به نام خدا

یادگیری عمیق، تکلیف ۱

مهدی کافی ۹۹۲۱۰۷۵۳

مسئله ۱. Linear Regression (۱۵ نمره)

مجموعه دادگان $S = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$ را در نظر بگیرید به گونه‌ای که $x \in \mathbb{R}^d$ و $y \in \mathbb{R}$ است. برای تخمین برچسب یک نمونه x از رابطه رگرسیون خطی به صورت $\hat{y} = \sum_{j=1}^{d} w_j x_j + b$ و برای آموزش مدل از تابع هزینه SSE به صورت $\mathcal{L}(w,b) = \frac{1}{2n}\sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2$ استفاده می‌کنیم. حال به سوالات زیر پاسخ دهید.

الف) با استفاده از گرادیان کاهشی رابطه به‌روز رسانی برای وزن‌ها ارائه دهید.

ب) حال رابطه محاسبه وزن‌ها را به صورت فرم بسته ارائه دهید.

ج) با فرض اینکه گرادیان کاهشی بعد از m به‌روز رسانی به پاسخ بهینه می‌رسد، این دو روش را از نظر مرتبه زمانی با یک‌دیگر مقایسه کنید.

الف) برای حل این بخش در ابتدا رابطه تابع هزینه را باز می‌کنیم و سعی می‌کنیم که برای یکی از وزن‌ها و بایاس مشتق را محاسبه کنیم. محاسبات به صورت زیر خواهدبود.

$$L = \frac{1}{2n}\sum_{i}^{n} (y^i - \hat{y}^i)^2 \qquad \hat{y} = \sum_{j}^{d} w_j x_j + b$$

$$L = \frac{1}{2n}\left[(y^1-\hat{y}^1)^2 + (y^2-\hat{y}^2)^2 + \cdots + (y^n-\hat{y}^n)^2\right]$$

$$= \frac{1}{2n}\left[(y^1 - \sum_{j}^{d} w_j x_j^1 - b)^2 + (y^2 - \sum_{j}^{d} w_j x_j^2 - b)^2 + \cdots + (y^n - \sum_{j}^{d} w_j x_j^n - b)^2\right]$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial}{\partial w_1}\frac{1}{2n}\left[(y^1 - \sum_{j}^{d} w_j x_j^1 - b)^2 + (y^2 - \sum_{j}^{d} w_j x_j^2 - b)^2 + \cdots + (y^n - \sum_{j}^{d} w_j x_j^n - b)^2\right]$$

$$= \frac{1}{2n}\left[2\times(y^1-\hat{y}^1)(-x_1^1) + 2\times(y^2-\hat{y}^2)(-x_1^2) + \cdots + 2\times(y^n-\hat{y}^n)(-x_1^n)\right]$$

$$= \frac{1}{n}\sum_{i=1}^{n}(\hat{y}^i - y^i)(x_1^i)$$

$$\frac{\partial L}{\partial b} = \frac{1}{2n}\left[2(y^1-\hat{y}^1)(-1) + 2(y^2-\hat{y}^2)(-1) + \cdots + 2(y^n-\hat{y}^n)(-1)\right]$$

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

حال می‌توانیم محاسبات انجام شده برای یک وزن را به صورت زیر به تمامی وزن‌ها تعمیم دهیم.

$$\frac{\partial L}{\partial w} = \frac{1}{n}\left[\sum_{i=1}^{n} (\hat{y}^i - y^i)(x_1^i), \sum_{i=1}^{n} (\hat{y}^i - y^i)(x_2^i), \dots, \sum_{i=1}^{n} (\hat{y}^i - y^i)(x_d^i) \right]$$

$$= \frac{1}{n}\left[\hat{y}^1 - y^1, \hat{y}^2 - y^2, \dots, \hat{y}^n - y^n\right] \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_d^1 \\ x_1^2 & x_2^2 & \dots & x_d^2 \\ \vdots & & & \\ x_1^n & x_2^n & \dots & x_d^n \end{bmatrix} = \frac{1}{n}(\hat{y} - y)^T X$$

و سپس با داشتن گرادیان‌ها می‌توانیم وزن‌ها و بایاس را به صورت زیر در جهت عکس گرادیان تغییر دهیم.

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

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

ب)محاسبات برای این بخش به صورت زیر خواهندبود.

$\hat{y} = Xw \qquad L = \frac{1}{2n} \sum (y^i - \hat{y}^i)^2 = \frac{1}{2n} (y - Xw)^T (y - Xw)$

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

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

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

*) $\dim(y) = (n \times 1) \quad \dim(X) = (n \times d) \quad \dim(w) = (d \times 1)$

$\rightarrow w^T X^T X w - y^T X w - w^T X^T y$

$(1 \times d)@(d \times n)@(n \times d)@(d \times 1) - (1 \times n)@(n \times d)@(d \times 1) - (1 \times d)@(d \times n)@(n \times 1) = \text{Scalar}$

بنابراین می‌توانیم عبارت بالا را با trace خودش جایگزین کنیم

$\nabla_w L = \frac{1}{2n} \nabla_w \, tr(w^T X^T X w - y^T X w - w^T X^T y)$

$tr(A+B) = tr(A) + tr(B), \; tr A^T = tr A$

$\Rightarrow \nabla_w L = \frac{1}{2n} \nabla_w (tr\, w^T X^T X w - 2\, tr\, y^T X w)$

$\nabla_A \, tr AB = B^T, \; \nabla_{A^T} f(A) = (\nabla_A f(A))^T$

$\Rightarrow \nabla_w L = \frac{1}{2n} [\nabla_w (tr\, w^T X^T X w) - 2 X^T y]$

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

$$= \frac{1}{n}(X^T X w - X^T y)$$

closed-form: $\nabla_w L = 0 \Rightarrow X^T X w = X^T y \Rightarrow w = (X^T X)^{-1} X^T y$

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

ج) اگر فرض بگیریم که ماتریس نمونه‌ها ابعادی برابر با $(n * d)$ داشته باشد. که n تعداد نمونه‌ها و d تعداد ویژگی‌ها باشد. برای حالت فرم بسته که به صورت $(X^T X)^{-1} X^T y$ است. نیاز داریم در ابتدا ضرت ماتریس X^T در X را انجام دهیم. با توجه به ابعاد ماتریس‌ها این ضرب هزینه‌ای از مرتبه $O(nd^2)$ خواهدداشت. سپس ماتریسی با ابعاد $(d * d)$ خواهیم داشت و هزینه وارون کردن این ماتریس (اگر وارون پذیر باشد)، از مرتبه $O(d^3)$ خواهدبود. هزینه ضرب ماتریس X^T در بردار y نیز از مرتبه $O(nd)$ است و در نهایت هزینه ضرب ماتریس وارون محاسبه شده با ابعاد $(d * d)$ در بردار $(d * 1)$ از مرتبه $O(d^2)$ خواهدبود. در مجموع هزینه محاسبه مقدار بهینه بردار وزن‌ها به صورت زیر است.

$$O(nd^2 + d^3 + nd + d^2) \;=\; O(nd^2 + d^3) \;=\; O(d^2(n + d))$$

در حالتی که تعداد ویژگی‌ها از ۵۰۰۰ بیشتر شود، محاسبه مقدار بهینه بردار وزن‌ها از این طریق تقریبا غیر ممکن می‌شود.
در حالت گرادیان کاهشی، نیاز داریم که محاسبه $w^T x_i$ را انجام دهیم که مرتبه $O(d)$ دارد. سپس محاسبات مشتق‌ها برای m به روز رسانی مرتبه $O(nm)$ خواهدداشت. در مجموع هزینه گرادیان کاهشی از مرتبه $O(dmn)$ خواهدبود.

مسئله ۲. Activation Function (۱۰ نمره)

به سوالات زیر پاسخ دهید.

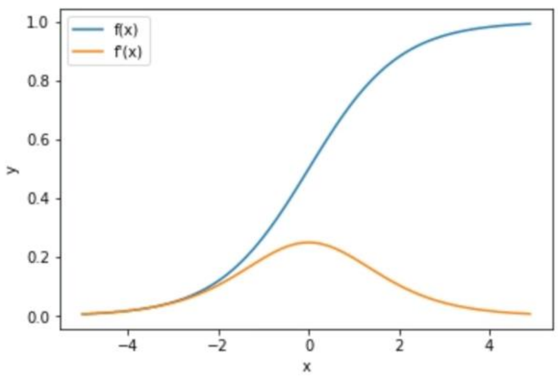
الف) ابتدا مشکل vanishing gradient را توضیح دهید. سپس توابع فعالسازی ReLU و sigmoid با بررسی مشتق از این نظر مقایسه کنید.

ب) ابتدا مقداردهی Xavier را توضیح دهید و سپس بررسی کنید که چگونه به مشکل محو شدن کمک می‌کند.

ج) فرایند آموزش یک شبکه عصبی با تابع فعال‌سازی sigmoid را در صورتی که مقداردهی اولیه وزن‌ها بزرگ است، بررسی کنید.

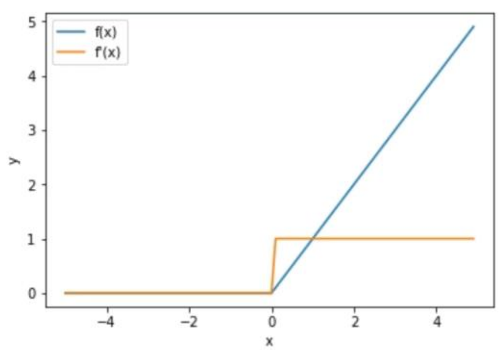
الف) هنگامیکه تعداد لایه‌ها در شبکه عصبی اضافه شوند و این لایه‌ها از توابع فعال سازی خاصی مثل sigmoid استفاده کنند باعث می‌شود که مقدار گرادیان در هنگام back propagation لایه به لایه کوچک و کوچکتر شود و در عمل گرادیانی که به لایه‌های اولیه شبکه (که لایه‌های بسیار مهمی برای تشخیص ویژگی‌های ورودی است) می‌رسد بسیار نزدیک به صفر می‌شود و وزن‌ها تغییر نمی‌کنند و آموزشی صورت نمی‌گیرد.

تابع sigmoid: این تابع ورودی با هر دامنه‌ای را به دامنه نسبتا کوچک بین صفر و یک نگاشت می‌کند و باعث می‌شود مشتق این تابع کوچک شود. به علاوه همانطور که در شکل زیر دیده می‌شود، هنگامیکه مقدار ورودی این تابع بزرگ یا کوچک شود، مقدار مشتق آن بسیار کوچک و نزدیک به صفر خواهدشد که مشکل vanishing gradient را ایجاد می‌کند. حتی اگر مقدار ورودی هم در محدوده مناسبی باشد، تعداد لایه‌های زیاد باز هم مشکل vanishgin gradient را ایجاد می‌کند.



Sigmoid gradient

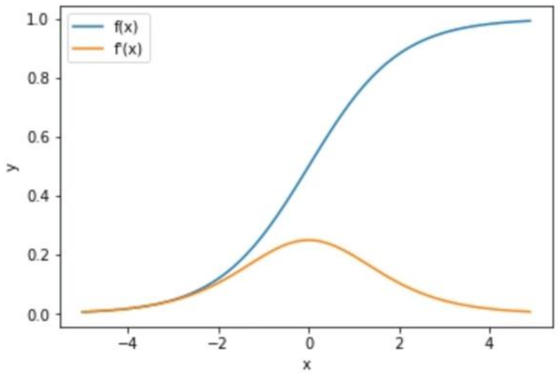
تابع ReLU: این تابع همانطور که در شکل زیر دیده می‌شود، مقدار گرادیانی برابر با صفر و یا یک دارد که باعث می‌شود در هنگامیکه ورودی تابع از صفر بیشتر باشد، مشتق یک شود و هر تعداد لایه هم داشته‌باشیم، مشتق‌هایی برابر با یک در یکدیگر ضرب می‌شوند که مشکل vanishing ایجاد نمی‌کنند زیرا که ضرب تمام یک‌ها باز هم یک خواهدشد و در عین حال باز هم خاصیت غیر خطی بودن را دارد. مشکل این تابع این است که برای مقادیر ورودی کمتر مساوی صفر گرادیان صفر ایجاد می‌کند و باعث می‌شود خطایی که در مرحله back propagation در حال منتشر شدن است ناگهان صفر شود و فرآیند آموزش را تحت تاثیر قرار می‌دهد.



ReLU activation and first derivative

ب) مقدار دهی Xavier سعی می‌کند که واریانس خروجی و ورودی یک نورون را ثابت نگه دارد. به این منظور در حالت استاندارد، وزن‌های یک نورون را از توزیع یکنواخت $W \sim U[-\frac{1}{\sqrt{n}},\frac{1}{\sqrt{n}}]$ در حالیکه n تعداد نورون‌های لایه ورودی (قبلی) است، مقدار دهی می‌کند. در حالت نرمال شده نیز وزن نورون را از توزیع یکنواخت $W \sim U[-\frac{\sqrt{6}}{\sqrt{n+m}},\frac{\sqrt{6}}{\sqrt{n+m}}]$ مقدار دهی می‌کند و m تعداد نورون‌های خروجی این لایه (لایه فعلی) است. حالت نرمال شده باعث می‌شود گرادیان‌های برگشتی نیز واریانس ثابتی داشته‌باشند. معمولا در عمل به این صورت عمل می‌شود که از تابع ReLU و وزن‌های انتخاب شده از توزیع نرمال $W \sim N(0,\sqrt{\frac{2}{n}})$ استفاده می‌شود و n تعداد نورون‌های لایه ورودی است. نگه داشتن واریانس ورودی و خروجی باعث می‌شود که به طور مثال تابع sigmoid در نواحی اشباع قرار نگیرد و مشتق آن از بین نرود.

ج) نمودار تابع sigmoid و مشتق آن در زیر آمده‌است.



Sigmoid gradient

اگر وزن‌ها با مقادیر بزرگی (چه منفی و چه مثبت) مقدار دهی شوند، خروجی تابع sigmoid در نواحی اشباع شده قرار می‌گیرد و همانطور که دیده می‌شود، مقدار گرادیان در این نواحی بسیار کوچک و نزدیک به صفر است و باعث کند شدن یادگیری می‌شود.

مسئله ۳. Regularization & Optimization (۲۰ نمره)

به سوالات زیر پاسخ دهید.

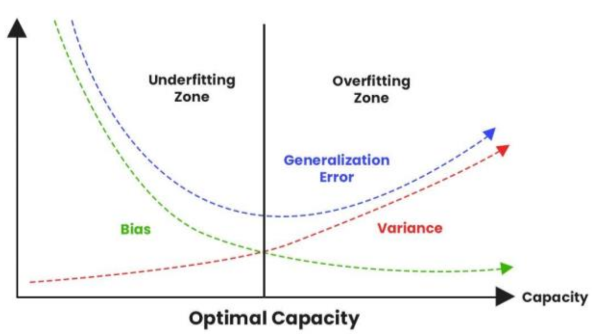
الف) چرا منظم‌ساز L_2 معمولا روی Bias شبکه اعمال نمی‌شود؟ همچنین توضیح دهید چرا منظم‌ساز L_1 منجر به صفر شدن برخی از وزن‌ها می‌شود؟

ب) مسئله رگرسیون خطی بر روی n داده با تابع هزینه $\mathcal{L}(w) = \frac{1}{2n}\sum_{i=1}^{n}\left(y^{(i)} - x^{(i)^T}w\right)^2$ را در نظر بگیرید. اثبات کنید اضافه کردن نویز از توزیع $\mathcal{N}(0, \sigma^2 I)$ به داده‌های ورودی معادل استفاده از منظم‌ساز L_2 در تابع هزینه است.

ج) توضیح دهید که Batch Normalization چگونه فرایند آموزش را سرعت می‌بخشد. همچنین توضیح دهید زمانی که سایز batch کوچک باشد، استفاده از Batch Normalization چه تاثیری در فرایند آموزش دارد.

د) رابطه گشتاور اول بهینه‌ساز آدام را به صورت $m_t = \beta_1 m_{t-1} + (1-\beta_1)\nabla_\theta J(\theta_t)$ در نظر بگیرید. نشان دهید چرا مقادیر m_t گرایش به صفر دارند و چرا مقدار $\widehat{m}_t$ که به صورت $\widehat{m}_t = \frac{m_t}{1-\beta_1^t}$ محاسبه می‌شود، با این مشکل روبرو نمی‌شود.

الف)



با توجه به شکل بالا، می‌دانیم در هنگام آموزش مدل‌ها به دو نکته توجه می‌کنیم؛ بایاس و واریانس که بایاس به معنای فاصله‌ای است که تابع تخمین زده شده ما از تابع اصلی دارد و واریانس به معنای فاصله‌ای است که خروجی تابع از داده‌های متفاوت تابع اصلی دارد. حال اگر مدل ما بسیار پیچیده شود یا به بیان دیگر وزن‌های بسیار زیادی داشته‌باشد در واقع به داده‌های آموزش بسیار نزدیک می‌شویم و باعث overfitting

می‌شود. حال از منظم سازی استفاده می‌کنیم که واریانس را کاهش دهیم و مدل general شود. بنابراین منظم سازی نیاز دارد که مقدار وزن‌ها را کاهش دهد، بنابراین منظم ساز 2L بر روی وزن‌ها اعمال می‌شود و نه بر روی مقدار بایاس.

منظم ساز 1L به این صورت عمل می‌کند که مجموع قدر مطلق وزن‌ها را با یک ضریب مثل λ به تابع هزینه اضافه می‌کند. اگر این ضریب صفر باشد که regularizaiton نداریم و اگر مقدار آن زیاد باشد باعث می‌شود که بسیاری از وزن‌ها که مربوط به ویژگی‌هایی هستند که در اصل به خروجی ربطی ندارند را صفر کند و فقط وزن‌های ویژگی‌های اصلی را نگه می‌دارد. معمولا در شرایطی از این منظم ساز استفاده می‌شود که ویژگی‌های بسیار زیادی داریم.

ب) اگر فرض بگیریم که می‌خواهیم مقدار وزن β را با استفاده از داده‌های ورودی x , y که به ترتیب ورودی و خروجی هستند به دست آوریم.

اگر فرض بگیریم که ورودی و خروجی رابطه خطی دارند با اضافه کردن مقدار نویز گاوسی به رابطه زیر خواهیم‌رسید.

$$y = \beta x + \epsilon$$

حال با توجه به اینکه ϵ از توزیع گاوسی آمده‌است، می‌توانیم از روش Maximum Likelihood برای محاسبه β استفاده کنیم، بنابراین سعی می‌کنیم مقدار زیر را بیشینه کنیم.

$$\prod_{n=1}^{N} N\left(y_n \mid \beta x_n,\, \sigma^2\right)$$

حال اگر β را از توزیع نرمال $N(0, \lambda^{-1})$ بدانیم که معادل اضافه کردن نویز گاوسی به داده‌های ورودی است. عبارت به صورت زیر خواهدشد که λ معیاری است که مشخص می‌کند چقدر β به میانگین خودش یعنی صفر نزدیک‌تر باشد.

$$\prod_{n=1}^{N} N\left(y_n \mid \beta x_n,\, \sigma^2\right) N\left(\beta \mid 0,\, \lambda^{-1}\right)$$

سپس اگر لگاریتم عبارت بالا را محاسبه کنیم (کاری که معمولا برای محاسبه Maximum Likelihood انجام می‌دهیم) به عبارت زیر خواهیم‌رسید.

$$\sum_{n=1}^{N} -\frac{1}{\sigma^2}\left(y_n - \beta x_n\right)^2 - \lambda \beta^2 + c$$

عبارت بالا بسیار شبیه به تابع هزینه با منظم ساز 2L است با این تفاوت که به دنبال بیشینه کردن عبارت بالا هستیم و با منفی کردن عبارت بالا دقیقا به تابع هزینه SSE با منظم ساز 2L خواهیم رسید که به دنبال کمینه کردن آن هستیم.

ج) Batch Normalization در هر لایه سعی می‌کند که خروجی تابع فعال سازی را نرمال کند. این کار باعث می‌شود که internal covariate shift کاهش یابد و سرعت یادگیری افزایش خواهدیافت. حال چطور این اتفاق میافتد؟ فرض کنیم که یک لایه ورودی X دارد و خروجی Y را تولید می‌کند لایه بعدی Y را گرفته و Z را تولید می کند، حال در هنگام یادگیری این لایه‌ها مستقل از یکدیگر سعی می‌کنند وزن و بایاس خود را برای خروجی نهایی و تابع هزینه بهینه کنند و فرض کنیم که لایه اول با تغییر وزن و بایاس خود خروجی را به A تبدیل کند که توزیع متفاوتی از Y دارد. حال لایه بعدی باید آموزش را ابتدا شروع کند زیرا با داده‌های جدید و متفاوتی روبرو شده‌است. حال اگر از BN استفاده کنیم توزیع A و Y تفاوت آنچنانی نخواهندداشت و لایه دوم نیز سریعتر می‌تواند خودش را بهبود ببخشد و آموزش در مجموع سریعتر می‌شود.

هنگامیکه اندازه batch را افزایش می‌دهیم باعث می‌شود که مقادیر میانگین و واریانس محاسبه شده روی batch برای BN به مقادیر واقعی آنها روی تمامی داده‌ها نزدیک‌تر شود و همینطور گرادیان‌های محاسبه شده نیز جهت بهتری خواهندداشت و بیشتر به سمت کمینه حرکت می‌کنند.

د) در ابتدا مقدار m_t را صفر قرار می‌دهیم و سپس شروع به بسط دادن این مقدار برای tهای بیشتر می‌کنیم.

$$m_0 = 0$$
$$m_1 = \beta m_0 + (1-\beta)\nabla J_1 = (1-\beta)\nabla J_1$$
$$m_2 = \beta(1-\beta)\nabla J_1 + (1-\beta)\nabla J_2$$
$$m_3 = \beta^2(1-\beta)\nabla J_1 + \beta(1-\beta)\nabla J_2 + (1-\beta)\nabla J_3$$
$$\vdots$$
$$m_n = \beta^{n-1}(1-\beta)\nabla J_1 + \beta^{n-2}(1-\beta)\nabla J_2 + \cdots + (1-\beta)\nabla J_n$$
$$m_n = (1-\beta)\sum_{i=1}^{n} \beta^{n-i}\nabla J_i$$

مقدار β معمولا برابر با 0.9 قرار می‌گیرد. بنابراین می‌بینیم که مقدار کمتر از ۱ β در حال ضرب شدن در خودش است و این مقدار به سمت صفر میل می‌کند.

در حالتی که از فرمول $\frac{m_t}{1-\beta^t}$ استفاده کنیم به ازای مقادیر کمتر t با توجه به اینکه β مقدار کمتر از یک دارد، m_t در مقدار بزرگی ضرب می‌شود و برای tهای بزرگتر، مقدار m_t در عدد کوچکتری ضرب می‌شود تا تمامی گرادیان‌ها در محاسبه $\widehat{m_t}$ نقش داشته‌باشند.

مسئله ۴. Backpropagation (۲۰ نمره)

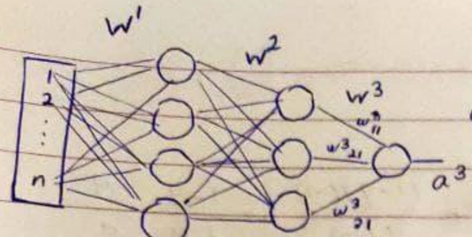
الف) یک شبکه عصبی feedforward را با دولایه نهان با تابع فعال‌سازی sigmoid برای مسئله دسته‌بندی دودویی در نظر بگیرید. لایه اول نهان شامل ۴ نورون و لایه دوم شامل ۳ نورون است. ابعاد ورودی دلخواه در نظر گرفته می‌شود. در ابتدا تمامی وزن‌ها و بایاس شبکه صفر مقداردهی می‌شوند. به‌ازای یک ورودی، شبکه چه مقداری را خروجی دهد؟ بعد از یک بار به‌روز رسانی وزن‌ها با استفاده از SGD، بررسی کنید مقادیر وزن‌ها چگونه تغییر می‌کند.

ب) شبکه زیر را در نظر بگیرید.

$$h = W^T x$$
$$u = W'^T h$$
$$\hat{y} = Softmax(u)$$
$$\mathcal{L}(W, W') = -y^T \log \hat{y}$$

که در آن $x \in \mathbb{R}^d$ و $h \in \mathbb{R}^H$ و $\hat{y} \in \mathbb{R}^d$ و y برچسب *one-hot* شده است. با در نظر گرفتن تابع هزینه برای یک نمونه ورودی، با استفاده از روش گرادیان کاهشی روابط به‌روز رسانی وزن‌ها را با نوشتن جزئیات مراحل بنویسید.

الف)



w_{ij}^{l} : وزن سلول l از نورون i به j

$x \in R^n \quad \dim(W^1) = (4 \times n) \quad \dim(W^2) = (3 \times 4) \quad \dim(W^3) = (1 \times 3)$

$$Z^1 = \begin{bmatrix} w_{11}^1 x_1 + w_{21}^1 x_2 + \cdots + w_{n1}^1 x_n + b_1^1 \\ w_{12}^1 x_1 + w_{22}^1 x_2 + \cdots + w_{n2}^1 x_n + b_2^1 \\ \vdots \qquad\qquad \vdots \\ w_{14}^1 x_1 + w_{24}^1 x_2 + \cdots + w_{n4}^1 x_4 + b_4^1 \end{bmatrix} = W^1 \times x + b^1 \xrightarrow[W,b=0]{} z^1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$a^1 = \begin{bmatrix} \text{Sigmoid}(z_1^1) \\ \vdots \\ \text{Sigmoid}(z_4^1) \end{bmatrix} = \text{Sigmoid}(z^1) \quad \dim(a^1) = (4 \times 1) \Rightarrow a^1 = [0.5, 0.5, 0.5, 0.5]^T$$

$$Z^2 = \begin{bmatrix} w_{11}^2 a_1^1 + w_{21}^2 a_2^1 + \cdots + w_{41}^2 a_4^1 + b_1^2 \\ \vdots \\ w_{13}^2 a_1^1 + w_{23}^2 a_2^1 + w_{33}^2 a_3^1 + w_{43}^2 a_4^1 + b_3^2 \end{bmatrix} = W^2 \times a^1 + b^2 \xrightarrow[W,b=0]{} z^2 = [0, 0, 0]^T$$

$$a^2 = \text{Sigmoid}(z^2) = [0.5, 0.5, 0.5]^T$$

$$z^3 = W^3 \times a^2 + b^3 = 0 \Rightarrow a^3 = \text{Sigmoid}(0) = 0.5 \longleftarrow \text{خروجی شبکه}$$

s.a.m

$$\frac{\partial L}{\partial z^3} = \frac{\partial L}{\partial a^3} \times \frac{\partial a^3}{\partial z^3}$$

$$\frac{\partial L}{\partial a^3} = \frac{\partial}{\partial a} -(y\log a + (1-y)\log(1-a)) = -\left(\frac{y}{a} + \frac{(1-y)(-1)}{(1-a}\right) = \frac{1-y}{1-a} - \frac{y}{a}$$

$$\frac{\partial a^3}{\partial z^3} = \sigma(z^3)(1-\sigma(z^3)) = a(1-a)$$

$$\Rightarrow \frac{\partial L}{\partial z^3} = a^3 - y$$

* اگر فرض کنیم تمامی لیبل برابر با صفر یا یک باشد مقدار $\frac{\partial L}{\partial z^3}$ برابر با 0.5 یا 0.5- خواهد بود.

$$\frac{\partial L}{\partial w^3} = \frac{\partial L}{\partial z^3} \cdot \frac{\partial z^3}{\partial w^3} = (a^3 - y)\, a^2 = (0.5) * [0.5, 0.5, 0.5]^T \quad w^3 = \begin{bmatrix} -0.5 * 0.5 \\ -0.5 * 0.5 \\ -0.5 * 0.5 \end{bmatrix}$$

$$\frac{\partial L}{\partial b^3} = \frac{\partial L}{\partial z^3} \cdot \frac{\partial z^3}{\partial b^3} = (a^3 - y)\, I = [0.5, 0.5, 0.5]^T \quad b^3 = \begin{bmatrix} -0.5 \\ -0.5 \\ -0.5 \end{bmatrix}$$

$$\frac{\partial L}{\partial w^2} = \frac{\partial L}{\partial z^3} \cdot \frac{\partial z^3}{\partial a^2} \cdot \frac{\partial a^2}{\partial z^2} \cdot \frac{\partial z^2}{\partial w^2}$$

$$\Rightarrow w^3 \xrightarrow[w=0]{} \text{gradient} = 0 \Rightarrow$$

پس چون وزن‌های W^2, b^2 آپدیت نمی‌شوند

$$\frac{\partial L}{\partial b^2} = \frac{\partial L}{\partial z^3} \cdot \frac{\partial z^3}{\partial a^2} \cdot \frac{\partial a^2}{\partial z^2} \cdot \frac{\partial z^2}{\partial b^2}$$

* همین اتفاق برای وزن‌ها و بایاس W^1, b^1 نیز خواهد افتاد و آن‌ها نیز تغییر نمی‌کنند.

ب)

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

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

$$\Rightarrow \frac{\partial \hat{y}_i}{\partial u_j} = \hat{y}_i \frac{\partial}{\partial u_j} \log(\hat{y}_i) = \hat{y}_i \frac{\partial}{\partial u_j} \left(u_i - \log\left(\sum_{l=1}^{\hat{}} e^{z_l} \right)\right) = \hat{y}_i \left(\frac{\partial u_i}{\partial u_j} - \frac{\partial}{\partial u_j} \log\left(\sum_{l=1}^{\hat{}} e^{z_l} \right)\right)$$

$$= \hat{y}_i \left(\mathbb{1}\{i=j\} - \frac{e^{z_j}}{\sum_{l=1}^{f} e^{z_l}} \right) = (1\{i=j\} - s_j)\hat{y}_i =$$

$$\frac{\partial L}{\partial u_i} = -\frac{\partial}{\partial u_i} \sum_{i=1}^{d} y_i \cdot \log(\hat{y}_i) = -\sum_{i=1}^{d} \frac{y_i}{\hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial u_j} = -\sum_{i=1}^{d} y_i \cdot (\mathbb{1}\{i=j\} - \hat{y}_j)$$

$$= \sum_{i=1}^{d} y_i \hat{y}_j - y_j = \hat{y}_j \times \sum_{i=1}^{d} y_i - y_j = \hat{y}_j - y_j \Rightarrow \frac{\partial L}{\partial u} = \boxed{\hat{y} - y}$$

$$\Rightarrow \frac{\partial L}{\partial W'} = \frac{\partial L}{\partial u} \times \frac{\partial u}{\partial W'} = h(\hat{y} - y)^T = h(\hat{y} - y)^T \xrightarrow{dim} (h \times d)$$

$$\Rightarrow \frac{\partial L}{\partial W} = \frac{\partial L}{\partial u} \times \frac{\partial u}{\partial h} \times \frac{\partial h}{\partial W} = (\hat{y} - y^T)(x)^T W'^T \xrightarrow{dim} (d \times h)$$

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

$$W' := W' - \alpha \frac{\partial L}{\partial W'}$$