آرمین سعادت بروجنی
96105829

تمرین سوم یادگیری ماشین

1.

(a)

$$L_{MSE}(\theta) = \frac{1}{n}\sum (x_i-\theta)^2 \Rightarrow \theta^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{n}\sum_1 (x_i-\theta)^2 = \underset{\theta}{\operatorname{argmin}} \sum_1 (\theta-x_i)^2$$

$$\Rightarrow \frac{d}{d\theta}\sum_1 (\theta-x_i)^2 = 0 \Rightarrow 2\sum_1 (\theta-x_i) = 0 \Rightarrow \sum_{i=1}^{n}\theta = \sum_{i=1} x_i$$

$$\Rightarrow n\theta = \sum x_i \Rightarrow \theta^* = \frac{\sum x_i}{n} \longrightarrow$$ میانگین نمونه‌ها.

$$L_{MAE}(\theta) = \frac{1}{n}\sum |x_i-\theta| = \frac{1}{n}\Big(\sum_{\text{for } x_i>\theta^*}(x_i-\theta) + \sum_{\text{for } x_j<\theta^*}(\theta-x_j) + \sum_{\text{for } x_k=\theta^*}(0) \Big)$$

Suppose there are m_1 x_j which are less than θ^*.
Suppose there are m_2 x_i which are greater than θ^*.

$$\frac{d}{d\theta}\frac{1}{n}\Big(\sum_{\text{for } x_i>\theta^*}(x_i-\theta) + \sum_{\text{for } x_j<\theta^*}(\theta-x_j) \Big) = 0 \Rightarrow \frac{1}{n}(-m_2+m_1) = 0$$

$$\Rightarrow \boxed{m_1 = m_2} \longrightarrow$$ پس θ^* باید به گونه ای انتخاب شود که تعداد داده‌ای از x_i ها از آن کمتر و بیشتر باشند. پس θ^*، میانه نمونه‌ها است.

b) نکته اول در رابطه با بحث مشتق پذیری است. از MSE می‌توان مشتق گرفت که محاسبات را ساده‌تر می‌کند، مقدار بهینه را راحت‌تر می‌دهد اما MAE مشتق پذیر نیست و نیاز به حالت بندی دارد.

نکته دوم در رابطه با انعطاف پذیری آن‌ها در مواجه با داده پرت است. همانگونه که مشاهده شد MSE به مقدار داده‌ها بسیار وابستگی نشان می‌دهد و حتی ۱ داده پرت بسیار بزرگ می‌تواند مقدار بهینه را به شدت تحت تاثیر قرار دهد. اما در MAE میانه مهم است و یک داده پرت خیلی بزرگ با یک داده صرفاً بزرگ‌تر از میانه تفاوتی در میزان تغییر مقدار بهینه ندارد. پس MAE به مقادیر کمتر حساس است تا MSE که بسته به شرایط مسئله می‌تواند به عنوان مزیت یا ایراد شناخته شود. بهتر است MAE به داده‌های نزدیک حساس‌تر شود و MSE بی‌تفاوت‌تر به داده دور.

$$\text{Huber loss} = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \le \delta \\ \delta(|a| - \frac{1}{2}\delta) & \text{o.w.} \end{cases}$$

$$\cosh = f(x) = \ln\left(\frac{e^{x} + e^{-x}}{2}\right), \quad L_{\text{log-cosh}}(x, \theta) = \sum_i \log(\cosh(\theta - x))$$

log-cosh برای مقادیر کوچک شبیه $\frac{x^2}{2}$ و برای مقادیر بزرگ شبیه $|x| - \log(2)$ عمل می‌کند.
بنابراین برای مقادیر کوچک (عملاً نزدیک به میانگین با یک شیفت) مانند MSE عمل می‌کند اما چون عملکرد متفاوتی برای مقادیر دور از میانگین دارد، حساسیت کمتری نسبت به داده های پرت نشان می‌دهد.
در ضمن مشتق پذیر هم هست اما ضعف e^{x}، e^{-x} و ln دارد که از نظر محاسباتی سنگین تر از MSE است.
Huber نیز همان طور که در تعریفش ذکر شده، برای داده های نزدیک $\frac{a^2}{2}$ است و عملاً عملکرد MSE را دارد. اما برای داده‌های دور عملکرد خطی دارد. بنابراین بسیار مشابه log-cosh عمل می‌کند.
اما ضعف حالت تابعی دارد برای محاسبه، البته که توابع سبکی هستند از نظر محاسباتی.

در نتیجه می‌توان گفت تا حدودی مشکلات MSE، MAE را حل می‌کنند. گویا نسبت به داده های نزدیک به میانگین شبیه MSE و نسبت به داده‌های پرت شبیه MAE عمل می‌کنند.

پس هر ۲ روش پیشنهادی، برای خطای زیاد مانند MAE و برای خطای کم شبیه MSE رفتار می‌کنند. (تقریباً!)
log-cosh مشتق پذیر هم هست در هر نقطه که از Huber بهتر است،
Huber محاسبات ریاضی ساده‌تری دارد.

سؤال ۲)

تحلیل کلی: با افزایش تعداد iteration ها و جلو رفتن فرایند یادگیری، خطای train کاهش یافته و به یک مقدار optimal رسیده است. خطای Validation2 نیز کاهش یافته و به یک مقدار optimal رسیده است که البته از مقدار خطای train بیشتر است که خب منطقی است چرا که پارامترها بر اساس بهبود train تغییر می‌کنند. مشاهده می‌شود که در هر ۳ نمودار، rate بهبود در ابتدا سریع و زیاد است، به مرور این rate کم می‌شود. دلیل آن این است که پارامترها در ابتدا مقدار خیلی پرتی دارند و در هر iteration، گام‌های بزرگی به سمت حالت بهینه برداشته می‌شود که با ادامه فرایند این کار سرعت کمتری به خود می‌گیرد. اما اتفاق عجیب این است که پس از یک نقطه‌ای، خطای Validation1 دیگر کم نشده و حتی افزایش می‌یابد.

توزیع متفاوت: اگر از ۲ توزیع متفاوت آمده باشند، می‌توان اینگونه تحلیل کرد که توزیع train با Validation2 یکسان است و با Validation1 متفاوت است. پارامترها در ابتدا خیلی بد هستند، بنابراین احتمال دارد که حرکت به سمت بهتر شدن train حتی باعث بهتر شدن Validation1 هم شده باشد. اما از یک نقطه به بعد پارامترها خیلی تخصصی و دقیق روی توزیع train متمرکز می‌شوند و خب این لزوما با توزیع Validation1 سازگار نیست و باعث افزایش خطا روی آن می‌شود. یعنی مسیر Validation1 جدا می‌شود و عملا overfit رخ می‌دهد برای آن.

توزیع یکسان: اگر توزیع یکی باشد، می‌توان به نحوه نمونه برداری (Sampling) ایراد گرفت. مثلا شاید مقادیر Validation1 ۱۰٪ بالای توزیع را تشکیل داده باشند. مثلا sort شده بعد sample شده!

یک احتمال دیگر این است که در زمان training، برای بهبود hyper-parameters به Validation2 رجوع شده باشد و تغییرات ایجاد شده باعث بهبود خطای Validation2 می‌شود اما این کار برای Validation1 ناسازگار بوده و باعث افزایش خطا می‌شود. مثلا اگر برای تعیین hyper-parameter به هر دو valid-set رجوع می‌شد، احتمالا خطای هر دو کاهش می‌یافت اما به مقدار بهینه بزرگ‌تری نسبت به مقدار بهینه الآن Validation2 می‌رسید.

← یک گزینه محتمل دیگر این است که تعداد داده‌ها خیلی کم است. به همین دلیل اینگونه دارد رفتار می‌کند.

سؤال ۳)

$$E[L_q] = \iint |f(x)-y|^q \, p(x,y)\, dx\, dy$$

در رابطه بالا، مقادیر $f(x)$ به ازای هر x مستقل محاسبه می‌شود. پس تقریباً می‌توانیم مسئله را به مسئله زیر تبدیل کنیم:

$\int |f(x)-y|^q \, p(y|x)\, dy \longrightarrow$ این عبارت باید کمینه شود.

$$\Rightarrow \frac{d}{df(x)} \int |f(x)-y|^q \, p(y|x)\, dy = 0$$

$$\Rightarrow \frac{d}{df(x)} \left\{ \int_{-\infty}^{f(x)} (f(x)-y)^q \, p(y|x)\, dy + \int_{f(x)}^{\infty} (y-f(x))^q \, p(y|x)\, dy \right\} = 0$$

$$\Rightarrow \int_{-\infty}^{f(x)} q(f(x)-y)^{q-1} \, p(y|x)\, dy - \int_{f(x)}^{\infty} q(y-f(x))^{q-1} \, p(y|x)\, dy = 0$$

$$\Rightarrow \int_{-\infty}^{f(x)} (f(x)-y)^{q-1} \, p(y|x)\, dy = \int_{f(x)}^{\infty} (y-f(x))^{q-1} \, p(y|x)\, dy$$

$$\int_{-\infty}^{f(x)} p(y|x)\, dy = \int_{f(x)}^{\infty} p(y|x)\, dy \quad : q=1$$

پس مساحت سمت چپ $f(x)$ با مساحت سمت راست آن باید برابر باشد. تقریباً همان میانه است در توزیع دیده شده.

$q=0$: اگر دقیقاً صفر باشد، مقدار $|f(x)-y|^q$ صفر شده و اصلا نمی‌توان مشتق گرفت و $E[L_q]=1$ می‌شود.

اگر بگوییم $q \to 0$ میل می‌کند، باز هم مقادیر $|f(x)-y|^q$ بسیار نزدیک به 1 است. البته در همسایگی نزدیک $f(x)$، این مقدار به صفر نزدیک است. با در نظر گرفتن این که y نرمال است، می‌توان گفت بیشترین کاهش در نزدیکی $y=f(x)$ است که بیشترین $P(y)$ را دارد.

سوال ۴)

$$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \quad , \quad S(x) = \frac{1}{1+e^{-x}}$$

$$S(2x) = \frac{1}{1+e^{-2x}} = \frac{e^{x}}{e^{x}+e^{-x}} \Rightarrow 2S(2x)-1 = \frac{2e^{x}}{e^{x}+e^{-x}}-1 = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$$

$$\Rightarrow \tanh(x) = 2S(2x)-1$$

در ویدیو گفته شد که رابطه پارامترها برای $f(2x, w^{sigmoid})$, $f(x, w^{tanh})$ گفته شود.

$$f(2x, w^{sig}) = w_0^{sig} + \sum_{j=1}^{m} w_j^{sig} S(2x)$$

$$f(x, w^{tanh}) = w_0^{tanh} + \sum_{j=1}^{m} w_j^{tanh} \tanh(x) = w_0^{tanh} + \sum_{j=1}^{m} w_j^{tanh} (2S(2x)-1)$$

$$= w_0^{tanh} + \sum_{j=1}^{m} \{2w_j^{tanh} S(2x) - w_j\} = w_0^{tanh} - \sum_{j=1}^{m} w_j^{tanh} + \sum_{j=1}^{m} 2w_j^{tanh} S(2x)$$

$$\Rightarrow w_0^{sigmoid} = w_0^{tanh} - \sum_{j=1}^{m} w_j^{tanh} \quad , \text{ for } j=1 \text{ to } m: \; w_j^{sigmoid} = 2w_j^{tanh}$$

یا به شکل دیگر:

$$w_0^{tanh} = w_0^{sigmoid} + \frac{1}{2}\sum_{j=1}^{m} w_j^{sigmoid} \quad , \text{ for } j=1 \text{ to } m: \; w_j^{tanh} = \frac{1}{2} w_j^{sigmoid}$$

۵)

$$y(x,w) = w_0 + \sum_{i=1}^{D} w_i x_i \longrightarrow \text{our linear model.}$$

$$E_D(w) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n,w) - t_n\}^2 \longrightarrow \text{sum square error.}$$

$$x = [x_1, x_2, \dots, x_d]^T \xrightarrow{\text{add noise}} x\_noisy = [x_1+\epsilon_1, x_2+\epsilon_2, \dots, x_d+\epsilon_d]^T$$

$$E_D(w)_{noisy} = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n\text{-noisy}, w) - t_n\}^2 = \frac{1}{2} \sum_{n=1}^{N} \Big\{\Big[w_0 + \sum_{i=1}^{D} w_i (x_i + \epsilon_i)\Big] - t_n\Big\}^2$$

$$= \frac{1}{2} \sum_{n=1}^{N} \Big\{\Big[w_0 + \sum_{i=1}^{D} w_i x_i\Big] - t_n + \sum_{i=1}^{D} w_i \epsilon_i\Big\}^2 = \frac{1}{2} \sum_{n=1}^{N} \Big\{y(x_n,w) - t_n + \sum_{i=1}^{D} w_i \epsilon_i\Big\}^2$$

$$= \frac{1}{2} \sum_{n=1}^{N} \Big\{\underbrace{(y(x_n,w) - t_n)^2}_{A} + \underbrace{\Big(\sum_{i=1}^{D} w_i \epsilon_i\Big)^2}_{B} + \underbrace{2(y(x_n,w) - t_n)\Big(\sum_{i=1}^{D} w_i \epsilon_i\Big)}_{C}\Big\}$$

$$E_\epsilon[E_D(w)] = E_\epsilon\Big[\frac{1}{2} \sum_{n=1}^{N} (A + B + C)\Big] = \frac{1}{2} \sum_{n=1}^{N} E_\epsilon[A] + \frac{1}{2} \sum_{n=1}^{N} E_\epsilon[B] + \frac{1}{2} \sum_{n=1}^{N} E[C]$$

$$E_\epsilon[A] = E_\epsilon[(y(x_n,w) - t_n)^2] = (y(x_n,w) - t_n)^2$$

$$E_\epsilon[B] = E_\epsilon\Big[\Big(\sum_{i=1}^{D} w_i \epsilon_i\Big)^2\Big] = E_\epsilon\Big[\sum_{i=1}^{D} \sum_{j=1}^{D} w_i w_j \epsilon_i \epsilon_j\Big] = \sum_{i=1}^{D} \sum_{j=1}^{D} w_i w_j E_\epsilon[\epsilon_i \epsilon_j]$$

$$= \sigma^2 \sum_{i=1}^{D} \sum_{j=1}^{D} w_i w_j \delta_{ij} \longrightarrow E_\epsilon[\epsilon_i \epsilon_j] = \sigma^2 \delta_{ij}, \quad \epsilon_i \sim N(0, \sigma^2)$$ چون

$$\delta_{ij} \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \Bigg\} \; E_\epsilon[B] = \sigma^2 \sum_{i=1}^{D} w_i^2$$

$$E_\epsilon[C] = E_\epsilon\left[2\left(\sum_{i=1}^{D} w_i\epsilon_i\right)(y(x_n,w)-t_n)\right]$$

$$= 2(y(x_n,w)-t_n)\, E_\epsilon\left[\sum_{i=1}^{D} w_i\epsilon_i\right] = 2(y(x_n,w)-t_n) \sum_{i=1}^{D} E_\epsilon[w_i\epsilon_i]$$

$$\epsilon_i \sim N(0,\sigma^2) \Rightarrow E_\epsilon[\epsilon_i]=0 \Rightarrow E_\epsilon[w_i\epsilon_i]=0$$

$$\Rightarrow E_\epsilon[C]=0$$

$$\Rightarrow E_\epsilon[E_D(w)] = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,w)-t_n\}^2 + \frac{\sigma^2}{2}\sum_{i=1}^{D} w_i^2$$

$$\Rightarrow E\left[E_{D\text{-noisy}}(w)\right] = E_D(w) + \frac{\sigma^2}{2}\sum_{i}^{D} w_i^2 \quad \boxed{\checkmark}$$

نکته: فرض شد که $\epsilon_i \sim N(0,\sigma^2)$ ، همچنین $\sigma^2\delta_{ij} = E(\epsilon_i\epsilon_j)$

که داریم: $\delta_{ij} = \begin{cases} 1 & i=j \\ 0 & o.w. \end{cases}$

$$L(w) = \sum_{i=1}^{n} f_i \left(y_i - w^T x_i\right)^2 \Rightarrow w^* = \underset{w}{\arg\min}\ L(w)$$

سؤال ۶)

derivative of $L(w)$ with respect to w = $\nabla_w L(w) = 0$

$$\nabla_w L(w) = -2 \sum_{i=1}^{n} F_i \left(y_i - w^T x_i\right) x_i^T = \sum F_i y_i x_i^T - \sum F_i w^T x_i x_i^T = 0$$

$$\Rightarrow \sum_{i=1}^{n} F_i y_i x_i^T = w^T \sum_{i=1}^{n} F_i x_i x_i^T \Rightarrow w^T = \sum_{i=1}^{n} F_i y_i x_i^T \left(\sum_{i=1}^{n} F_i x_i x_i^T\right)^{-1}$$

$$\Rightarrow w = \left(\sum F_i y_i x_i^T \left(\sum F_i x_i x_i^T\right)^{-1}\right)^T = \left(\sum F_i x_i x_i^T\right)^{-T} \left(\sum F_i y_i x_i^T\right)^T$$

$$= \left(\sum F_i x_i x_i^T\right)^{-1} \sum F_i x_i y_i^T$$

$$\sum F_i x_i x_i^T = \left[\, x_1 \,\middle|\, x_2 \,\middle|\, \cdots \,\middle|\, x_n \,\right] \begin{bmatrix} F_1 & & 0 \\ & F_2 & \\ 0 & \ddots & F_n \end{bmatrix} \begin{bmatrix} x_1 \\ \hline x_2 \\ \hline \vdots \\ \hline x_n \end{bmatrix} = X F X^T$$

$$\sum F_i x_i y_i^T = \left[\, x_1 \,\middle|\, x_2 \,\middle|\, \cdots \,\middle|\, x_n \,\right] \begin{bmatrix} F_1 & & 0 \\ & F_2 & \\ 0 & \ddots & F_n \end{bmatrix} \begin{bmatrix} y_1 \\ \hline y_2 \\ \hline \vdots \\ \hline y_n \end{bmatrix} = X F Y^T$$

$$\Rightarrow w^* = \left(X F X^T\right)^{-1} X F Y^T$$

$$X = \left[\, x_1 \,\middle|\, x_2 \,\middle|\, \cdots \,\middle|\, x_n \,\right],\quad Y = \left[\, y_1 \,\middle|\, y_2 \,\middle|\, \cdots \,\middle|\, y_n \,\right],\quad F = \begin{bmatrix} F_1 & 0 & \cdots & 0 \\ 0 & F_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & F_n \end{bmatrix}$$

یا با ماتریس مرسوم تر:

$$X = \begin{bmatrix} x_1^{(1)}, & \cdots & x_m^{(1)} \\ & \vdots & \\ x_1^{(n)} & \cdots & x_m^{(n)} \end{bmatrix},\quad Y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix} \Rightarrow w^* = \left(X^T F X\right)^{-1} X^T F Y$$

سؤال ۷)

a)

$$E_D(w) = \frac{1}{2}\sum_{i=1}^{n}\left(y^{(i)} - w^T x^{(i)}\right)$$

$$\Rightarrow w^* = (X^T X)^{-1} X^T Y$$

از طرفی فقط فیچر j ام برای ما مهم است، پس هر داده x به شکل زیر دیده می‌شود:

$$x = [x_1, x_2, \dots, x_m] \longrightarrow x = [0, 0, \dots, x_j, 0, \dots 0]$$

پس عملاً ماتریس X به این شکل دیده می‌شود:

$$X = \left[\, 0 \,\middle|\, 0 \,\middle|\, \cdots \,\middle|\, x_j \,\middle|\, 0 \,\middle|\, \cdots \,\middle|\, 0 \,\right]$$

$$\Rightarrow X^T X = x_j^T x_j \Rightarrow (X^T X)^{-1} = (x_j^T x_j)^{-1} = \frac{1}{x_j^T x_j} \quad , \quad X^T y = x_j^T y$$

$$\Rightarrow w_j^* = \frac{x_j^T y}{x_j^T x_j}$$

b)

$$w^* = (X^T X)^{-1} X^T y$$

$$X \rightarrow \text{orthogonal} \Rightarrow X^T X = \begin{bmatrix} \|x_1\|^2 & & 0 \\ & \ddots & \\ 0 & & \|x_m\|^2 \end{bmatrix}$$

$$\Rightarrow w^* = \begin{bmatrix} \frac{1}{\|x_1\|^2} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\|x_m\|^2} \end{bmatrix} X^T y \Rightarrow w_j = \frac{x_j^T y}{\|x_j\|^2} = \frac{x_j^T y}{x_j^T x_j}$$

از طرفی در (a) ثابت کردیم که w_j به دست آمده از تنها فیچر j، $\frac{x_j^T y}{x_j^T x_j}$ می‌شود.

پس مسئله حل شد!

c)

$$w^* = (X^T X)^{-1} X^T y$$

عملاً می‌توان به X اینطور نگاه کرد:

$$X = \begin{bmatrix} 1, 0, 0, \dots, x_j^{(1)}, 0 \dots 0 \\ \vdots \\ 1, 0, \dots, x_j^{(n)}, 0 \dots 0 \end{bmatrix}$$

که خب مثل این می‌ماند که X کلا ۲ ویژگی داشته باشد، پس عملاً برای X داریم:

$$X = \begin{bmatrix} 1 & x_j^{(1)} \\ 1 & x_j^{(2)} \\ \vdots & \vdots \\ 1 & x_j^{(n)} \end{bmatrix} \Rightarrow X^T = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_j^{(1)} & x_j^{(2)} & \dots & x_j^{(n)} \end{bmatrix}$$

$$\Rightarrow X^T X = \begin{bmatrix} n & nE[x_j] \\ nE[x_j] & \sum x_j^2 \end{bmatrix}, \quad Var(x_j) = E[x_j^2] - E[x_j]^2 = \frac{\sum x_j^2}{n} - E[x_j]^2$$

$$\Rightarrow \sum x_j^2 = n(Var(x_j) + E[x_j]^2)$$

$$X^T y = \begin{bmatrix} nE[y] \\ x_j^T y \end{bmatrix}, \quad Cov(x_j, y) = \sum_i (x_j^{(i)} - E[x_j])(y^{(i)} - E[y]) \times \frac{1}{n}$$

$$\Rightarrow x_j^T y = n(Cov(x_j, y) + E[x_j]E[y])$$

$$\Rightarrow (X^T X)^{-1} X^T y = \frac{n}{n \times Var(x_j)} \begin{bmatrix} Var(x_j) + E[x_j]^2 & -E[x_j] \\ -E[x_j] & 1 \end{bmatrix} \begin{bmatrix} E[y] \\ Cov(x_j, y) + E[x_j]E[y] \end{bmatrix}$$

$$= \begin{bmatrix} \dfrac{Var(x_j)E[y] - E[x_j]Cov(x_j, y)}{Var(x_j)} \\ \dfrac{Cov(x_j, y)}{Var(x_j)} \end{bmatrix} = \begin{bmatrix} w_0 \\ w_j \end{bmatrix} \Rightarrow \quad w_j = \frac{Cov(x_j, y)}{Var(x_j)}$$

$$w_0 = E[y] - w_j E[x_j]$$