دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

استاد درس: دکتر حمید بیگی

پاییز ۱۴۰۱

تمرین اول

درس یادگیری ژرف

نام و نام خانوادگی: امیر پورمند

شماره دانشجویی: ۹۹۲۱۰۲۵۹

آدرس ایمیل : pourmand1376@gmail.com

۱ سوال ۱

۱.۱ الف

ابتدا در نظر میگیرم که $w_0 = b$ و $x_0 = 1$ پس داریم.

$$\hat{y} = \sum_{i=1}^{d} w_i x_i + b = \sum_{i=0}^{d} w_i x_i$$

$$\frac{\partial L}{\partial w_j} = \frac{\partial}{\partial w_j} \frac{1}{2n} \sum_{i=1}^{n} \left(y^{(i)} - \hat{y}^{(i)} \right)^2$$

$$= \frac{1}{n} \sum_{i=1}^{n} \left(y^{(i)} - \hat{y}^{(i)} \right) \frac{\partial}{\partial w_j} \sum_{i=0}^{d} w_i x_i$$

$$= \frac{1}{n} \sum_{i=1}^{n} \left(y^{(i)} - \hat{y}^{(i)} \right) x_j^{(i)}$$

نهایتا رابطه به روز رسانی به فرم زیر خواهد بود.

$$w_j := w_j - \alpha \frac{\partial L}{\partial w_j}$$

$$w_j := w_j - \frac{\alpha}{n} \sum_{i=1}^{n} \left(y^{(i)} - \hat{y}^{(i)} \right) x_j^{(i)}$$

۲.۱ ب

ابتدا $\hat{y}^{(i)}$ را به فرم زیر مینویسیم. $\hat{y}^{(i)} = w^T x$

پس داریم: (در نظر داشته باشید در خط دوم عبارت $\frac{1}{2n}$ به این علت حذف شد که بود و نبود آن نهایتا تاثیری در مشتق گیری ندارد چون میخواهیم عبارت را مساوی صفر قرار دهیم.)

$$L(w) = \frac{1}{2n} \sum_{i=1}^{n} \left(\hat{y}^{(i)}) - y^{(i)} \right)^2$$

$$L(w) = \frac{1}{2n} (Xw - y)^T (Xw - y)$$

$$L(w) = \left((Xw)^T - y^T \right) (Xw - y)$$

$$L(w) = (Xw)^T Xw - (Xw)^T y - y^T (Xw) + y^T y$$

$$L(w) = w^T X^T Xw - 2(Xw)^T y + y^T y$$

حال که قدری ساده سازی انجام شد از عبارت نسبت به w مشتق میگیریم.

$$\frac{\partial L}{\partial w} = 2X^T Xw - 2X^T y = 0$$

که نتیجه میشود:

$$X^T X w = X^T y$$

یا میتوان گفت:

$$w = \left(X^T X\right)^{-1} X^T y$$

۳.۱ ج

در مسئله form closed اگر نیاز باشد که N متغیر مستقل محاسبه شوند یک ماتریس $N \times N$ باید معکوس شود که از مرتبه $O(N^3)$ است ولی اگر مسئله را با روش descent gradient حل کنیم از مرتبه خطی یا همان N میشود که به مراتب ساده تر است.

۲ سوال ۲

مشکل محوشوندگی مشتق وقتی پیش می‌آید که در فرایند backpropagation یک سری عدد خیلی کوچک در هم ضرب میشوند که باعث میشود لایه های اول در اپدیت شدن وزن ها مشکل داشته باشند و تقریبا میتوان گفت اگر این مشکل پیش بیاید، مسئله هیچ وقت converge نمیکند.

حال این مشکل در توابعی اصولا پیش میآیند که مشتق انها بسیار محدود است و این باعث میشود که مشکل vanishing gradient یا gradient diminishing بوجود آید.

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$

به سادگی مشخص است که

$$\sigma(t)_{t \to \infty} = 1.0$$

و البته

$$\sigma(t)_{t \to -\infty} = 0.0$$

از طرفی مشتق تابع نیز همواره بین صفر و یک محدود است زیرا

$$\sigma'(t) = \sigma(t)(1.0 - \sigma(t))$$
$$0 \leqslant \sigma(t) \leqslant 1$$
$$0 \leqslant 1 - \sigma(t) \leqslant 1$$
$$0 \leqslant \sigma(t)\left(1 - \sigma(t)\right) \leqslant 1$$

که این باعث میشود مشکل محوشوندگی مشتق را داشته باشیم. در حالی که تابع ReLU به صورت زیر تعریف میشود:

$$ReLU(x) \ = \ max(0, x) = \begin{cases} x & x \geqslant 0 \\ 0 & x < 0 \end{cases}$$

$$ReLU'(x) = \begin{cases} 1 & x > 0 \\ 0 & x < 0 \\ undefined & x = 0 \end{cases}$$

پس این تابع مشکل محوشوندگی گرادیان را ندارد ولی مشکل دیگری دارد به نام gradient dead که اگر مشتق صفر شود بوجود میاید.

۱.۲ ب

هدف مقداردهی Xaviar این است که وزن ها به ترتیبی نسبت داده شوند که واریانس توابع فعال ساز در تمامی لایه ها با هم برابر باشند. این واریانس برابر تضمین میکند که مشکل gradient vanishing پیش نیاید.

البته فرض میکنیم میانگین مقادیر ورودی و وزن‌ها صفر است که اگر صفر نبود میتوانیم به سادگی داده ورودی را شیفت دهیم و البته فرض idd نیز که همیشه موجود است و از تابع تانژانت هایپربولیک استفاده میکنیم که داریم:

$$\mathrm{Var}\left(a^{[l]}\right) \approx \mathrm{Var}\left(z^{[l]}\right)$$

ابتدای امر با توجه به فرض الگوریتم زیور داریم:

$$\mathrm{Var}\left(a_k^{[l]}\right) = \mathrm{Var}\left(z_k^{[l]}\right) = \mathrm{Var}\left(\sum_{j=1}^{n^{[l-1]}} w_{kj}^{[l]} a_j^{[l-1]}\right) = \sum_{j=1}^{n^{[l-1]}} \mathrm{Var}\left(w_{kj}^{[l]} a_j^{[l-1]}\right)$$

حال با توجه به فرمول زیر ضرب را به جمع تبدیل میکنیم

$$\mathrm{Var}(XY) = E[X]^2 \mathrm{Var}(Y) + \mathrm{Var}(X) E[Y]^2 + \mathrm{Var}(X)\,\mathrm{Var}(Y)$$

پس داریم:

$$\mathrm{Var}\left(w_{kj}^{[l]} a_j^{[l-1]}\right) = E\left[w_{kj}^{[l]}\right]^2 \mathrm{Var}\left(a_j^{[l-1]}\right) + \mathrm{Var}\left(w_{kj}^{[l]}\right) E\left[a_j^{[l-1]}\right]^2 + \mathrm{Var}\left(w_{kj}^{[l]}\right) \mathrm{Var}\left(a_j^{[l-1]}\right)$$

که به خاطر این است که در سوال فرض کرده ایم

$$\mathrm{Var}\left(w_{kj}^{[l]}\right) = \mathrm{Var}\left(w_{11}^{[l]}\right) = \mathrm{Var}\left(w_{12}^{[l]}\right) = \cdots = \mathrm{Var}\left(W^{[l]}\right)$$

و البته فرض کرده ایم:

$$\mathrm{Var}\left(a_j^{[l-1]}\right) = \mathrm{Var}\left(a_1^{[l-1]}\right) = \mathrm{Var}\left(a_2^{[-1]}\right) = \cdots = \mathrm{Var}\left(a^{l-1}\right)$$

پس با همین ایده نتیجه میشود:

$$\mathrm{Var}\left(z^{[l]}\right) = \mathrm{Var}\left(z_k^{[l]}\right)$$

که نهایتا عبارت زیر نتیجه میشود:

$$\mathrm{Var}\left(a^{[l]}\right) = n^{[l-1]}\,\mathrm{Var}\left(W^{[l]}\right)\mathrm{Var}\left(a^{[l-1]}\right)$$

پس نتیجه میگیریم:

$$\mathrm{Var}\left(W^{[l]}\right) = \frac{1}{n^{[l-1]}}$$

یعنی بهتر است واریانس تک تک وزن های هر لایه رابطه معکوس با تعداد نورون های لایه داشته باشد. حال با توجه به این که ضرب $\mathrm{Var}\left(W^{[l]}\right) * n^{[l-1]}$ ۱ میشود مشکل محوشوندگی حل میشود و اگر کمتر از یک و بیشتر از یک بود هر کدام میتوانست باعث بوجود آمدن مشکل انفجار یا محوشوندگی گرادیان شود.

۲.۲ ج

میدانیم در هر شبکه عصبی

$$z = \sum_{i=0}^{n} w_i a_i + b$$

است بنابراین وقتی هر کدام از ضرایب a_i یا z_i خیلی بزرگ باشند باعث میشود که مشتق ها بسیار کوچک شود و نتیجتا فرایند یادگیری بسیار کند میشود. این بدین علت است که تابع سیگموید به صورت زیر تعریف میشود:

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$

به سادگی مشخص است که

$$\sigma(t)_{t \to \infty} = 1.0$$

و البته

$$\sigma(t)_{t \to -\infty} = 0.0$$

حال برای مشتق داریم:

$$\sigma'(t)_{t \to -\infty} = \sigma(t)(1 - \sigma(t)) \quad (۱)$$

که به ازای مقادیر خیلی کوچک یا مقادیر خیلی بزرگ یکی از دو ضرب شونده به صفر نزدیک میشوند که کل عبارت را بسیار بسیار کوچک میکند.

۳ سوال ۳

۱.۳ الف

۱ ۱.۱.۳

علت این کار در واقع این است که ما در فرآیند منظم سازی، وزن های نسبت داده شده به خروجی های توابع فعال ساز را محدود میکنیم تا مدلی که بدست آورده ایم overfit نشود و این طور نباشد که مقدار آن برای تمام خروجی ها عدد بسیار بزرگی باشد. به جای آن با محدود کردن ضرایب سعی میکنیم به مدل بفهمانیم که بهتر است از ضرایب کمتر و کوچکتری استفاده کند تا مدل بهتری بدست بیاید اما ذات بایاس اینگونه است که صرفا بردار را در فضا جا به جا میکند و تفاوتی در overfit شدن و نشدن ندارد.

۲ ۲.۱.۳

برای مشخص تر شدن قضیه ابتدا عبارت زیر را در نظر بگیرید:

$$L1 = w^{(t+1)} = w^{(t)} - \epsilon(\Delta E(w) + \lambda)$$

حال وقتی که یک ضریب به صفر نردیک میشود مقدار $\epsilon\lambda$ ثابت است که باعث میشود دوباره به صفر بیشتر نزدیک شود و نهایتا سعی میکند که تک تک وزن ها را تا جای ممکن صفر کند.

۲.۳ ب

برای بدست آوردن تابع likelihood داریم:

$$\begin{aligned}\mathcal{L}(w \mid \mathbf{y}) &:= P(\mathbf{y} \mid w) \\ &= \prod_{i=1}^{n} P_Y\left(y_i \mid w, \sigma^2\right) \\ &= \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i-(w_0+w_1x_{i,1}+\ldots+w_px_{i,p}))^2}{2\sigma^2}}\end{aligned}$$

از طرفی posterior برابر است با:

$$\begin{aligned}\hat{\theta}_{\mathrm{MAP}} &= \arg\max_{\theta} P(\theta \mid y) \\ &= \arg\max_{\theta} \frac{P(y \mid \theta)P(\theta)}{P(y)} \\ &= \arg\max_{\theta} P(y \mid \theta)P(\theta) \\ &= \arg\max_{\theta} \log(P(y \mid \theta)P(\theta)) \\ &= \arg\max_{\theta} \log P(y \mid \theta) + \log P(\theta)\end{aligned}$$

پس با جایگذاری فرمول اول در posterior خواهیم داشت:

$$\begin{aligned}&\arg\max_w \left[\log \textstyle\prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i-(w_0+w_1x_{i,1}+\ldots+w_px_{i,p}))^2}{2\sigma^2}} + \log \prod_{j=0}^{p} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{w_j^2}{2\sigma^2}}\right] \\ &= \arg\max_w \left[-\sum_{i=1}^{n} \frac{(y_i-(w_0+w_1x_{i,1}+\ldots+w_px_{i,p}))^2}{2\sigma^2} - \sum_{j=0}^{p} \frac{w_j^2}{2\sigma^2}\right] \\ &= \arg\min_w \frac{1}{2\sigma^2}\left[\sum_{i=1}^{n} \left(y_i - (w_0 + w_1x_{i,1} + \ldots + w_px_{i,p})\right)^2 + \frac{\sigma^2}{\sigma^2}\sum_{j=0}^{p} w_j^2\right] \\ &= \arg\min_w \left[\sum_{i=1}^{n} \left(y_i - (w_0 + w_1x_{i,1} + \ldots + w_px_{i,p})\right)^2 + \lambda\sum_{j=0}^{p} w_j^2\right]\end{aligned}$$

که این همان منظم ساز L۲ است.

۳.۳ ج

در واقع فرایند نرمال سازی دسته ای در هنگام عملیات یادگیری ژرف برای کاهش shift covariate انجام میشود که در این عملیات میانگین مقادیر لایه خاص با توجه به میانگین و واریانس همان batch اپدیت میشود.

در حین فرآیند اموزش با توجه نوع داده ای که از شبکه میگذرد توزیع خروجی لایه ها با توجه به ورودی تغییر میکند که باعث میشود توزیع داده های جدید را یاد بگیرد و ما با انجام normalization لایه های شبکه را مجبور میکنیم که به نوعی توزیع خود را زیاد تغییر ندهند و به نوعی توزیع ها تقریبا یکسان میشود. در این حالت شبکه بسیار زودتر اموزش داده میشود.

در واقع به طور ساده میتوان گفت در حین فرآیند اموزش، توزیع خروجی لایه های فعال ساز میانی در هر مرحله تغییر میکند که به این فرایند shift covariance internal گویند. پس اگر جلوی این کار را بگیریم سرعت به شدت بالا میرود.

ولی مشکلی که دارد این است که وقتی سایز batch ها کوچک است میانگین و واریانس batch نماینده خوبی برای کل مجموعه نیستند که باعث میشود دقت کل شبکه پایین بیاید که نتیجه این است که در واقع تعداد داده ها به قدری بزرگ نبوده است که میانگین و واریانس batch بتواند estimation خوبی از میانگین و واریانس اصلی باشند.

۴.۳ د

ابتدا یک ساده سازی انجام دهیم و ببینیم چه میشود:

$$f_t = \nabla_\theta J(\theta_t)$$
$$m_t = b_1 m_{t-1} + (1 - b_1) f_t$$
$$m_0 = 0$$

سپس به روش بازگشتی مقادیر را بدست می‌آوریم:

$$m_1 = b_1 m_0 + (1 - b_1) f_1 = (1 - b_1) f_1$$
$$m_2 = b_1 m_1 + (1 - b_1) f_2 = b_1 (1 - b_1) f_1 + (1 - b_1) f_2$$
$$= (1 - b_1)(b_1 f_1 + f_2)$$
$$m_3 = b_1 m_2 + (1 - b_1) f_3$$
$$= (1 - b_1) \left(b_1^2 f_1 + b_1 f_2 + f_3 \right)$$
$$\dots$$
$$m_n = (1 - b_1) \sum_{i=1}^{n} b_1^{n-i} f_i$$

پس داریم:

$$E[m_n] = E\left[(1 - \beta_1) \sum_{i=1}^{n} \beta_1^{n-i} f_i \right]$$
$$= E[f_t] (1 - \beta_1) \sum_{i=1}^{n} \beta_1^{n-i} + \zeta$$
$$= E[f_t] (1 - \beta_1^n) + \zeta$$

حال چون مقدار $(1 - B_1^n)$ یک مقدار بسیار نزدیک به ۱ است و البته همیشه کوچکتر از یک است میتوان تقریبا عبارت های زیر را معادل دانست.

$$E[m_n] = E[f_t]$$

که چون m_t نزدیک به صفر است این دو تقریبا نزدیک به صفر خواهند بود. حال برای این که مقادیر بزرگ شوند یک راهکار این است که بر روی $(1 - B_1)$ تقسیم کنیم که اعداد نسبتا بزرگی تولید خواهد کرد پس دیگه مشکلی بابت میل به صفر نداریم.

۴ سوال ۴

۱.۴ الف

خب بصورت برداری لایه اول یا همون ورودی را X معرفی میکنیم.پس داریم:

$$Input = X = Z_0 = \begin{bmatrix}1 & 2\end{bmatrix}$$

$$A_1 = Z_0W_0 + B_0 = \begin{bmatrix}1 & 2\end{bmatrix}\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix} + \begin{bmatrix}0 & 0 & 0 & 0\end{bmatrix} = \begin{bmatrix}0 & 0 & 0 & 0\end{bmatrix}$$

$$Z_1 = \sigma(A_1) = \begin{bmatrix}\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2}\end{bmatrix}$$

$$A_2 = Z_1W_1 + B_1$$

$$= \begin{bmatrix}\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2}\end{bmatrix}\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix} + \begin{bmatrix}0 & 0 & 0\end{bmatrix} = \begin{bmatrix}0 & 0 & 0\end{bmatrix}$$

$$Z_2 = \sigma(A_2) = \begin{bmatrix}\frac{1}{2} & \frac{1}{2} & \frac{1}{2}\end{bmatrix}$$

$$A_3 = Z_2W_2 + B_2$$

$$= \begin{bmatrix}\frac{1}{2} & \frac{1}{2} & \frac{1}{2}\end{bmatrix}\begin{bmatrix}0\\ 0\\ 0\end{bmatrix} + \begin{bmatrix}0\end{bmatrix} = [0]$$

$$Z_3 = \sigma(A_3) = \begin{bmatrix}\frac{1}{2}\end{bmatrix} \Longrightarrow output = 1$$

خب به هر حال خروجی یا یک است یا صفر که ما فرض میکنیم ۱ باشد.
حال با توجه به فرمول زیر مقدار خطای هر لایه را حساب میکنیم:

$$Error = \frac{1}{2}(Predicted - Real)^2 = \frac{1}{2}\left(\frac{1}{2} - 1\right)^2 = \frac{1}{4}$$

$$\delta^L = \nabla_z L \odot \sigma'(a_L) = \frac{1}{4}([0] * (1 - [0]) = 0$$

و با توجه به این که میدانیم

$$\delta^l = \left(\left(w^{l+1}\right)^T \delta^{l+1}\right) \odot \sigma'\left(z^l\right)$$

پس متوجه میشویم که کل مشتق های کل وزن های دیگر صفر است زیرا تک تک مشتق ها به مشتق لایه بعدی خود نیاز دارند که با توجه به این که مشتق آخرین لایه صفر است تا اولین لایه یک صفر در همه ضرب میشود که باعث میشود کل تغییرات ضرایب صفر شود. پس هیچ اتفاقی در شبکه نداریم! البته فرمول بالا در کتاب های کلاسیک یادگیری ماشین اثبات میشود و من نیازی به اثبات آن ندیدم.

۲.۴ ب

$$L(y, y') = -y^T log\hat{y} = \sum_{i=1}^{d} -y_i log\widehat{y_i}$$

$$\frac{\partial L}{\partial \hat{y}_i} = -\frac{y_i}{\hat{y}_i}$$

$$\hat{y} = softmax(u)$$

ابتدا قدری در مورد سافت مکس توضیح بدهیم:

$$\sigma(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}}$$

برای بدست آوردن مشتق آن داریم:

$$\frac{\partial \sigma(\mathbf{x})}{\partial \mathbf{x}} = \frac{\partial \sigma(x)_k}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\frac{e^{x_k}}{\sum_{j=1}^{N} e^{x_j}}\right)$$

که میتوان نوشت:

$$f = e^{x_k}$$

$$g = \sum_{j=1}^{N} e^{x_j}$$

$$z = x_i$$

$$\frac{\partial f}{\partial z} = \frac{\partial\left(e^{x_k}\right)}{\partial x_i} = e^{x_k}\frac{\partial x_k}{\partial x_i} = e^{x_k}\delta_{ik}$$

$$\frac{\partial g}{\partial z} = \sum_{j=1}^{N} \frac{\partial e^{x_j}}{\partial x_i} = \sum_{j=1}^{N} e^{x_j}\frac{\partial x_j}{\partial x_i}$$

$$= e^{x_1}\frac{\partial x_1}{\partial x_i} + e^{x_2}\frac{\partial x_2}{\partial x_i} + \cdots + e^{x_N}\frac{\partial x_N}{\partial x_i} = e^{x_i} \text{ any for } i$$

$$\frac{\partial f}{\partial z} f - \frac{\partial g}{\partial z} f_j = \frac{e^{x_k}\delta_{ik}\sum_{j=1}^{N} e^{x_j} - e^{x_i}e^{x_k}}{\left[\sum_{j=1}^{N} e^{x_j}\right]^2}$$

پس داریم:

$$\frac{\partial \sigma(x)_k}{\partial x_i} = \frac{e^{x_k}\delta_{ik}\sum_{j=1}^{N} e^{x_j} - e^{x_i}e^{x_k}}{\left[\sum_{j=1}^{N} e^{x_j}\right]^2}$$

$$= \frac{e^{x_k}}{\sum_{j=1}^{N} e^{x_j}}\delta_{ik}\frac{\sum_{j=1}^{N} e^{x_j}}{\sum_{j=1}^{N} e^{x_j}} - \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}\frac{e^{x_k}}{\sum_{j=1}^{N} e^{x_j}}}$$

$$= \sigma(\mathbf{x})_k\delta_{ik} - \sigma(\mathbf{x})_i\sigma(\mathbf{x})_k$$

$$= \sigma(\mathbf{x})_k\left(\delta_{ik} - \sigma(\mathbf{x})_i\right)$$

که نهایتا میتوان همچنین نتیجه گیری را انجام داد. البته با توجه به این که سوال نوشته مراحل را با جزئیات نشان دهید مطمئن نیستم بدست آوردن مشتق تابع softmax نیز خواسته سوال بوده است یا خیر.

$$\frac{\partial \sigma_k}{\partial x_i} = \sigma_k\left(\delta_{ik} - \sigma_i\right)$$

در بقیه سوال نیز داریم

$$u = W'^T h$$
$$\frac{\partial u}{\partial h} = W'$$
$$h = W^T x$$
$$\frac{\partial h}{\partial x} = W$$

و البته با توجه به قاعده زنجیری میتوان نوشت که:

$$\frac{\partial L}{\partial u} = \frac{\partial L}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial u}$$

$$\frac{\partial L}{\partial h} = \frac{\partial L}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial u} \times \frac{\partial u}{\partial h}$$

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial u} \times \frac{\partial u}{\partial h} \times \frac{\partial h}{\partial x}$$