به نام خدا

تمرین سری اول
درس یادگیری ماشین
دکتر علی شریفی زارچی

فرزان رحمانی
۴۰۳۲۱۰۷۲۵

سوال اول

الف) تابع softmax معمولاً برای مسائل طبقه‌بندی، به‌ویژه طبقه‌بندی چند کلاسه، به چند دلیل استفاده می‌شود:

۱. تفسیر احتمال

تابع softmax خروجی خام (logits) یک مدل را به احتمالات تبدیل می کند. این کار را با تبدیل مقادیر خروجی به محدوده ای بین ۰ و ۱ انجام می دهد و اطمینان حاصل می کند که مجموع این مقادیر ۱ است و آنها را به عنوان احتمال قابل تفسیر می کند. برای یک مدل خروجی n کلاس، تابع softmax احتمال تعلق ورودی به هر کلاس را محاسبه می کند:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$$

این به مدل اجازه می دهد تا عدم قطعیت را با اختصاص احتمالات مختلف به هر کلاس بیان کند.

۲. طبقه بندی چند کلاسه

Softmax به ویژه برای کارهای طبقه بندی چند کلاسه که هدف آن اختصاص یکی از چندین کلاس ممکن به یک ورودی است مفید است. برای مثال، اگر می‌خواهید یک تصویر را به یکی از ۱۰ دسته طبقه‌بندی کنید (مانند ارقام ۰ تا ۹)، softmax با مقایسه احتمالات نسبی برای هر کلاس، راهی برای تعیین محتمل‌ترین کلاس ارائه می‌کند.

۳. سازگاری با تابع ضرر آنتروپی متقابل (Cross-Entropy Loss)

هنگامی که در ترکیب با تابع ضرر آنتروپی متقابل استفاده می شود، softmax به خوبی کار می کند زیرا آنتروپی متقابل فاصله بین توزیع احتمال پیش بینی شده (از softmax) و توزیع واقعی (برچسب های کدگذاری شده) را اندازه می گیرد. خروجی softmax توزیع احتمالی را می دهد که می تواند با برچسب واقعی مقایسه شود و به حداقل رساندن آنتروپی متقابل مدل را تشویق می کند تا کلاس صحیح را با اطمینان بالاتر پیش بینی کند.

۴. مشتق پذیری

تابع softmax قابل مشتق گرفتن است، به این معنی که می توان از آن در فرآیند پس انتشار برای به روز رسانی پارامترهای مدل استفاده کرد. گرادیان های صاف آن باعث بهینه سازی کارآمد مدل در طول آموزش می شود و به gradient-based optimization ها کمک می‌کند.

۵. Score های normal

تابع softmax خروجی مدل را normal می‌کند تا امتیازهای بزرگ بر پیش‌بینی تسلط نداشته باشند، این مهم زمانی است که score خام (logits) می‌توانند دلخواه یا نامحدود باشند. بدون normal سازی، نمرات خام ممکن است خیلی شدید یا ناهموار باشد که می تواند مانع یادگیری شود.

ب)

- **Variance**: Sensitivity of the model to training data

$$\text{Variance}(x) = \mathbb{E}[(h_w(x) - \mathbb{E}[h_w(x)])^2]$$

Explanation: Complex models tend to overfit

$$h_w(x) = w_0 + w_1 x + w_2 x^2 + \cdots + w_m x^m$$

- Variance dominates when the model is too complex

$$\text{Variance} \gg \text{Bias}$$

- Fits noise, leading to high test error

در یادگیری ماشین، واریانس بالا به مدلی اطلاق می شود که بیش از حد داده های آموزشی را برازش می کند. (بر داده های آموزشی overfit می‌شود) برازش بیش از حد(overfitting) زمانی اتفاق می‌افتد که یک مدل نه تنها الگوهای اساسی در داده‌ها، بلکه نویز یا جزئیات نامربوط را نیز یاد می‌گیرد. در نتیجه، مدل بر روی داده های آموزشی به خوبی عمل می کند اما در داده های دیده نشده (test) ضعیف عمل می کند.

شاخص های واریانس بالا

- مدل دارای خطای بسیار کم در مجموعه آموزشی است اما خطای بالایی در اعتبارسنجی یا مجموعه تست دارد.
- مدل به تغییرات کوچک در داده های آموزشی بسیار حساس است، به این معنی که به خوبی تعمیم نمی دهد.

علل واریانس بالا (overfitting)

- مدل نسبت به مقدار داده های آموزشی بسیار پیچیده است (به عنوان مثال، استفاده از یک شبکه عصبی عمیق پیچیده برای یک مجموعه داده کوچک).
- مدل دارای پارامترهای بسیار زیادی است که به آن امکان می دهد به نویز موجود در داده های آموزشی fit شود.
- فقدان تکنیک های منظم سازی (L1, L2) برای محدود کردن یادگیری مدل.

راه هایی برای کاهش واریانس در مدل

۱. افزایش مقدار داده های آموزشی
   - با ارائه داده‌های بیشتر، مدل نمونه‌های بیشتری برای یادگیری دارد و احتمال برازش بیش از حد نویز را کاهش می‌دهد.
   - واریانس بالا اغلب زمانی اتفاق می‌افتد که مدل به دلیل داده‌های ناکافی، جزئیات بسیار خاصی را یاد می‌گیرد.
۲. تکنیک های منظم سازی
   - منظم سازی L1/L2: برای محدود کردن وزن مدل، یک عبارت جریمه به تابع ضرر اضافه کنید.
     - L1 (Lasso): sparsity را تشویق می کند، برخی از وزن ها را به صفر می رساند و منجر به مدل های ساده تر می شود.
     - L2 (Ridge): وزن های بزرگ را جریمه می کند و از پیچیده شدن مدل جلوگیری می کند.
   - Drop out (برای شبکه‌های عصبی): به‌طور تصادفی نورون‌ها را در حین آموزش «drop» می‌کند و شبکه را مجبور می‌کند تا با جلوگیری از سازگاری مشترک نورون‌ها، ویژگی‌های قوی‌تری را بیاموزد.
۳. ساده تر کردن مدل

- از یک مدل کمتر پیچیده با پارامترهای کمتر استفاده کنید، مانند کاهش تعداد لایه‌ها یا نورون‌ها در یک شبکه عصبی یا انتخاب یک الگوریتم ساده‌تر مانند رگرسیون لجستیک به جای شبکه عصبی عمیق در صورت لزوم.
- هرس کردن ویژگی های غیر ضروری یا استفاده از تکنیک های کاهش ابعاد مانند PCA نیز می تواند کمک کننده باشد.

۴. اعتبار سنجی متقابل (Cross-Validation)

- از تکنیک هایی مانند اعتبار سنجی متقابل k-fold استفاده کنید تا اطمینان حاصل کنید که مدل بیش از حد به زیر مجموعه خاصی از داده ها تناسب ندارد.
- با آموزش زیرمجموعه های مختلف داده و آزمایش بر روی fold باقی مانده، می توانید نحوه تعمیم مدل در پارتیشن های مختلف داده را بررسی کنید.

۵. توقف زودهنگام (Early Stopping)

- هنگام آموزش مدل های یادگیری عمیق، از توقف زودهنگام استفاده کنید. این تکنیک عملکرد مدل را در یک مجموعه اعتبارسنجی نظارت می‌کند و پس از شروع بدتر شدن عملکرد، آموزش را متوقف می‌کند و از برازش بیش از حد مدل با داده‌های آموزشی جلوگیری می‌کند.

۶. داده افزایی (Data Augmentation)

برای داده‌های تصویر، می‌توانید از داده افزایی برای افزایش مصنوعی اندازه مجموعه داده با اعمال تبدیل‌ها (به عنوان مثال، flipping, rotating) به تصاویر آموزشی استفاده کنید. این به تعمیم بهتر مدل کمک می کند.

خلاصه

واریانس بالا به این معنی است که مدل شما بیش از حد با داده های آموزشی fit شده است و به خوبی به داده های دیده نشده تعمیم نمی دهد. برای کاهش واریانس، می توانید:

- داده های آموزشی را افزایش دهید.
- از تکنیک های منظم سازی (L1/L2، Drop out) استفاده کنید.
- مدل را ساده کنید.
- اعتبارسنجی متقابل را اعمال کنید.
- از توقف زودهنگام استفاده کنید.
- Data Augmetation را اعمال کنید.

پ) وقتی همه ویژگی ها تا حد خوبی با خروجی همبستگی داشته باشند رگرسیون Ridge بر رگرسیون Lasso ترجیح داده می شود زیرا Ridge تمایل دارد ضرایب همه ویژگی ها را به طور یکنواخت کوچک کند، در حالی که Lasso انتخاب ویژگی را با صفر کردن برخی از ضرایب انجام می دهد و به طور موثر ویژگی های کم اهمیت تر را حذف می کند. در اینجا نگاهی عمیق تر به این که چرا Ridge در این سناریو بهتر کار می کند می اندازیم:

تفاوت های کلیدی بین رگرسیون Ridge و Lasso

- رگرسیون Ridge (L2 regularization) جریمه ای متناسب با مجذور ضرایب اضافه می کند:

$$\text{Ridge Loss} = \text{MSE} + \lambda \sum_{j=1}^{p} \beta_j^2$$

این جریمه همه ضرایب را به سمت صفر کوچک می کند اما هرگز دقیقاً به صفر نمی رسد.

- رگرسیون Lasso (L1 regularization) جریمه ای متناسب با قدر مطلق ضرایب اضافه می کند:

$$\text{Lasso Loss} = \text{MSE} + \lambda \sum_{j=1}^{p} |\beta_j|$$

جریمه آن تمایل دارد برخی از ضرایب را دقیقاً به صفر کاهش دهد و انتخاب خودکار ویژگی (feature selection) را انجام دهد.

چرا Ridge در این حالت ترجیح داده می شود؟

۱. Feature Shrinkage vs. Feature Selection
   - رگرسیون Ridge ضرایب ویژگی های همبسته را به طور متناسب کوچک می کند، به این معنی که هیچ ویژگی را به طور کامل حذف نمی کند بلکه تأثیر آنها را به طور یکنواخت کاهش می دهد. هنگامی که ویژگی ها به خوبی با خروجی همبستگی دارند، احتمالاً همه آنها حاوی اطلاعات ارزشمندی هستند و حذف هر یک از آنها ممکن است منجر به از دست دادن قدرت پیش بینی شود. (Feature Shrinkage)
   - از سوی دیگر، رگرسیون Lasso می‌تواند برخی از ضرایب را دقیقاً به صفر برساند و برخی از ویژگی‌ها را به طور کامل از مدل حذف کند. اگر همه ویژگی‌ها با خروجی مرتبط باشند، حذف هر یک از آنها ممکن است اطلاعات مفید را از بین ببرد و منجر به کاهش عملکرد مدل شود. (Feature Selection)
۲. مدیریت Multicollinearity
   - رگرسیون Ridge، Multicollinearity را (زمانی که ویژگی ها با یکدیگر همبستگی دارند) بهتر از Lasso کنترل می کند زیرا وزن ضرایب را در بین ویژگی های همبسته توزیع می کند. از آنجایی که Ridge ضرایب را کوچک می کند اما آنها را صفر نمی کند، به مدل اجازه می دهد تا قدرت پیش بینی همه ویژگی های مرتبط را حفظ کند.
   - با این حال، Lasso ممکن است به طور تصادفی یک ویژگی را از گروهی از ویژگی‌های همبسته انتخاب کند و بقیه را روی صفر قرار دهد، که می‌تواند باعث بی‌ثباتی در مدل در هنگام برخورد با داده‌های بسیار همبسته شود.
۳. ثبات در برآورد ضرایب
   - Ridge تخمین های پایدارتری را در حضور همبستگی بالا بین ویژگی ها ارائه می دهد. از آنجایی که ضرایب را بدون اجبار به صفر کوچک می کند، مدل قوی باقی می ماند، حتی اگر درجه بالایی از multicollinearity وجود داشته باشد.
   - Lasso می تواند منجر به انتخاب ویژگی ناپایدار شود، به خصوص زمانی که همبستگی بین ویژگی ها بالا باشد، که بسته به اینکه کدام ویژگی حفظ یا حذف شود، منجر به نتایج متناقض می شود.

خلاصه

- Ridge زمانی ترجیح داده می‌شود که همه ویژگی‌ها به خوبی با خروجی همبستگی داشته باشند، زیرا ضرایب را بدون حذف هیچ ویژگی کوچک می‌کند و تضمین می‌کند که همه ویژگی‌های همبسته به مدل کمک می‌کنند.
- Lasso بیشتر برای انتخاب ویژگی مناسب است، و در موقعیت‌هایی با همبستگی ویژگی‌های بالا، ممکن است به‌طور خودسرانه برخی از ویژگی‌ها را حذف کند و به طور بالقوه منجر به مدلی با ثبات یا عملکرد پایین‌تر شود.

بنابراین، در مواردی که همه ویژگی‌ها آموزنده هستند و به خوبی با خروجی همبستگی دارند، رگرسیون Ridge به طور کلی انتخاب بهتری است زیرا در عین مدیریت Overfitting از طریق منظم‌سازی، تمام ویژگی‌ها را حفظ می‌کند.

ت)

$$\mathbb{E}[(y - h_w(x))^2] = (\text{Bias})^2 + \text{Variance} + \text{Noise}$$

منظم‌سازی L2 (که در رگرسیون Ridge برای مدل‌های خطی استفاده می‌شود) نقش کلیدی در مدیریت تعادل بایاس واریانس در مدل‌های یادگیری ماشین بازی می‌کند. در اینجا آمده که چگونه منظم سازی L2 بر این تعادل تأثیر می گذارد:

۱. درک Bias-Variance Trade-Off
- سوگیری (Bias) به خطای ناشی از فرضیات ساده‌سازی شده در مدل اشاره دارد که باعث می‌شود مدل underfit شود و الگوهای مهم را از دست بدهد. سوگیری زیاد(High Bias) منجر به خطای زیاد آموزشی و تعمیم ضعیف می شود.
- واریانس به حساسیت مدل نسبت به نوسانات کوچک در داده های آموزشی اشاره دارد. مدلی با واریانس بالا به داده های آموزشی overfit می شود، نویز را یاد می گیرید و منجر به تعمیم ضعیف به داده های دیده نشده می شود.

هدف یافتن تعادل مناسب است که در آن مدل الگوهای اساسی (سوگیری کم) را بدون حساسیت بیش از حد به نویز (واریانس کم) ثبت می‌کند.

۲. اثر منظم سازی L2 بر مبادله بایاس-واریانس

منظم‌سازی L2 یک عبارت جریمه را معرفی می‌کند که با مجذور ضرایب مدل متناسب است:

$$\text{Loss with L2} = \text{MSE (or classification loss)} + \lambda \sum_{j=1}^{p} \beta_j^2$$

که در آن λ قدرت منظم سازی است (یک هایپر پارامتر)، و β نشان دهنده ضرایب (وزن) مدل است.

چگونه منظم سازی L2 بر bias و واریانس تأثیر می گذارد:

- افزایش bias:
  - منظم‌سازی L2 ضرایب مدل را کوچک می‌کند، به این معنی که انعطاف‌پذیری مدل کاهش می‌یابد. با محدود کردن ضرایب مدل، مرز تصمیم گیری یا پیچیدگی مدل را ساده می کند. این امر بیش از حد برازش را کاهش می دهد، اما به قیمت ایجاد مقدار کمی سوگیری.
  - منظم‌سازی مدل را تشویق می‌کند تا به جای برازش کامل داده‌های آموزشی، بر الگوهای عمومی‌تر تکیه کند، که منجر به خطای آموزشی بالاتر و احتمالاً سوگیری کمی بالاتر می‌شود.
- کاهش واریانس:
  - با جریمه کردن ضرایب بزرگ، منظم سازی L2 از حساس شدن بیش از حد مدل به داده های آموزشی جلوگیری می کند و در نتیجه توانایی آن را برای تناسب با نویز کاهش می دهد. این باعث می شود مدل کمتر مستعد تغییرات شدید در پاسخ به تغییرات جزئی در داده های آموزشی باشد و واریانس را کاهش دهد.
  - مدلی با واریانس کمتر، بهتر به داده های جدید و دیده نشده تعمیم می دهد و عملکرد را در مجموعه های آزمایشی بهبود می بخشد.

۳. چگونه L2 Regularization تعادل مناسب را ایجاد می کند؟

- بدون منظم‌سازی، یک طبقه‌بندی‌کننده خطی ممکن است بایاس کم اما واریانس بالایی داشته باشد که منجر به بیش از حد برازش می‌شود. تمام جزئیات داده های آموزشی، از جمله نویز، را یاد می گیرد و روی داده های دیده نشده عملکرد ضعیفی دارد.
- تنظیم L2 جریمه‌ای برای وزن‌های بزرگ اضافه می‌کند که با نرم‌تر کردن و کلی‌تر کردن مدل، به کاهش بیش از حد برازش(overfitting) کمک می‌کند. با کنترل پیچیدگی مدل، سوگیری را اندکی افزایش می‌دهد، اما واریانس را به میزان قابل توجهی کاهش می‌دهد، که معمولاً منجر به تعمیم بهتر می‌شود.

با تنظیم قدرت تنظیم λ:

- λ کوچک: مدل دارای آزادی بیشتری است، که منجر به یک مرز تصمیم گیری پیچیده تر می شود، که ممکن است سوگیری را کاهش دهد اما واریانس را افزایش می دهد.
- λ بزرگ: مدل محدودتر می شود، واریانس را کاهش می دهد اما سوگیری را افزایش می دهد. مرز تصمیم مدل هموارتر و کلی تر می شود.

۴. نمایش گرافیکی

به طور کلی، با افزایش قدرت منظم سازی λ:

- سوگیری(bias) مدل به تدریج افزایش می یابد (از آنجایی که مدل انعطاف پذیری کمتری دارد).
- واریانس به شدت کاهش می یابد (از آنجایی که مدل نویز را در داده ها fit نمی کند).

منظم‌سازی بهینه تعادلی را پیدا می‌کند که در آن مجموع خطاهای بایاس و واریانس به حداقل برسد، که معمولاً منجر به تعمیم بهتر می‌شود.

خلاصه ای از اثر منظم سازی L2 بر تعادل سوگیری-واریانس:

- منظم‌سازی L2 با کاهش انعطاف‌پذیری مدل و کاهش توانایی آن در تناسب(fit) کامل با داده‌های آموزشی، سوگیری را کمی افزایش می‌دهد.
- منظم‌سازی L2 با جلوگیری از fit شدن نویز در داده‌ها و قوی‌تر کردن مدل نسبت به داده‌های دیده نشده، واریانس را کاهش می‌دهد.
- پارامتر منظم‌سازی λ قدرت جریمه را کنترل می‌کند و به مدل اجازه می‌دهد تا trade-off بایاس واریانس را هدایت کند و عملکرد تعمیم را بهبود بخشد.

با معرفی منظم سازی L2، شما اساساً افزایش اندکی در سوگیری را برای کاهش قابل توجهی در واریانس معامله می کنید، که اغلب منجر به عملکرد بهتر در داده های دیده نشده می شود.

سوال دوم

الف) در رگرسیون خطی، هدف ما به حداقل رساندن مجذور خطا بین مقادیر پیش‌بینی‌شده و مقادیر واقعی است:

$$\min_{w} || \ y - Xw \ ||^2$$

هنگامی که رگرسیون را فقط روی ویژگی j انجام می دهیم، اساساً در حال حل مسئله ساده تر زیر هستیم، که X_j به ردیف j (ویژگی) ماتریس داده X اشاره دارد و بردار پیش بینی ها عبارت است از:

$$\hat{y} = w_j X_j$$

خطای مجذور تبدیل می شود به:

$$L(w_j) = \sum_{i=1}^{N} (y_i - w_j X_{ji})^2$$

ما باید L(w_j) را با توجه به w_j کمینه کنیم. برای یافتن w_j بهینه، مشتق L(w_j) را نسبت به w_j میگیریم:

$$\frac{\partial L(w_j)}{\partial w_j} = -2 \sum_{i=1}^{N} X_{ji}(y_i - w_j X_{ji}) = -2 \sum_{i=1}^{N} (X_{ji} y_i - w_j {X_{ji}}^2)$$

ساده سازی به شکل ماتریسی:

$$\frac{\partial L(w_j)}{\partial w_j} = -2(X_j y - w_j X_j X_j^\top)$$

برای یافتن مینیمم خطا، مشتق را برابر صفر قرار دهید:

$$-2(X_j y - w_j X_j X_j^\top) = 0 \qquad --\rightarrow \qquad X_j y - w_j X_j X_j^\top = 0$$

حل برای w_j:

$$X_j y = w_j X_j X_j^\top \qquad --\rightarrow \qquad w_j = \frac{X_j y}{X_j X_j^\top}$$

بنابراین، نشان داده‌ایم که وزن w_j هنگام انجام رگرسیون فقط روی ویژگی j است:

$$w_j = \frac{X_j y}{X_j X_j^\top}$$

ب) ایده کلیدی در اینجا این است که وقتی ویژگی‌ها مستقل هستند، term های متقابل در ماتریس کوواریانس $X^\top X$ صفر می‌شوند و منجر به ساختار block-diagonal می‌شوند. این ویژگی مسئله را ساده می کند تا هر ویژگی به طور مستقل حل شود.

برای رگرسیون خطی با همه در نظر گرفتن همه ویژگی‌ها، ما عبارت زیر را به حداقل می‌رسانیم:

$$L(w) = || y - Xw ||^2$$

راه حل بهینه کلی برای رگرسیون خطی چند متغیره به شکل زیر است که در اسلاید های درس اثبات کردیم:

$$w = (X^\top X)^{-1} X^\top y$$

از آنجایی که ویژگی ها مستقل هستند، $(X^\top X)_{ij} = 0$ برای $i \neq j$. بنابراین، ماتریس کوواریانس $X^\top X$ به یک ماتریس diagonal تبدیل می شود:

$$X^\top X = \text{diag}(X_1 X_1^\top, X_2 X_2^\top, \dots, X_L X_L^\top)$$

همان طور که میدانیم معکوس آن نیز diagonal است که درایه های قطر اصلی آن به شکل زیر هستند:

$$X^\top X^{-1} = \text{diag}\left(\frac{1}{X_1 X_1^\top}, \frac{1}{X_2 X_2^\top}, \dots, \frac{1}{X_L X_L^\top}\right)$$

پس معادله بردار وزن ها $w = (X^\top X)^{-1} X^\top y$ به شکل زیر ساده می شود:

$$w = \text{diag}\left(\frac{1}{X_1 X_1^\top}, \frac{1}{X_2 X_2^\top}, \dots, \frac{1}{X_L X_L^\top}\right) X^\top y$$

بنابراین برای هر ویژگی j در بردار w داریم:

$$w_j = \frac{X_j^\top y}{X_j X_j^\top}$$

که این عبارت (پارامتر های بهینه) دقیقاً همان نتیجه ای است که هنگام انجام رگرسیون روی هر ویژگی به طور مستقل داریم، همانطور که در قسمت (الف) نشان داده شده است. از این رو، نتیجه می‌گیریم که وقتی ویژگی‌ها مستقل هستند، پارامترهای بهینه به‌دست‌آمده از آموزش رگرسیون روی همه ویژگی‌ها با پارامترهای به‌دست‌آمده از آموزش روی هر ویژگی به‌طور مستقل یکسان است.

پ)

۱۵ چهارشنبه 3
بهمن February 2021
۲۰ جمادی‌الثانیه ۱۴۴۲ Wednesday

۱۳۹۹

در این قسمت جمله عرض از مبدأ w_0 را به معادله regression اضافه می‌کنیم.

$$\hat{y} = w_j X_j + w_0$$

بنابراین معادله prediction به شکل $\hat{y} = w_j X_j + w_0$ می‌شود.

پس MSE به شکل زیری شود:

$$L(w_0, w_j) = \sum_{i=1}^{N} (y_i - w_0 - w_j X_{ji})^2$$

برای minimum کردن این عبارت مشتق آن را می‌گیریم و برابر با صفر قرار می‌دهیم.

$$\text{I)}\ \frac{\partial L}{\partial w_j} = -2 \sum_{i=1}^{N} (y_i - w_j X_{ji} - w_0) X_{ji} = 0$$

$$\text{II)}\ \frac{\partial L}{\partial w_0} = -2 \sum_{i=1}^{N} (y_i - w_j X_{ji} - w_0) = 0$$

$$\text{I)}\ \sum_{i=1}^{N} y_i X_{ji} = w_j \sum_{i=1}^{N} X_{ji}^2 + w_0 \sum_{i=1}^{N} X_{ji}$$

$$\text{II)}\ \sum_{i=1}^{N} y_i = w_j \sum_{i=1}^{N} X_{ji} + \sum_{i=1}^{N} w_0 = w_j \sum_{i=1}^{N} X_{ji} + N w_0$$

فرم بسته معادله I:

$$\text{I)}\ w_j = \frac{X_j (y - w_0)}{X_j X_j^T}$$

چ	پ	ج	ش	ی	د	س	چ	پ	ج	ش	ی	د	س	چ	پ	ج	ش	ی	د	س	چ	پ	ج	ش	ی	د	س	چ	پ
۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰	۲۱	۲۲	۲۳	۲۴	۲۵	۲۶	۲۷	۲۸	۲۹	۳۰

ولادت حضرت فاطمه زهرا سلام‌الله‌علیها (هشتم قبل از هجرت) و روز زن - تولد حضرت امام خمینی (رحمة‌الله‌علیه) رهبر کبیر انقلاب اسلامی (۱۳۲۰ هـ.ق)

II) $W_0 = \frac{\sum_{i=1}^{N} y_i - W_j \sum_{i=1}^{N} x_{ji}}{N} = \frac{\sum_{i=1}^{N} y_i}{N} - \frac{W_j \sum_{i=1}^{N} x_{ji}}{N}$

$\bar{y} = \frac{\sum_{i=1}^{N} y_i}{N}$ ، $\frac{\sum_{i=1}^{N} x_{ji}}{N} = \bar{x}_j$ ← $W_0 = \bar{y} - W_j \bar{x}_j$

در معادلهٔ I قرار می‌دهیم.

$$\sum_{i=1}^{N} y_i x_{ji} = W_j \sum_{i=1}^{N} x_{ji}^2 + (\bar{y} - W_j \bar{x}_j) \sum_{i=1}^{N} x_{ji}$$

$$\sum y_i x_{ji} = W_j \sum x_{ji}^2 + \bar{y} \sum x_{ji} - W_j \bar{x}_j \sum_{i=1}^{N} x_{ji}$$

$$\sum y_i x_{ji} - \bar{y} \sum x_{ji} = W_j \left(\sum x_{ji}^2 - \bar{x}_j \sum x_{ji} \right)$$

$$W_j = \frac{\sum_{i=1}^{N} y_i x_{ji} - \bar{y} \sum_{i=1}^{N} x_{ji}}{\sum_{i=1}^{N} x_{ji}^2 - \bar{x}_j \sum_{i=1}^{N} x_{ji}}$$

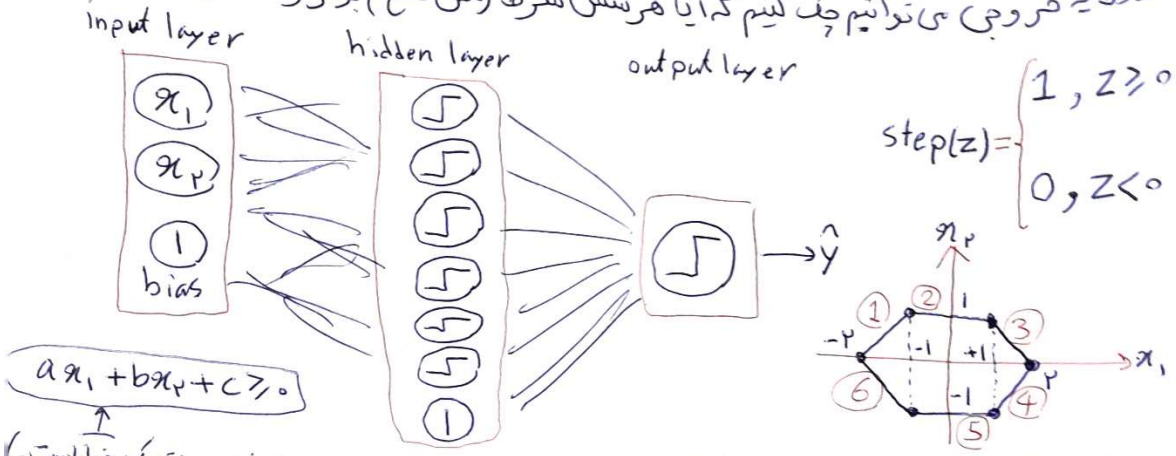
$$\bar{y} = \frac{\sum_{i=1}^{N} y_i}{N}$$

$$W_0 = \bar{y} - W_j \bar{x}_j$$

$$\bar{x}_j = \frac{\sum_{i=1}^{N} x_{ji}}{N}$$

جواب نهایی: ابتدا $\bar{y}$ و $\bar{x}_j$ را محاسبه می‌کنیم و سپس W_j را می‌یابیم. پس آن W_0 به دست می‌آید.

۳- برای حل این مسئله نیاز است تا شش ضلعی را به ۶ ضلع (linear boundary) تقسیم کنیم و سپس هرضلع با یک نورون قابل تشخیص است. در نهایت نیز با استفاده از یک نورون دولایه خروجی می توانیم چک کنیم که آیا هرشش شرط (شش ضلع) برقرار است یا نه.



حال باید معادله داخل ورودی شش ضلعی را برای هر ضلع بنویسیم. (هرضلع در واقع یک خط است.)

1 → $x_2 = x_1 + 2$ → پایین ورودی خط → $x_2 \leq x_1 + 2$ → $x_1 - x_2 + 2 \geq 0$

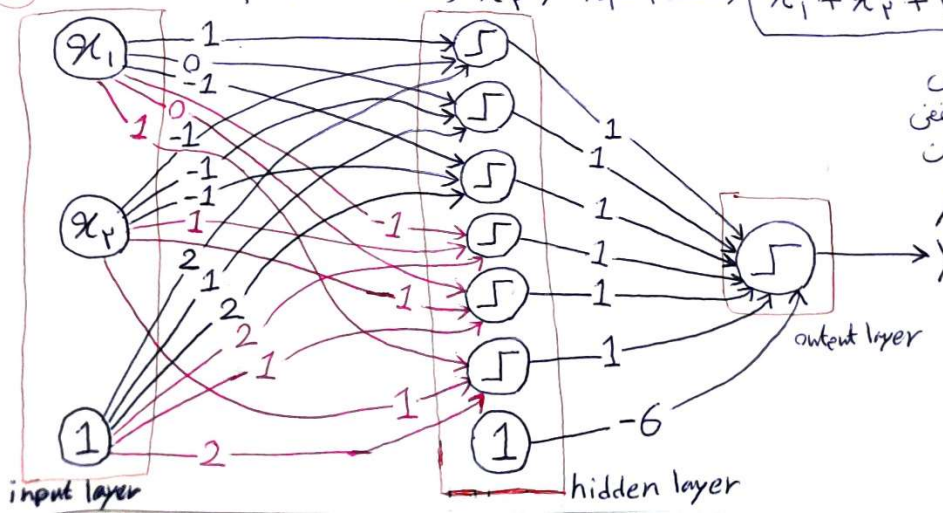
2 → $x_2 = 1$ → پایین ورودی خط → $x_2 \leq 1$ → $-x_2 + 1 \geq 0$

3 → $x_2 = -x_1 + 2$ → پایین ورودی خط → $x_2 \leq -x_1 + 2$ → $-x_1 - x_2 + 2 \geq 0$

4 → $x_2 = +x_1 - 2$ → بالا ورودی خط → $x_2 \geq x_1 - 2$ → $x_2 - x_1 + 2 \geq 0$

5 → $x_2 = -1$ → بالا ورودی خط → $x_2 \geq -1$ → $x_2 + 1 \geq 0$

6 → $x_2 = -x_1 - 2$ → بالا ورودی خط → $x_2 \geq -x_1 - 2$ → $x_1 + x_2 + 2 \geq 0$



برای نورون لایه خروجی وزن هر نورون لایه مخفی را یک می گذاریم و وزن بایاس را ۶- بنابراین فقط زمانی که هرشش شرط برقرار باشند خروجی 1 می شود.

الف) $\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$

Case 1: $k = i$ $\quad \frac{\partial \hat{y}_i}{\partial z_i} = \frac{\partial}{\partial z_i}\left(\frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}\right)$

quotient rule $\rightarrow$ $\frac{\partial \hat{y}_i}{\partial z_i} = \frac{e^{z_i}\left(\sum_{j=1}^{n} e^{z_j}\right) - e^{z_i}\cdot e^{z_i}}{\left(\sum_{j=1}^{n} e^{z_j}\right)^2} = \frac{e^{z_i}\left(\sum_{j=1}^{n} e^{z_j} - e^{z_i}\right)}{\left(\sum_{j=1}^{n} e^{z_j}\right)^2} = \frac{e^{z_i}\left(\sum_{j\neq i} e^{z_j}\right)}{\left(\sum_{j=1}^{n} e^{z_j}\right)^2}$

$$\frac{\partial \hat{y}_i}{\partial z_i} = \frac{e^{z_j}}{\sum_{j=1}^{n} e^{z_j}} \times \frac{\sum_{j\neq i} e^{z_j}}{\sum_{j=1}^{n} e^{z_j}} = \hat{y}_i \times (1 - \hat{y}_i) \rightarrow \boxed{\frac{\partial \hat{y}_i}{\partial z_i} = \hat{y}_i(1-\hat{y}_i)}$$

Case 2: $k \neq i$ $\quad \frac{\partial \hat{y}_i}{\partial z_k} = \frac{\partial}{\partial z_k}\left(\frac{e^{z_j}}{\sum_{j=1}^{n} e^{z_j}}\right) = \frac{0 - e^{z_i}e^{z_k}}{\left(\sum_{j=1}^{n} e^{z_j}\right)^2}$ (عدد ثابت)

$$\frac{\partial \hat{y}_i}{\partial z_k} = -\frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \times \frac{e^{z_k}}{\sum_{j=1}^{n} e^{z_j}} = -\hat{y}_i\,\hat{y}_k \rightarrow \boxed{\frac{\partial \hat{y}_i}{\partial z_k} = -\hat{y}_i\,\hat{y}_k}$$

خلاصه: $k = i \rightarrow \boxed{\frac{\partial \hat{y}_i}{\partial z_i} = \hat{y}_i(1-\hat{y}_i)}$, $k \neq i \rightarrow \boxed{\frac{\partial \hat{y}_i}{\partial z_k} = -\hat{y}_i\,\hat{y}_k}$

ب) $L = -\sum_{i=1}^{n} y_i \log(\hat{y}_i)$ از قانون زنجیری (chain rule) استفاده می‌کنیم.

$$\frac{\partial L}{\partial z_k} = \frac{\partial L}{\partial \hat{y}_i} \times \frac{\partial \hat{y}_i}{\partial z_k} = \frac{\partial}{\partial \hat{y}_i}\left(-\sum_{i=1}^{n} y_i \log(\hat{y}_i)\right)\left(\frac{\partial \hat{y}_i}{\partial z_k}\right) = -\sum_{i=1}^{n} \frac{\partial}{\partial \hat{y}_i}\left(y_i \log(\hat{y}_i)\right)\left(\frac{\partial \hat{y}_i}{\partial z_k}\right)$$

$$\frac{\partial L}{\partial z_k} = \left(-\sum_{i=1}^{n} \frac{y_i}{\hat{y}_i}\right)\left(\frac{\partial \hat{y}_i}{\partial z_k}\right) \xrightarrow[\text{تقسیم می‌کنیم}]{\text{با توجه به قسمت الف به دو قسمت}} \frac{\partial L}{\partial z_k} = -\frac{y_k}{\hat{y}_k} \times \hat{y}_k(1-\hat{y}_k) - \sum_{i\neq k} \frac{y_i}{\hat{y}_i} \times (-\hat{y}_i\,\hat{y}_k)$$

$$\frac{\partial L}{\partial z_k} = -y_k(1-\hat{y}_k) + \sum_{i\neq k} y_i\,\hat{y}_k = -y_k + y_k\hat{y}_k + \sum_{i\neq k} y_i\,\hat{y}_k = -y_k + \sum_{i=1}^{n} y_i\,\hat{y}_k$$

(باز می‌کنیم)

$$\frac{\partial L}{\partial z_k} = -y_k + \hat{y}_k \sum_{i=1}^{n} y_i \xrightarrow[\text{های واقعی است}]{\text{چون } y \text{ بردار برچسب}} \sum_{i=1}^{n} y_i = 1 \text{ می‌دانیم}$$

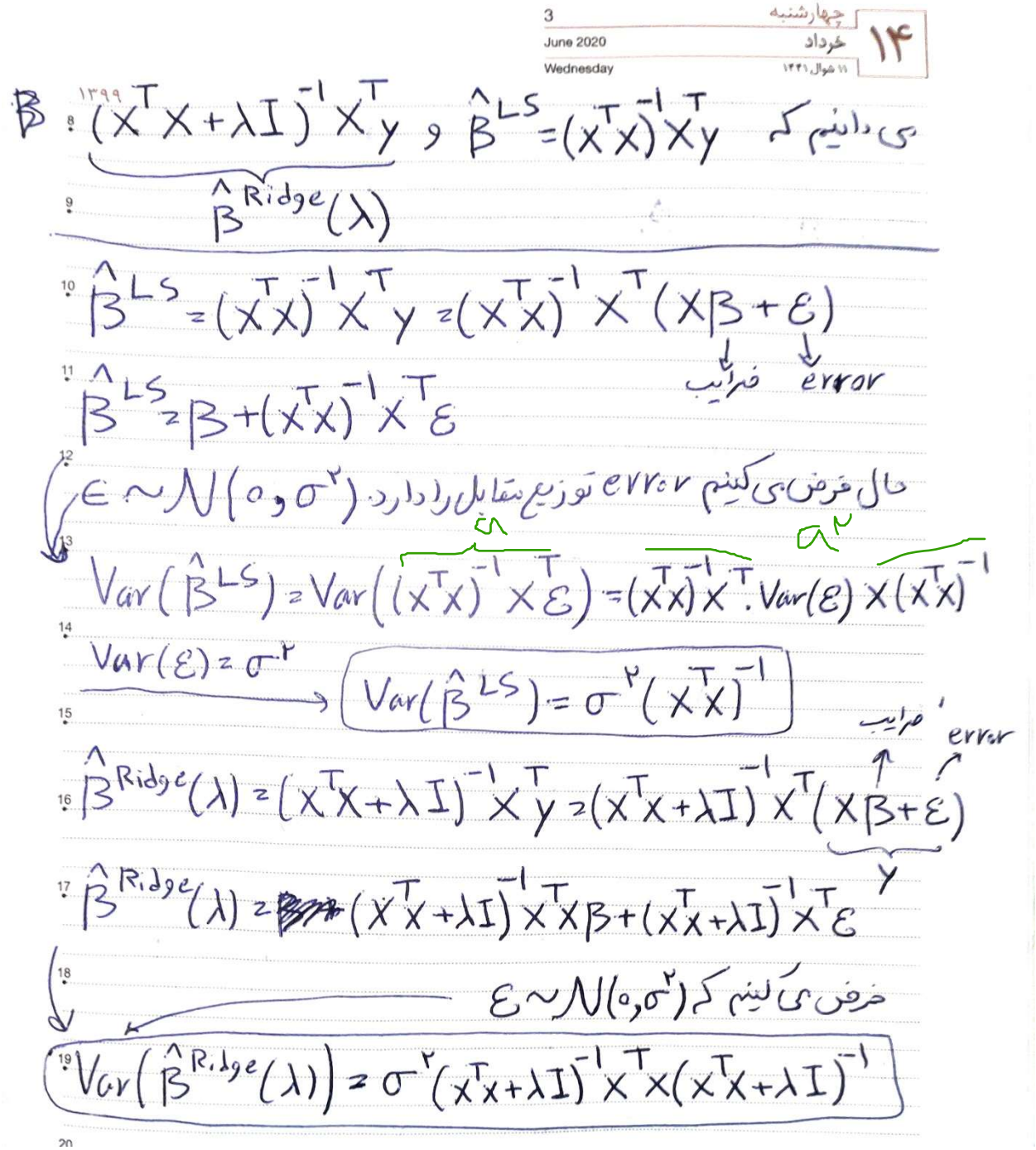
و معمولاً one hot است. ← مثلاً $y = [0,0,1,0]$

$$\boxed{\frac{\partial L}{\partial z_k} = -y_k + \hat{y}_k = \hat{y}_k - y_k}$$

این یک نتیجه بسیار ساده است. نشان می‌دهد گرادیان فقط تفاوت بین احتمال پیش‌بینی شده و برچسب واقعی است که حس شهودی دارد: ما می‌خواهیم پیش‌بینی‌هایمان را متناسب با خطا و آنها با برچسب‌های واقعی تنظیم کنیم.

سوال پنجم

الف)



می خواهیم اثبات کنیم اگر $\lambda > 0$ آنگاه $Var(\hat{\beta}^{LS}) > Var(\hat{\beta}^{Ridge}(\lambda))$

$X^TX = A \rightarrow Var(\hat{\beta}^{LS}) = \sigma^2 A^{-1}$

$Var(\hat{\beta}^{Ridge}(\lambda)) = \sigma^2 (A+\lambda I)^{-1} A (A+\lambda I)^{-1}$

فرض می کنیم تجزیه A مقدارهای ویژه به این شکل است.

$A = Q \Lambda Q^T$

Q = ماتریس متعامد بردارهای ویژه

Λ = ماتریس قطری شامل مقادیر ویژه A

$Var(\hat{\beta}^{LS}) = \sigma^2 Q \Lambda^{-1} Q^T$

$Var(\hat{\beta}^{Ridge}(\lambda)) = \sigma^2 Q (\Lambda+\lambda I)^{-1} \Lambda (\Lambda+\lambda I)^{-1} Q^T$

برای هر مقدار ویژه a_i از A واریانس به این شکل است.

$\rightarrow Var(\hat{\beta}^{LS}_{(i)}) = \frac{\sigma^2}{a_i}$

(به عناصر قطری توجه می کنیم.)

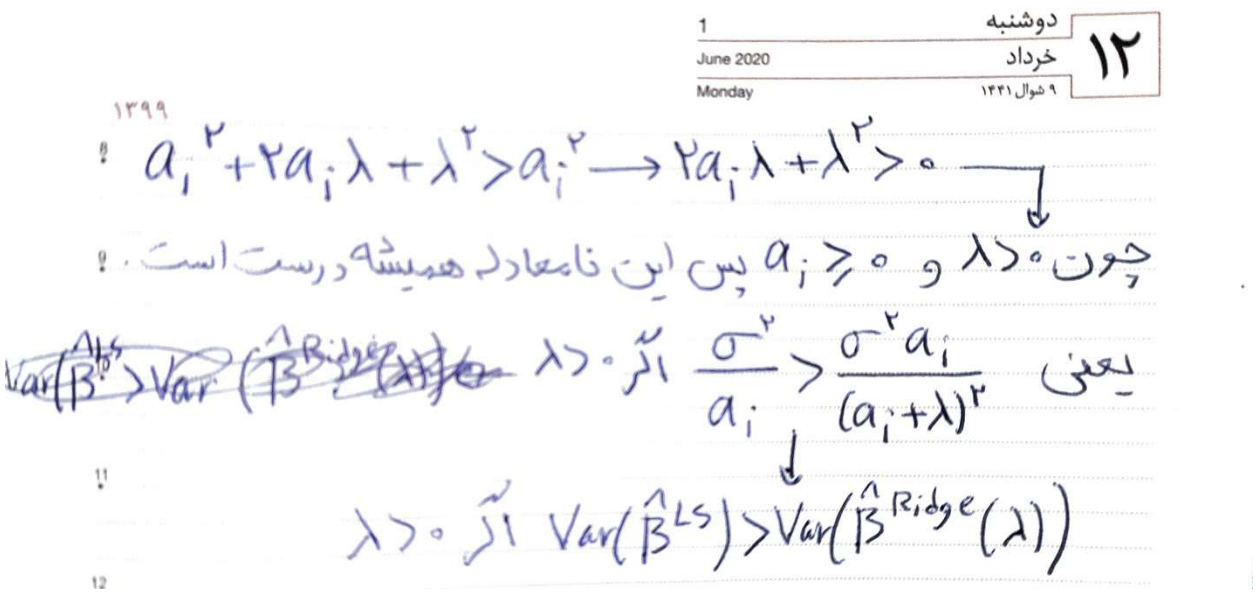
diagonal

$Var(\hat{\beta}^{Ridge}(\lambda)_{(i)}) = \frac{\sigma^2 a_i}{(a_i+\lambda)^2}$

حال باید اثبات کنیم اگر $\lambda > 0$ باشد آنگاه

$\frac{\sigma^2}{a_i} > \frac{\sigma^2 a_i}{(a_i+\lambda)^2}$

$\frac{1}{a_i} > \frac{a_i}{(a_i+\lambda)^2}$ $\times a_i (a_i+\lambda)^2$ $\rightarrow (a_i+\lambda)^2 > a_i^2 \rightarrow$ صفحه بعد



ب)

مرحله ۱: بیان واریانس پیش بینی ها

در اینجا، D_x یک ماتریس diagonal حاوی مقادیر تکین X است و $\hat{Y}(\lambda)$ مقدار پیش‌بینی‌شده برای رگرسیون Ridge است.

پیش‌بینی $\hat{Y}(\lambda)$ برای رگرسیون Ridge توسط فرم بسته زیر انجام میشود:

$$\hat{Y}(\lambda) = X\widehat{\beta^{\text{Ridge}}}(\lambda)$$
$$\widehat{\beta^{\text{Ridge}}}(\lambda) = (X^\top X + \lambda I)^{-1} X^\top y$$
$$\hat{Y}(\lambda) = X\widehat{\beta^{\text{Ridge}}}(\lambda) = X.(X^\top X + \lambda I)^{-1} X^\top y$$

با استفاده از عبارت واریانس برای $\widehat{\beta^{\text{Ridge}}}(\lambda)$، می‌توانیم واریانس $\hat{Y}(\lambda)$ را محاسبه کنیم. واریانس $\hat{Y}(\lambda)$ برابر است با:

$$\text{Var}\left[\hat{Y}(\lambda)\right] = X\text{Var}\left[\widehat{\beta^{\text{Ridge}}}(\lambda)\right]X^\top$$
$$\text{Var}\left[\widehat{\beta^{\text{Ridge}}}(\lambda)\right] = \sigma^2(X^\top X + \lambda I)^{-1}X^\top X(X^\top X + \lambda I)^{-1}$$
$$\text{Var}\left[\hat{Y}(\lambda)\right] = \sigma^2 X(X^\top X + \lambda I)^{-1}X^\top X(X^\top X + \lambda I)^{-1}X^\top$$

مرحله ۲: استفاده از SVD

با استفاده از تجزیه مقدار های منفرد X (SVD)، داریم:

$$X = UD_xV^\top$$

که در آن U و V ماتریس های متعامد(orthogonal) هستند و D_x ماتریس diagonal حاوی مقادیر منفرد X است. می توانیم واریانس پیش بینی ها را به صورت زیر بازنویسی کنیم:

$$\text{Var}\left[\hat{Y}(\lambda)\right] = \sigma^2 UD_xV^\top.(VD_x^\top U^\top UD_xV^\top + \lambda I)^{-1}.VD_x^\top U^\top UD_xV^\top.(VD_x^\top U^\top UD_xV^\top + \lambda I)^{-1}VD_x^\top U^\top$$
$$\text{Var}\left[\hat{Y}(\lambda)\right] = \sigma^2 UD_xV^\top.(VD_x^\top D_xV^\top + \lambda I)^{-1}.VD_x^\top D_xV^\top.(VD_x^\top D_xV^\top + \lambda I)^{-1}VD_x^\top U^\top$$
$$\text{Var}\left[\hat{Y}(\lambda)\right] = \sigma^2 UD_x(D_x^\top D_x + \lambda I)^{-1}D_x^\top D_x(D_x^\top D_x + \lambda I)^{-1}D_x^\top U^\top$$

مرحله ۳: trace واریانس

trace ماتریس کوواریانس مجموع عناصر قطری آن است. از آنجایی که U یک ماتریس متعامد است، عملیات trace تحت تبدیل های متعامد ثابت است، بنابراین می توانیم روی ماتریس D_x diagonal تمرکز کنیم. trace می شود:

$$\mathrm{tr}\{\mathrm{Var}[\hat{Y}(\lambda)]\} = \mathrm{tr}\{\sigma^2 U D_x (D_x^\top D_x + \lambda I)^{-1} D_x^\top D_x (D_x^\top D_x + \lambda I)^{-1} D_x^\top U^\top\}$$
$$\mathrm{tr}\{\mathrm{Var}[\hat{Y}(\lambda)]\} = \sigma^2 \mathrm{tr}\{U D_x (D_x^\top D_x + \lambda I)^{-1} D_x^\top D_x (D_x^\top D_x + \lambda I)^{-1} D_x^\top U^\top\}$$
$$\mathrm{tr}\{\mathrm{Var}[\hat{Y}(\lambda)]\} = \sigma^2 \mathrm{tr}\{D_x (D_x^\top D_x + \lambda I)^{-1} D_x^\top D_x (D_x^\top D_x + \lambda I)^{-1} D_x^\top\}$$
$$\mathrm{tr}\{\mathrm{Var}[\hat{Y}(\lambda)]\} = \sigma^2 \mathrm{tr}\{D_x D_x^\top D_x D_x^\top (D_x^\top D_x + \lambda I)^{-2}\}$$
$$\mathrm{tr}\{\mathrm{Var}[\hat{Y}(\lambda)]\} = \sigma^2 \mathrm{tr}\{(D_x D_x^\top)^2 (D_x^\top D_x + \lambda I)^{-2}\}$$
$$\mathrm{tr}\{\mathrm{Var}[\hat{Y}(\lambda)]\} = \sigma^2 \sum_{j=1}^{p} (D_x)_{jj}^4 \left[(D_x)_{jj}^2 + \lambda\right]^{-2}$$

این رابطه مورد نیاز را ثابت می کند. نتیجه نشان می‌دهد که چگونه رگرسیون Ridge بر واریانس پیش‌بینی‌ها از طریق مقادیر منفرد X و پارامتر منظم‌سازی λ تأثیر می‌گذارد. با افزایش λ، واریانس کاهش می‌یابد که خاصیت کاهش واریانس رگرسیون Ridge را نشان می‌دهد.

پایان