Assignment NO.4 Solutions

Deep learning | winter 1400 | Dr.Mohammadi

Teacher Assistant:

Mohammad Hosein khojaste

Student name : **Amin Fathi**

Student id : **400722102**

Problem 1

در این مقاله روش RandAugment توضیح داده می شود ، روشی که بر خلاف روش های داده افزایی معمول که نیاز به یک مرحله جست و جوی جداگانه در مدل داشتند و مدل را پیچیده تر و بار محاسباتی داشتند است ، روش های داده افزایی آموخته شده (معمول) حتی تا بیش از ۳۰ پارامتر هم داشتند که نگارندگان در تلاش هستند که این ۳۰ پارامتر را به ۲ پارامتر تقلیل بدهند .در واقع در این حالت افزایش داده ما یک هایپرپارامتر N داریم و یک هایپرپارامترM که به وسیله هایپرپارامترN ، N تبدیل به صورت رندم از مجموعه TRANSFORMS که مجموعه شامل ۱۴ تغییرو تبدیل روی عکس (مثلا : اکوالایز کردن ، روشن تر کردن ، تیز تر کردن عکس ، چرخش و ...) که در شکل زیر آورده شده است، انتخاب میکنیم

```
transforms = [
'Identity', 'AutoContrast', 'Equalize',
'Rotate', 'Solarize', 'Color', 'Posterize',
'Contrast', 'Brightness', 'Sharpness',
'ShearX', 'ShearY', 'TranslateX', 'TranslateY']
```

و به وسیله هاپرپارامتر M هم بزرگی تبدیل اجرا شده بر روی عکس را تعیین میکنیم. که مقدار آن عدد صحیحی ما بین ۰ تا ۱۰ است (۱۰ بیشترین مقدار بزرگی است)

به طور مثال مشخصا از آنجا که ۱۴ تبدیل داریم ، با انتخاب N=2 ، میتوان ۱۴*۱۴ عکس با سیاست های تولید عکس متفات به ازای هر M منحصر به فرد ایجاد کرد (در این مثال در هر داده افزایی دو بار باید تبدیل انتخاب کرد و در هر بار تبدیل میشود ۱۴ حالت مختلف داشت پس در کل ۱۴*۱۴ حالت متفاوت می توان داشت) مقادیر بهینه M N را میتوان با استاندارد های بهینه سازی مناسب به دست آورد .

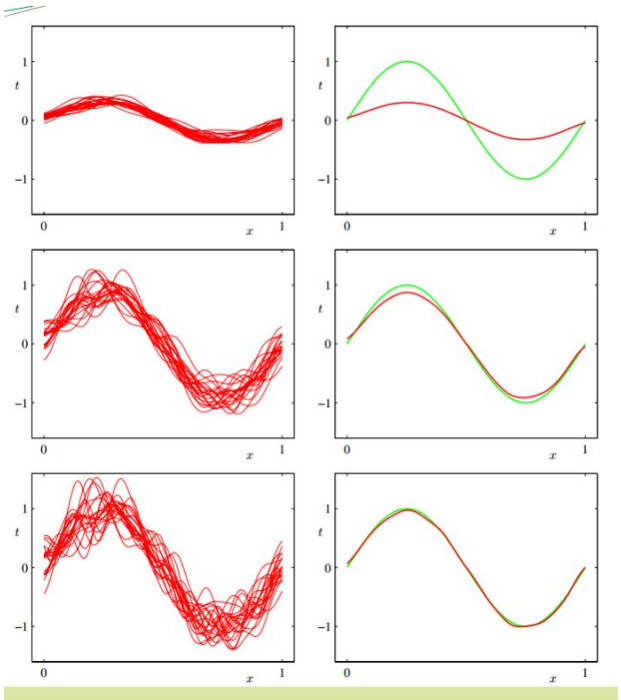
گفتنی است که نگارندگان برای به دست اوردن M چهار حالت مختلف را تست کردند یک بار مقدار ثابت برای همه اننتقال ها که مقدار اعوجاج برابر برای همه انتقال ها را به همراه داشت ، یک بار به صورت تصادفی انتخاب کردند برای هر انتقال و یک بار به صورت خطی افزایش دادند مقدار M را و یک بار هم تصادفی ولی به صورتی افزایشی با دارا بودن کران بالا و در نهایت مشاهده کردند که هر ۴ حالت تقریبا عملکرد مشابه و بالایی داشتند (مقدار تصادفی کمی بهتر بود) ؛ بنابراین، اندازه ثابت را انتخاب کردند تا تنها فقط یک ابرپارامتر را تعیین کنند (Nو بقیه‌ی آزمایش‌ها را با این استراتژی پیش بردند که در نهایت هم نتایج نسبت به داده افزایی معمولی بهتر بود .

Problem2

- Bias-Variance tradeoff را با رسم شکل توضیح دهید.
- بایاس زیاد (high bias) چطور قابل تشخیص است و برای مقابله با آن چه راه حل‌هایی وجود دارد؟
- واریانس زیاد (high variance) چطور قابل تشخیص است و برای مقابله با آن چه راه حل‌هایی وجود دارد؟

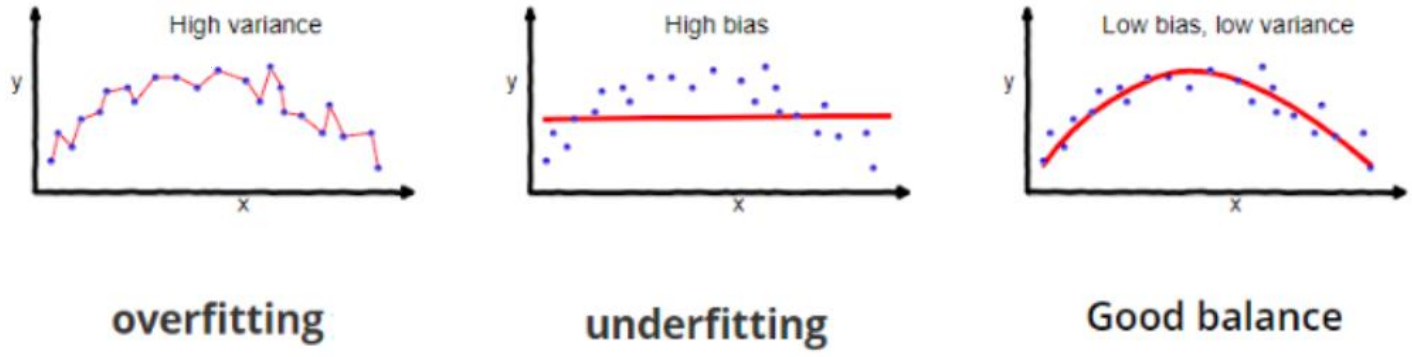
پاسخ

خطای بایاس خطایی است که از مفروضات اشتباه در الگوریتم یادگیری ناشی می شود و هر چه بایاس کم باشد یعنی مدل ما به طور میانگین دارد درست عمل می کند اما خطای واریانس خطایی است که به دلیل حساسیت بالا نسبت به تغییرات کوچک در مجموعه داده ایجاد می شود ، مجموع خطای بایاس و واریانس می شود خطای ما و در واقع هدف ما این است که هر دو خطا را کم کنیم ، مثالی که استاد در جلسه درس زدند ، در مورد مجموعه داده ای که یک خط نمیتوانست به درستی طبقه بندی کند و خطای بایاس بالایی داشت اما اضافه شدن یک داده جدید چندان خطای واریانس زیادی ایجاد نمیکرد (بر خلاف مدل پیچیده با مرز تصمیم زیگزاگی که مرز تصمیم به اضافه و کم کردن داده های جدید حساس تر است) به این معنا که الگوریتم های با پیچیدگی پایین و ساده خطای بایاس زیاد و واریانس کمی دارند (بالعکس برای الگوریتم های پیچیده) و بنابراین لازم است که توازن و trade off ای بین این دو برقرار شود که به این bias – variance tradeoff می گویند .

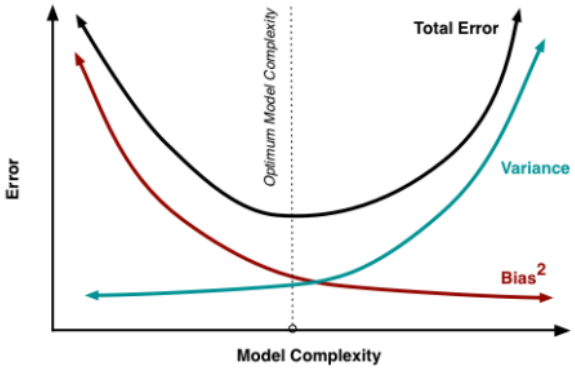


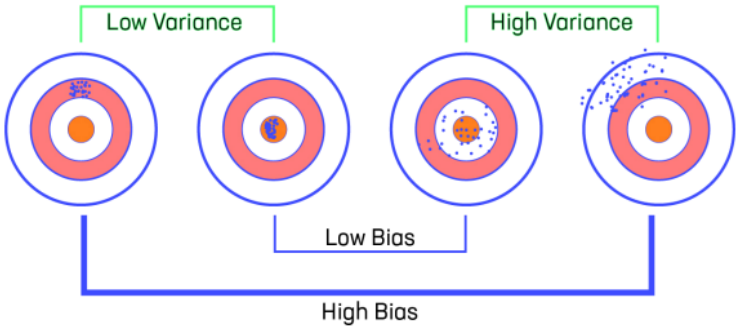
مثلا در تصویر دو تایی اول ،که سمت چپ نماینده ۱۰۰ مدل است ، مشاهده میشود که در شکل راست مدل ما(سبز رنگ) نسبت به میانگینآن ۱۰۰ تا (قرمز رنگ) تفاوت زیادی دارد پس بایاس زیاد است ، اما واریانس کم است (قرمز ها تقریبا یک شکلند)

در شکل دوم مشاهده میشود واریانس متوسط است (قرمز ها کمی تفاوت دارند) وبایاس در عوض کمی کم شده و مدل ما (سبز) به میانگین (قرمز در شکل سمت راستی) نزدیک تر است و اما در مجموعه سوم هم بایاس خیلی کم است اما واریانس (تفاوت قرمز ها درشکل سمت چپ) بسیار زیاد است .



overfitting underfitting Good balance





خط وسط در نمودار سمت چپ ، خط بهینه خط وسط است که در آن سطح از پیچیدگی مدل ، هم واریانس نسبتا خطای کمی دارد و هم خطای بایاس نسبتا کم است .

مشاهده میشود با پیچیده تر شدن مدل بایاس کم اما واریانس افزایش میابد و بالعکس.

high variance برای حالتی است که خطای مدل ما بر روی داده آموزش بسیار کم باشد و در عین حال خطای ما بر روی داده تست بسیار بالا باشد یا به اصطلاح overfit شده باشیم . و چنانچه که هم خطای مدل ما بر روی داده آموزشی زیاد باشد و هم خطا بر روی داده تست ، میگوییم under fit شده ایم یا مدل ما از high bias رنج می برد .

برای رفع high variance از آنجا که مشکل مربوط به این است که پارامتر های مدل ما به گونه ای تنظیم شدند که دقیقا خروجی مناسب ورودی را تشخیص بدهند بنابراین یکی از راه های رفع overfit شدن ، تنظیم مجدد پارامتر های مدل (کاهش مقدارشان) باعث رفع overfit شدن می شود (رگولاریزاسیون) و یا هم میتوان ویژگی های بیشتری را به مدل ارایه داد که بتواند generalization داشته باشد ، متنوع کردن تیپ داده ورودی هم کارساز است.

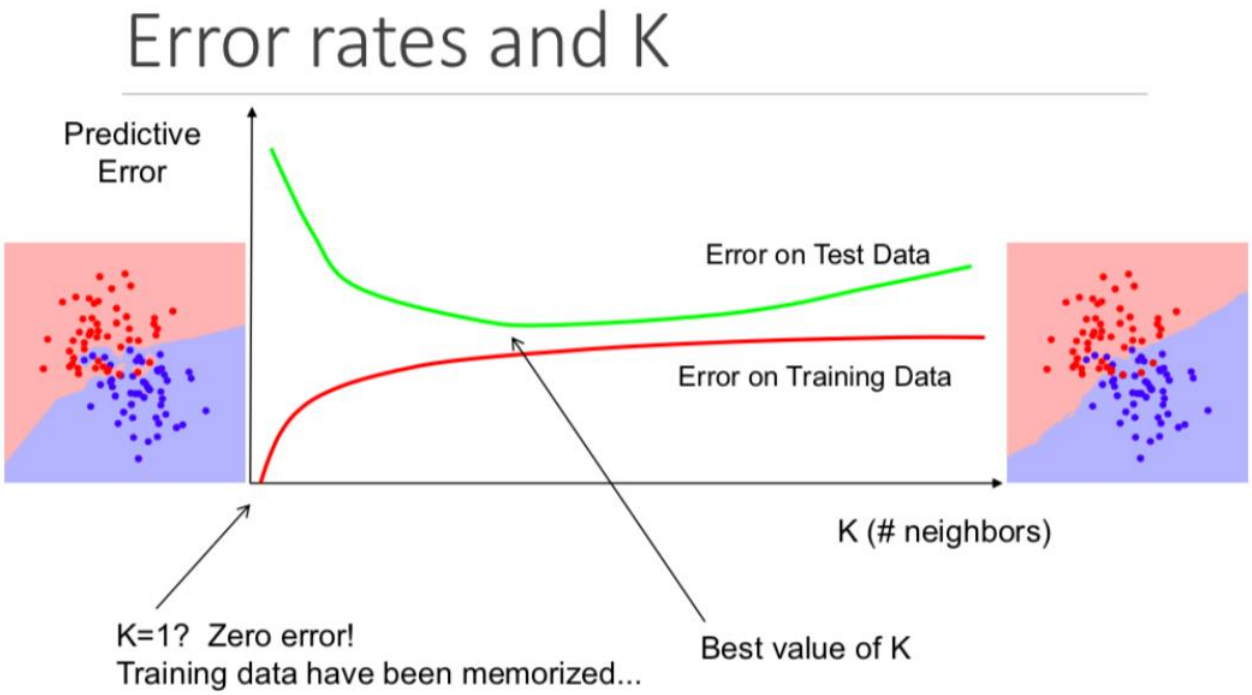
برای رفع high bias ، میتوانیم مدلی پیچیده تر را انتخاب کنیم که عملکرد بهتری با فیچر های موجود داشته باشد ، یکی دیگر از کار های ممکن ، کاهش پارامتر های رگولاریزاسیون است که در بالا هم توضیح داده شد.

- یکی از الگوریتم‌هایی که در حوزه یادگیری ماشین مورد استفاده قرار می‌گیرد، الگوریتم نزدیک‌ترین همسایگی (KNN) است. برای مطالعه بیشتر درباره این الگوریتم می‌توانید به این لینک مراجعه کنید. توضیح دهید که با تغییر مقدار K، بایاس و واریانس چه تغییری می‌کنند.

پاسخ

چنانچه k مقدار کمی باشد فرض کنید k برابر ۱ باشد ، بنابراین مدل داده آموزشی ما را به طور کامل یاد خواهد گرفت و این احتمال overfit شدن را بالا می برد که یعنی بایاس کم و واریانس زیاد (high variance) ، چنانچه هم مقدار kبسیار بالا باشد ، در واقع در داده آموزشی با خطا رو به رو هستیم (بایاس زیاد) و بعد از مدتی کاهش در خطای تست (کم شدن واریانس) بعد از حدی از k دوباره خطای تست افزایش میابد (واریانس زیاد میشود)

شکل زیر به خوبی این قضیه را روشن می سازد



- الگوریتم SVM یکی دیگر از الگوریتم‌های پرکاربرد حوزه یادگیری ماشین می‌باشد. توضیح دهید که با تغییر پارامتر C، مقدار بایاس و واریانس چه تغییری می‌کنند. برای مطالعه بیشتر می‌توانید به این لینک مراجعه کنید.

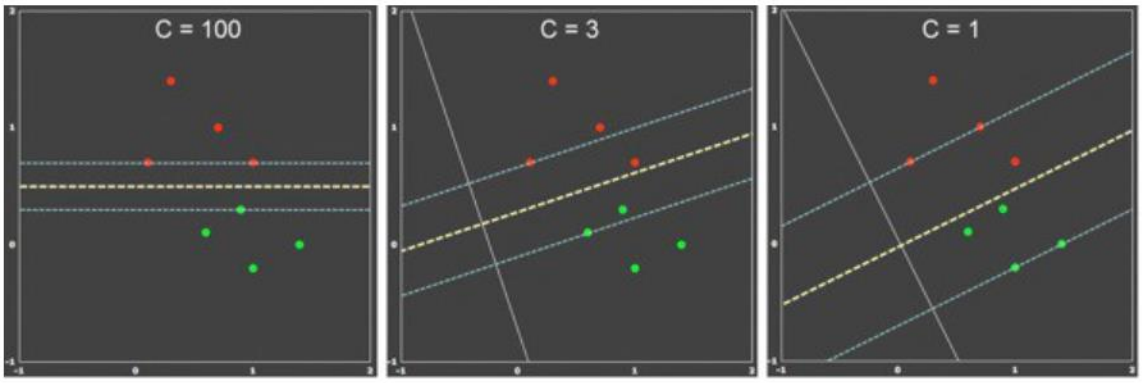
پاسخ

الگوریتم ماشین بردار پشتیبان دارای بایاس کم و واریانس بالا است، اما trade off را می توان با افزایش پارامتر C تغییر داد چرا که افزایش c باعث کم شدن margin می شود و این یعنی بایاس بیشتر و در عین حال واریانس را کم می کند .

Large Value of parameter C => small margin

Small Value of paramerter C => Large margin

Diagram below will give what exactly I am trying to say



Change in margin with change in C

- منظم سازی پارامتر L1 و L2 را مقایسه کنید.

پاسخ

همانطور که گفتیم برای آنکه شبکه OVERFIT نشود لازم است تا پارامتر های شبکه هم بهینه شودند و کاهش یابند مقدارشان ، برای این منظور از رگولاریزاسیون استفاده می شود ، دو روش مهم در رگولاریزاسیون L1 L2 را توضیح می دهیم :

L2 یا همان Ridge regression

در این حالت علاوه بر اینکه دربهینه سازی مدل ، تابع ضرر خروجی را در نظر میگیریم ، تابع ضرری هم برای بهینه سازی پارامتر ها در نظر میگیریم که مقدار آن جمع مجذور وزن ها ضرب در یک ضریب است (چنانچه ضریب صفر باشد باز همان حالت بدون در نظر گرفت رگولاریزاسیون است و اگر لاندا خیلی زیاد باشد ، منجر به undetfit شدن مدل خواهد شد)

$$\tilde{L}(\mathbf{w}, b) = L(\mathbf{w}, b) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

چنانچه از این تابع مشتق بگیریم و قصد آپدیت کردن وزن ها را داشته باشیم به معادله زیر میرسیم

$$\mathbf{w} \leftarrow (1 - \eta\lambda)\mathbf{w} - \eta\nabla_{\mathbf{w}} L(\mathbf{w}, b)$$

که همانطور که میبینید در قبل از انجام به روزرسانی معمولی ، مبتنی بر گرادیان، بردار وزن را در هر گام با یک ضریب ثابت کاهش می دهد و وزن هایی که تغییر کمتری در تابع ضرر ایجاد می کنند، اهمیت کمتری دارند و بیشتر کاهش می یابند.

در منظم سازی L1 یا همان Lasso Regression به جای مجموع نرم دو وزن ها از مجموع قدر ملق وزن ها برای بهینه سازی آن ها استفاده میکنیم ، با توجه به حضور قدر مطلق در این صورت مشتق این عبارت بر روی وزن ها برابر با علامت sgn خواهد بود و اندازه w مهم نیست دیگر و فقط علامت آن مهم است به این صورت که اگر w مثبت است سعی میکند مقدار آن را کاهش دهد و اگر مقدار آن منفیی بود سعی میکند آن را افزایش دهد و بنابراین ممکن است که وزن های کوچکتر صفر شوند .

$$\Omega(\mathbf{w}) = \|\mathbf{w}\|_1 = \sum_i |w_i|$$

$$\tilde{L}(\mathbf{w}, b) = L(\mathbf{w}, b) + \lambda\|\boldsymbol{w}\|_1$$

$$\nabla_{\mathbf{w}}\tilde{L}(\mathbf{w}, b) = \nabla_{\mathbf{w}} L(\mathbf{w}, b) + \lambda\ \mathrm{sign}(\mathbf{w})$$

وقتی از l1 استفاده میکنیم در واقع تلاشمان این است که ضرایب بیشتری را صفر کنیم و این در صورتی که ما قصد حذف تعدادبیشتری ویژگی را داشته باشیم به درد خواهد خورد ، l2 اما قصدش این است که همه مقادیر کوچک باشند به جای اینکه چند تاییشان صفر مطلق باشد

- درست یا غلط بودن گزاره‌های زیر را مشخص کنید و دلیل پاسخ خود را نیز بیان کنید.
  - استفاده از منظم سازی، ممکن است باعث تضعیف عملکرد مدل شود.
  - اضافه کردن تعداد زیاد فیچرهای جدید، باعث جلوگیری از بیش برازش می‌شود.
  - با زیاد کردن ضریب منظم‌سازی، احتمال بیش‌برازش بیشتر می‌شود.

پاسخ

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

بله اگر مقدار لاندای انتخاب شده کوچک باشد عملا باز تاثیر بهینه سازی پارامتر ها در تابع خطا کم است و overfit می شویم و با وزن بیشتر دادن به بهینه سازی پارامتر ها هم underfit می شویم .

اضافه کردن تعداد زیاد فیچرهای جدید، باعث جلوگیری از بیش برازش میشود

خیر ، افزایش بیش از حد فیچر باعث گسترده شدن وپراکندگی دادگان ما میشود که مدل هم برای یادگیری آن ها پیچیده تر خواهد شد و این باعث overfitting خواهد شد

با زیاد کردن ضریب منظم سازی، احتمال بیشبرازش بیشتر میشود

خیر ، زیاد کردن ضریب ، وزن بهینه سازی پارامتر ها را بیشتر میکند که این باعث جلوگیری از overfitting می شود هر چند این مقوله میتواند منجر به under fitting شود

- فرض کنید یک مدل داریم که یکبار بدون منظم‌سازی و یکبار با منظم‌سازی آموزش داده می‌شود. کدام یک از دو مجموعه ضرایب زیر مربوط به منظم سازی و کدام یک بدون منظم سازی است؟ دلیل انتخاب خود را توضیح دهید. همچنین توضیح دهید که به نظر شما ضرایبی که با استفاده از منظم‌سازی به دست آمده است، مربوط به منظم-سازی پارامتر L1 است یا L2؟
  - 13.3, 23.5, 53.2, 5.1
  - 0.5, 1.2, 8.5, 0

پاسخ

پایینی مربوط به منظم سازی است چون در منظم سازی ضرایب کمتر میشوند و مربوط به منظم سازی l1 است چون در این منظم سازی ضرایب صفر هم میشوند برعکس l2 که در آن کم میشوند و به نزدیکی صفر میرسند اما صفر نمیشوند.

منابع :

اسلاید های استاد

https://stackoverflow.com/questions/37776333/why-too-many-features-cause-over-fitting

Day 3 — K-Nearest Neighbors and Bias–Variance Tradeoff | by Tzu-Chi Lin | 30 days of Machine Learning | Medium

https://towardsdatascience.com/contents-9b2e49f49fe9

Gentle Introduction to the Bias-Variance Trade-Off in Machine Learning (machinelearningmastery.com)