم. در بسیاری از مسائل supervised learning، label داده‌ها توسط انسان زده شده است. از طرفی می‌دانیم که کار انسان همراه با خطا می‌باشد، پس ممکن است که ما مدل را درست انتخاب کرده باشیم اما label غلط داده‌ها باعث شود که learning، fail شود.

۵.۲ استفاده از ویژگی‌های فراتر از مورد نیاز

در دنیای امروز داده به اندازه زیادی وجود دارد. در مباحث عملی هنگامی که ما می‌خواهیم یک مدل را یاد بگیریم نباید داده‌هایی که هیچ کمکی به ما نمی‌کنند (یعنی هیچ اطلاعات مفیدی در آن نیست) را به داده‌های اصلی اضافه کنیم. برای مثال یک بیمارستان قصد دارد که وزن نوزادان را قبل از تولد پیش بینی کند. ویژگی‌هایی که بیمارستان نیاز دارد عبارت است از سن مادر، وزن مادر، وزن پدر و به داده‌هایی مانند اسم مادر، اسم پدر، آدرس محل سکونت و ... نیاز ندارد. اضافه کردن این ویژگی‌ها باعث می‌شود که زمان و حافظه بیشتری مصرف کنیم و مدل را پیچیده تر کنیم در حالیکه ممکن است یادگیری به درستی انجام نشود. دلیل اصلی چنین اشتباهی فهم نادرست مسئله می‌باشد.

۳ Dimension Reduction

همانگونه که در قسمت قبل بیان شد، در مواردی ما تعداد زیادی ویژگی داریم. اما از بین این ویژگی‌ها، برخی از آنها ویژگی‌های اصلی می‌باشند. پس این ایده به ذهن می‌رسد که آن‌هایی که اهمیت کمتری دارند را حذف کنیم. به این کار feature selection می‌گویند. دربرخی از مسائل ما می‌توانیم تشخیص بدهیم که کدام ویژگی اهمیت بیشتری دارد و آن را نگه داریم. در مثال پیش بینی وزن نوزاد می‌دانیم که سن مادر و وزن مادر دارای اهمیت بیشتری است پس می‌توانیم این‌ها را نگه داریم و وزن پدر را حذف کنیم.
از طرفی دیگر این ایده نیز به ذهن خطور می کند که یک سری ویژگی جدید که ترکیبی از ویژگی های اولیه می‌باشد را بدست آوریم. به این کار feature extraction می‌گویند. از دیگر دلایل کاهش dimension می‌توان به کاهش زمان، کاهش حافظه و نیز ساده‌تر بودن مدل اشاره کرد. اما باید در نظر داشت که dimension به اندازه‌ای کاهش نیابد که مدل بسیار ساده شود.
برای کاهش بعد چند روش بسیار معروف وجود دارد که شما باید هرکدام از این روش‌ها را بخوانید و در گزارش خود هر روش را حداکثر در دو پاراگراف توضیح دهید.

- PCA
- LDA
- GDA

۴ Multi-class Classification

در این بخش به این موضوع می‌پردازیم که چگونه یک classifier که بین دو کلاس طبقه بندی انجام می‌دهد به چند کلاس تعمیم بدهیم. شما می بایست دو روش زیر را مطالعه کنید و در گزارش خود هر روش را حداکثر در دو پاراگراف توضیح دهید.

- One against all classification
- One against one classification

روش اول را با یک مثال توضیح کوتاهی می‌دهیم. در نظر بگیرید که سه کلاس صندلی، میز و نیز گوشی تلفن همراه را دارید. حالا شما می‌خواهید با استفاده از Multi-class Classification یک سری عکس که مربوط به این کلاس‌ها می‌باشد را کلاس بندی کنید. به عنوان مثال در نظر بگیرید که می‌خواهید عکس‌های گوشی را تشخیص بدهید. روش اول اینگونه است که شما دو کلاس در نظر می‌گیرید گوشی و هر چیز غیر از گوشی. حال هر عکس را تشخیص می‌دهد که آیا گوشی می‌باشد یا خیر.

۵ مسئله

هدف ما در این تمرین استفاده از روش های مختلفی است که در بخش های قبل در مورد آن صحبت شد تا فرآیند یادگیری بهبود یابد. برای این کار دیتاست معروف به نام MNIST در نظر گرفته شده است. این دیتاست، مجموعه‌ای است شامل عکس‌های اعداد صفر تا نه انگلیسی که توسط انسان های مختلف با دست خط‌های مختلف نوشته شده است. مدل نهایی شما باید قادر باشد، در حد امکان این عکس ها را بدرستی کلاس بندی کند.
در مرحله آموزش از داده‌های پوشه train استفاده کنید (در نظر داشته باشید که داده‌های این پوشه را باید به دو قسمت train و نیز validation تقسیم کنید). به این داده ها نویز اضافه شده است.[۱] برای این که خطای کمتری در یادگیری داشته باشید، این نویز را حذف کنید.
ویژگی‌هایی که برای یادگیری این داده‌ها نیاز دارید بسیار کمتر از ویژگی‌هایی است که در اختیار دارید. پس روش مناسبی را برای کاهش بعد این داده‌ها انتخاب کرده و در نهایت، به بهترین عملکرد ممکن برای classifier های SVM، K-NN، Random Forest که پیاده سازی می‌کنید، برسید.
(توجه کنید که برای پیاده سازی classifier هایی همچون SVM که در حالت کلی بین دو کلاس عملیات کلاس بندی را انجام می‌دهند، باید از تکنیک‌های multi-class classifier استفاده نمایید.) در گزارش خود برای هرکدام از classifier ها، نحوه پیاده سازی و موارد زیر را گزارش بدهید:

- True Positive, True Negative, False Positive, True Negative
- Accuracy, Precision, Recall, Specificity, F1 Score
- ROC curve, Confusion Matrix

در نهایت، به نکات زیر توجه داشته باشید:

۱. در این تمرین نه تنها پیاده سازی شما مهم است، بلکه انتخاب روش های حذف نویز، کاهش بعد و multi-class classifier مناسب نیز دارای اهمیت است. بعنوان مثال در کاهش بعد، باید بعد نهایی بصورتی انتخاب شود که علاوه بر ساده‌تر شدن مدل نهایی، کمترین میزان اطلاعات از دست رفته را نیز داشته باشد.

۲. در نظر داشته باشید که در پیاده سازی الگوریتم های یادگیری شما مجاز می‌باشید که از کتابخانه scikit-learn استفاده کنید.

۳. در انتها شما باید مدل آموزش دیده خود را ذخیره کنید. مدل شما توسط داده‌ای که در اختیار ندارید تست خواهد شد و دقت شما گزارش می شود.

[۱] برای حذف نویز راه‌های بسیاری وجود دارد که در این لینک چکیده‌ای در این مورد بیان شده است.