بسم الله الرحمن الرحیم

دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

یادگیری عمیق

نیم‌سال دوم ۰۴-۰۳

مدرس: مهدیه سلیمانی

تمرین اول — ددلاین تمرین : ۱۹ اسفند

- برای ارسال هر تمرین تا ساعت ۲۳:۵۹ روز ددلاین فرصت دارید. مهلت تاخیر (مجاز و غیر مجاز) برای این تمرین، ۷ روز است (یعنی حداکثر تاریخ ارسال تمرین ۲۶ اسفند است)
- در هر کدام از سوالات، اگر از منابع خارجی استفاده کرده‌اید باید آن را ذکر کنید. در صورت همفکری با افراد دیگر هم باید نام ایشان را در سوال مورد نظر ذکر نمایید.
- پاسخ تمرین باید ماحصل دانسته‌های خود شما باشد. در صورت رعایت این موضوع، استفاده از ابزارهای هوش مصنوعی با ذکر نحوه و مصداق استفاده بلامانع است.
- پاسخ ارسالی واضح و خوانا باشد. در غیر این صورت ممکن است منجر به از دست دادن نمره شود.
- پاسخ ارسالی باید توسط خود شما نوشته شده باشد. به اسکرین‌شات از منابع یا پاسخ افراد دیگر نمره‌ای تعلق نمی‌گیرد.
- در صورتی که بخشی از سوال‌ها را جای دیگری آپلود کرده و لینک آن را قرار داده باشید، حتما باید تاریخ آپلود مشخص و قابل اتکا باشد.
- محل بارگذاری سوالات نظری و عملی در هر تمرین مجزا خواهد بود. به منظور بارگذاری بایستی تمارین تئوری در یک فایل pdf با نام HW1_[First-Name]_[Last-Name]_[Student-Id].pdf و تمارین عملی نیز در یک فایل مجزای زیپ با نام HW1_[First-Name]_[Last-Name]_[Student-Id].zip بارگذاری شوند.
- در صورت وجود هرگونه ابهام یا مشکل، در کوئرای درس آن مشکل را بیان کنید و از پیغام دادن مستقیم به دستیاران آموزشی خودداری کنید.
- طراحان این تمرین : پیام تائبی-رامتین مسلمی-امیرمهدی میقانی-کسری ملیحی

بخش نظری (۱۰۰ نمره)

پرسش ۱. Matrix Differentiation (۲۰ نمره)

برای یک تابع $f : \mathbb{R}^n \to \mathbb{R}^m$ می‌توان مشتق آن را به ازای یک ورودی نظیر $x \in \mathbb{R}^n$ به صورت زیر تعریف کرد:

$$J_{i,j} = \frac{\partial f_i}{\partial x_j}, \quad J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

اولین نکته قابل توجه این است که این ماتریس $J \in \mathbb{R}^{m \times n}$ بوده و سطرهای آن ترانهاده گرادیان هر یک از ابعاد خروجی نسبت به ورودی می‌باشند. برخی منابع ترانهاده این ماتریس را به عنوان مشتق در نظر می‌گیرند و شما باید همواره به این نکته دقت داشته باشید. در صورتی که قرار باشد از یک تابع مانند $f(X) \in \mathbb{R}^{n \times m}$ بر حسب یک ماتریس همچون $X \in \mathbb{R}^{k \times p}$ مشتق بگیریم، حاصل این کار یک تنسور[۱] $\frac{\partial f(X)}{\partial X} \in \mathbb{R}^{n \times m \times k \times p}$ از مرتبه چهار خواهد شد.

[۱] Tensor

به صورت مشابه می‌توان برای ابعاد بالاتر نیز مشتق‌گیری را انجام داد. به یاد داشته باشید که برای مشتق‌گیری از هر تابعی با هر ابعادی نسبت به هر ورودی با هر ابعادی، کافیست نسبت به المان‌های آن‌ها، نظیر به نظیر مشتق جزئی را محاسبه نماییم و مقادیر بدست آمده را کنار هم قرار دهیم. یک روش ساده برای جلوگیری از مواجهه با تنسورها این است که در صورت نیاز، ماتریسی همچون $A \in \mathbb{R}^{m\times n}$ را به شکل یک بردار تخت نظیر $a \in \mathbb{R}^{mn}$ درآورد و نسبت به آن مشتق بگیریم. به زبان ریاضی خواهیم داشت:

$$A \in \mathbb{R}^{n\times m}, a \in \mathbb{R}^{nm} \to A_{ij} = a_{(i-1)n+j}$$

با استفاده از این تعریف تلاش کنید تا به پرسش‌های زیر پاسخ دهید و هرجا که نیاز شد، بجای ایجاد تنسور از تخت‌سازی ماتریس‌ها استفاده کنید. (ذکر دقیق مراحل برای کسب نمره ضروری است)

۱. اگر $a, x \in \mathbb{R}^n$ باشند، نشان دهید که:

$$\frac{\partial(a^\top x)}{\partial x} = \frac{\partial(x^\top a)}{\partial x} = a^\top.$$

طبق صورت سوال داریم:

$$a = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \quad \text{و} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

بنابراین، می‌توان نوشت:

$$a^\top x = \sum_{i=1}^{n} a_i\, x_i.$$

از طرفی مشتق جزئی نسبت به x_j به صورت زیر محاسبه می‌شود:

$$\frac{\partial}{\partial x_j}\left(a^\top x\right) = \frac{\partial}{\partial x_j}\left(\sum_{i=1}^{n} a_i\, x_i\right) = a_j$$

با جمع‌آوری مشتقات جزئی نسبت به $x_1, x_2, \dots, x_n$ به صورت یک بردار سطری، خواهیم داشت:

$$\frac{\partial(a^\top x)}{\partial x} = \left(a_1, a_2, \dots, a_n\right) = a^\top$$

از آنجا که $x^\top a = a^\top x$ (هر دو عبارت یک اسکالر هستند) نتیجه می‌گیریم:

$$\frac{\partial(x^\top a)}{\partial x} = a^\top.$$

۲. برای $x \in \mathbb{R}^n$ و $A \in \mathbb{R}^{m\times n}$، مقدار

$$\frac{\partial(Ax)}{\partial x}$$

را بیابید.

ماتریس A به صورت زیر است:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

و بردار x به صورت زیر تعریف می‌شود:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

پس ضرب Ax برابر است با:

$$Ax = \begin{bmatrix} \sum_{j=1}^{n} a_{1j}x_j \\ \sum_{j=1}^{n} a_{2j}x_j \\ \vdots \\ \sum_{j=1}^{n} a_{mj}x_j \end{bmatrix}$$

برای هر مؤلفه‌ی iـام داریم:

$$(Ax)_i = \sum_{j=1}^{n} a_{ij}x_j.$$

با مشتق‌گیری جزئی نسبت به x_k خواهیم داشت:

$$\frac{\partial (Ax)_i}{\partial x_k} = a_{ik}$$

پس مشتق کلی به صورت ماتریس Jacobian خواهد بود:

$$\frac{\partial (Ax)}{\partial x} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = A$$

بنابراین نتیجه می‌گیریم که:

$$\frac{\partial (Ax)}{\partial x} = A.$$

۳. برای $A \in \mathbb{R}^{n\times n}$ و $x \in \mathbb{R}^n$، عبارت

$$\frac{\partial (x^\top Ax)}{\partial x}$$

را محاسبه کنید. همچنین مشتق نسبت به A به صورت

$$\frac{\partial (x^\top Ax)}{\partial A}$$

را نیز تعیین کنید.

عبارت داده‌شده برابر است با:

$$f(x) = x^\top Ax$$

ابتدا این مقدار را به‌صورت زیر بازنویسی می‌کنیم:

$$x^\top Ax = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i a_{ij} x_j$$

حال مشتق جزئی نسبت به x_k را محاسبه می‌کنیم:

$$\frac{\partial}{\partial x_k}\sum_{i=1}^{n}\sum_{j=1}^{n} x_i a_{ij} x_j$$

با توجه به مشتق‌گیری جزئی، اگر $i = k$ یا $j = k$، هر جمله شامل x_k مشتق گرفته می‌شود:

$$\frac{\partial(x^\top Ax)}{\partial x_k} = \sum_{j=1}^{n} a_{kj} x_j + \sum_{i=1}^{n} x_i a_{ik}$$

این را به زبان ماتریسی بازنویسی می‌کنیم:

$$\frac{\partial(x^\top Ax)}{\partial x} = (A + A^\top)x$$

بنابراین داریم:

$$\frac{\partial(x^\top Ax)}{\partial x} = (A + A^\top)x$$

حال نسبت به A مشتق‌گیری می‌کنیم:

$$\frac{\partial(x^\top Ax)}{\partial A}$$

مشتق یک اسکالر نسبت به یک ماتریس متقارن به صورت زیر است:

$$\frac{\partial(x^\top Ax)}{\partial A} = xx^\top$$

بنابراین:

$$\frac{\partial(x^\top Ax)}{\partial A} = xx^\top.$$

۴. برای $A, X \in \mathbb{R}^{n\times n}$، مقدار

$$\frac{\partial\,\mathrm{tr}(X^\top AX)}{\partial X}$$

را محاسبه کنید.

ابتدا تابع مورد نظر را می‌نویسیم:

$$f(X) = \mathrm{tr}(X^\top AX).$$

با استفاده از خاصیت‌های trace ماتریس‌ها داریم:

$$\mathrm{tr}(X^\top AX) = \sum_{i=1}^{n}\sum_{j=1}^{n}(X^\top AX)_{ij}$$

با گسترش هر جمله داریم:

$$\mathrm{tr}(X^\top AX) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} X_{ki} A_{ij} X_{jk}$$

حال مشتق جزئی نسبت به X_{pq} را محاسبه می‌کنیم. تنها جملاتی که X_{pq} در آنها حضور دارد مشتق گرفته می‌شوند:

$$\frac{\partial}{\partial X_{pq}} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} X_{ki} A_{ij} X_{jk}.$$

هر جمله‌ای که در آن X_{pq} ظاهر می‌شود، دو حالت دارد: ۱. زمانی که X_{pq} از سمت چپ ظاهر شده است:

$$\sum_{j=1}^{n} A_{qj} X_{jp}$$

۲. زمانی که X_{pq} از سمت راست ظاهر شده است:

$$\sum_{i=1}^{n} X_{qi} A_{ip}$$

در نتیجه، فرم ماتریسی آن به‌صورت زیر خواهد بود:

$$\frac{\partial \operatorname{tr}(X^\top A X)}{\partial X} = AX + A^\top X$$

در صورتی که A متقارن باشد (یعنی $A^\top = A$)، نتیجه ساده‌تر خواهد شد:

$$\frac{\partial \operatorname{tr}(X^\top A X)}{\partial X} = 2AX.$$

پرسش ۲. Backpropagation (۲۵ نمره)

در این سوال، با یک مسئله دسته بندی سه کلاسه روبه‌رو هستیم. معماری شبکه‌ی عصبی مورد استفاده به صورت زیر است:

$$l^{(1)} = \operatorname{ReLU}(W^{(1)}x), \quad l^{(21)} = \operatorname{ReLU}(W^{(21)}l^{(1)}), \quad l^{(22)} = \sigma(W^{(22)}l^{(1)}), \quad z = \max(l^{(21)}, l^{(22)})$$

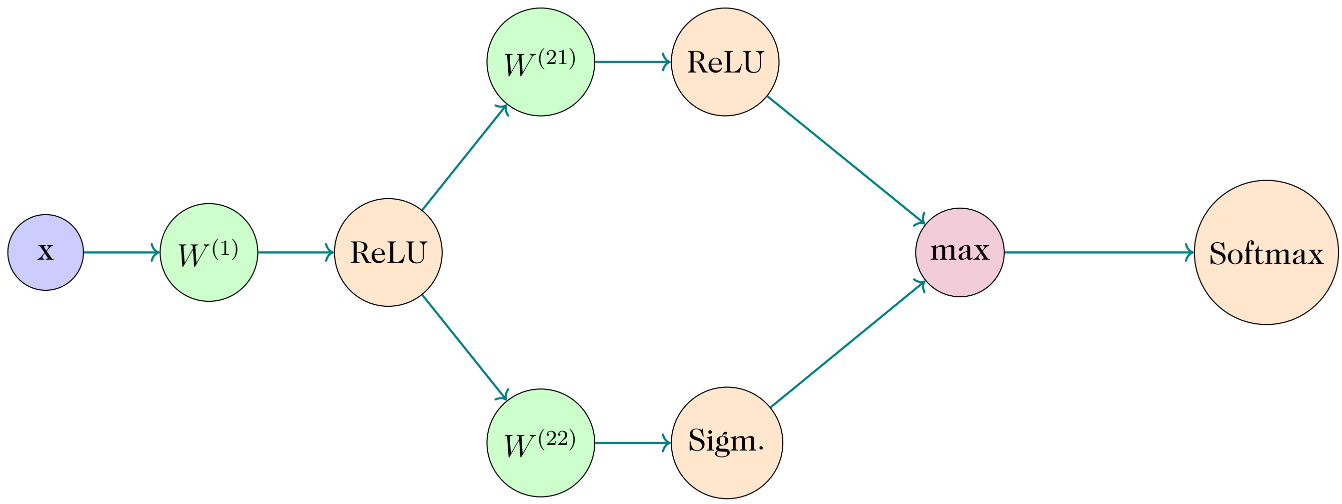
علاوه بر این، از لایه‌ی Softmax برای خروجی شبکه استفاده شده است:

$$\hat{y} = \operatorname{softmax}(z)$$

ابعاد متغیرها به‌صورت زیر داده شده‌اند:

$$x \in \mathbb{R}^4, \quad W^{(1)} \in \mathbb{R}^{2\times 4}, \quad W^{(21)}, W^{(22)} \in \mathbb{R}^{3\times 2}$$

۱. گراف محاسباتی این شبکه را رسم کنید. در این گراف، هر گره نشان‌دهنده‌ی یک عملیات (نظیر ReLU، sigmoid، ضرب ماتریسی، و انتخاب ماکسیمم) است و یال‌ها وابستگی بین این مقادیر را نشان می‌دهند.



۲. در مرحله‌ی Backward Pass، گرادیان تابع هزینه $\mathcal{L}$ نسبت به وزن‌های شبکه را محاسبه کنید. توجه کنید که در Forward Pass برخی مقادیر محاسبه شده و ذخیره می‌شوند و نیازی به محاسبه‌ی مجدد آن‌ها نیست. در پاسخ، از گراف محاسباتی استفاده کنید و روابط زیر را به‌دست آورید:

$$\frac{\partial \mathcal{L}}{\partial z}, \quad \frac{\partial \mathcal{L}}{\partial l^{(21)}}, \quad \frac{\partial \mathcal{L}}{\partial l^{(22)}}, \quad \frac{\partial \mathcal{L}}{\partial W^{(21)}},$$
$$\frac{\partial \mathcal{L}}{\partial W^{(22)}}, \quad \frac{\partial \mathcal{L}}{\partial l^{(1)}}, \quad \frac{\partial \mathcal{L}}{\partial W^{(1)}}$$

در این بخش، مشتق‌ها را گام‌به‌گام بر اساس زنجیره‌ی محاسباتی به‌دست آورید.

فرض می‌کنیم تابع هزینه Cross-Entropy به صورت زیر تعریف می‌شود:

$$L = -\sum_{i=1}^{3} y_i \log(\hat{y}_i).$$

حال مشتقات لازم را گام به گام محاسبه می‌کنیم:

۱. مشتق تابع هزینه نسبت به z:

ابتدا تابع Softmax به صورت زیر تعریف می‌شود:

$$\hat{y}_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

مشتق این تابع نسبت به z_k به صورت زیر بدست می‌آید:

$$\frac{\partial \hat{y}_i}{\partial z_k} = \begin{cases} \hat{y}_i\,(1-\hat{y}_i), & \text{if } i = k \\ -\hat{y}_i\,\hat{y}_k, & \text{if } i \neq k \end{cases}$$

حال تابع هزینه به صورت

$$L = -\sum_i y_i \log(\hat{y}_i)$$

تعریف می‌شود. مشتق L نسبت به $\hat{y}_i$ به صورت

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

می‌باشد. طبق قانون زنجیره‌ای داریم:

$$\frac{\partial L}{\partial z_k} = \sum_i \frac{\partial L}{\partial \hat{y}_i} \frac{\partial \hat{y}_i}{\partial z_k}.$$

برای بررسی دو حالت مجزا عمل می‌کنیم:

حالت ۱: اگر $y_k = 1$ (یعنی کلاس درست)، بنابراین برای $i = k$ داریم:

$$\frac{\partial L}{\partial z_k} = -\frac{y_k}{\hat{y}_k}\, \hat{y}_k(1-\hat{y}_k) + \sum_{i\neq k} \left(-\frac{0}{\hat{y}_i}\right) \left(-\hat{y}_i\, \hat{y}_k\right) = -1\,(1-\hat{y}_k).$$

با توجه به اینکه $y_k = 1$ می‌توان نوشت:

$$-1\,(1-\hat{y}_k) = \hat{y}_k - 1 = \hat{y}_k - y_k.$$

حالت ۲: اگر $y_k = 0$ (کلاس اشتباه)؛ در این حالت برای $i = k$:

$$\frac{\partial L}{\partial z_k} = -\frac{0}{\hat{y}_k}\, \hat{y}_k(1-\hat{y}_k) = 0$$

و تنها یک جمله از جمع برای $i = t$ وجود دارد که $y_t = 1$ (کلاس درست) و $t \neq k$:

$$\frac{\partial L}{\partial z_k} = -\frac{y_t}{\hat{y}_t} \left(-\hat{y}_t\, \hat{y}_k\right) = \hat{y}_k.$$

بنابراین، چون $y_k = 0$ داریم:

$$\hat{y}_k = \hat{y}_k - 0 = \hat{y}_k - y_k$$

در نتیجه برای هر k:

$$\frac{\partial L}{\partial z_k} = \hat{y}_k - y_k, \quad \text{یا به صورت برداری:} \quad \frac{\partial L}{\partial z} = \hat{y} - y.$$

۲. مشتق نسبت به $\ell^{(21)}$ و $\ell^{(22)}$:

در مرحله‌ی چهارم داریم:

$$z = \max\left(\ell^{(21)},\, \ell^{(22)}\right)$$

که به صورت مؤلفه‌ای محاسبه می‌شود. برای هر مؤلفه i:

$$z_i = \begin{cases} \ell_i^{(21)} & \text{if } \ell_i^{(21)} \geq \ell_i^{(22)} \\ \ell_i^{(22)} & \text{if } \ell_i^{(22)} > \ell_i^{(21)} \end{cases}$$

بنابراین مشتق‌های مؤلفه‌ای به شکل زیر خواهند بود:

$$\frac{\partial z_i}{\partial \ell_i^{(21)}} = \begin{cases} 1, & \text{if } \ell_i^{(21)} > \ell_i^{(22)}, \\ 0, & \text{o.w.} \end{cases} \qquad \frac{\partial z_i}{\partial \ell_i^{(22)}} = \begin{cases} 0, & \text{if } \ell_i^{(21)} > \ell_i^{(22)}, \\ 1, & \text{o.w.} \end{cases}$$

از این رو:

$$\frac{\partial L}{\partial \ell_i^{(21)}} = \frac{\partial L}{\partial z_i}\,\mathbf{1}\big(\ell_i^{(21)} \geq \ell_i^{(22)}\big) = (\hat{y}_i - y_i)\,\mathbf{1}\big(\ell_i^{(21)} \geq \ell_i^{(22)}\big),$$
$$\frac{\partial L}{\partial \ell_i^{(22)}} = \frac{\partial L}{\partial z_i}\,\mathbf{1}\big(\ell_i^{(22)} > \ell_i^{(21)}\big) = (\hat{y}_i - y_i)\,\mathbf{1}\big(\ell_i^{(22)} > \ell_i^{(21)}\big).$$

۳. مشتق نسبت به $W^{(22)}$:

با توجه به اینکه

$$\ell^{(22)} = \sigma\big(W^{(22)}\,\ell^{(1)}\big),$$

فرض می‌کنیم

$$\mathrm{pre}^{(22)} = W^{(22)}\,\ell^{(1)}$$

و $\ell_i^{(22)} = \sigma\big(\mathrm{pre}_i^{(22)}\big)$. مشتق تابع سیگموید به صورت زیر است:

$$\frac{\partial \ell_i^{(22)}}{\partial \mathrm{pre}_i^{(22)}} = \sigma'\big(\mathrm{pre}_i^{(22)}\big) = \ell_i^{(22)}\,\big(1 - \ell_i^{(22)}\big).$$

با استفاده از قانون زنجیره‌ای داریم:

$$\frac{\partial L}{\partial \mathrm{pre}_i^{(22)}} = \frac{\partial L}{\partial \ell_i^{(22)}}\,\ell_i^{(22)}\,\big(1 - \ell_i^{(22)}\big) = (\hat{y}_i - y_i)\,\mathbf{1}\big(\ell_i^{(22)} > \ell_i^{(21)}\big)\,\ell_i^{(22)}\,\big(1 - \ell_i^{(22)}\big).$$

از آنجا که

$$\mathrm{pre}_i^{(22)} = \sum_j W_{i,j}^{(22)}\,\ell_j^{(1)},$$

داریم:

$$\frac{\partial \mathrm{pre}_i^{(22)}}{\partial W_{i,j}^{(22)}} = \ell_j^{(1)}.$$

بنابراین:

$$\frac{\partial L}{\partial W_{i,j}^{(22)}} = (\hat{y}_i - y_i)\,\mathbf{1}\big(\ell_i^{(22)} > \ell_i^{(21)}\big)\,\ell_i^{(22)}\,\big(1 - \ell_i^{(22)}\big)\,\ell_j^{(1)}.$$

۴. مشتق نسبت به $W^{(21)}$:

به طور مشابه، چون

$$\ell^{(21)} = \mathrm{ReLU}\big(W^{(21)}\,\ell^{(1)}\big),$$

فرض کنید

$$\mathrm{pre}^{(21)} = W^{(21)}\,\ell^{(1)}$$

و $\ell_i^{(21)} = \mathrm{ReLU}\big(\mathrm{pre}_i^{(21)}\big) = \max\{0,\, \mathrm{pre}_i^{(21)}\}$. مشتق ReLU به صورت زیر است:

$$\frac{\partial \ell_i^{(21)}}{\partial \mathrm{pre}_i^{(21)}} = \mathbf{1}\big(\mathrm{pre}_i^{(21)} > 0\big).$$

بنابراین:

$$\frac{\partial L}{\partial \mathrm{pre}_i^{(21)}} = \frac{\partial L}{\partial \ell_i^{(21)}}\,\mathbf{1}\big(\mathrm{pre}_i^{(21)} > 0\big) = (\hat{y}_i - y_i)\,\mathbf{1}\big(\ell_i^{(21)} \geq \ell_i^{(22)}\big)\,\mathbf{1}\big(\mathrm{pre}_i^{(21)} > 0\big).$$

با توجه به اینکه

$$\frac{\partial \mathrm{pre}_i^{(21)}}{\partial W_{i,j}^{(21)}} = \ell_j^{(1)}$$

نتیجه می‌گیریم:

$$\frac{\partial L}{\partial W_{i,j}^{(21)}} = (\hat{y}_i - y_i)\, \mathbf{1}\big(\ell_i^{(21)} \geq \ell_i^{(22)}\big)\, \mathbf{1}\big(\mathrm{pre}_i^{(21)} > 0\big)\, \ell_j^{(1)}.$$

۵. مشتق نسبت به $\ell^{(1)}$:

از آنجا که $\ell^{(1)}$ در هر دو شاخه (برای $l^{(21)}$ و $l^{(22)}$) استفاده می‌شود، داریم:

$$\frac{\partial L}{\partial \ell^{(1)}} = \sum_i \left[\frac{\partial L}{\partial \ell_i^{(21)}} \frac{\partial \ell_i^{(21)}}{\partial \ell^{(1)}} + \frac{\partial L}{\partial \ell_i^{(22)}} \frac{\partial \ell_i^{(22)}}{\partial \ell^{(1)}} \right]$$

که در آن:

$$\frac{\partial \ell_i^{(21)}}{\partial \ell_j^{(1)}} = \mathbf{1}\big(\mathrm{pre}_i^{(21)} > 0\big)\, W_{i,j}^{(21)}$$

$$\frac{\partial \ell_i^{(22)}}{\partial \ell_j^{(1)}} = \sigma'\big(\mathrm{pre}_i^{(22)}\big)\, W_{i,j}^{(22)} = \ell_i^{(22)}\, \big(1 - \ell_i^{(22)}\big)\, W_{i,j}^{(22)}.$$

۶. مشتق نسبت به $W^{(1)}$:

در نهایت، چون:

$$\ell^{(1)} = \mathrm{ReLU}\big(W^{(1)}\, x\big),$$

فرض کنید

$$\mathrm{pre}^{(1)} = W^{(1)}\, x$$

و $\ell_k^{(1)} = \mathrm{ReLU}\big(\mathrm{pre}_k^{(1)}\big)$. مشتق ReLU به صورت:

$$\frac{\partial \ell_k^{(1)}}{\partial \mathrm{pre}_k^{(1)}} = \mathbf{1}\big(\mathrm{pre}_k^{(1)} > 0\big)$$

محاسبه می‌شود. بنابراین:

$$\frac{\partial L}{\partial \mathrm{pre}_k^{(1)}} = \frac{\partial L}{\partial \ell_k^{(1)}}\, \mathbf{1}\big(\mathrm{pre}_k^{(1)} > 0\big).$$

از آنجا که

$$\mathrm{pre}_k^{(1)} = \sum_m W_{k,m}^{(1)}\, x_m$$

داریم:

$$\frac{\partial \mathrm{pre}_k^{(1)}}{\partial W_{k,m}^{(1)}} = x_m$$

و در نتیجه:

$$\frac{\partial L}{\partial W_{k,m}^{(1)}} = \frac{\partial L}{\partial \mathrm{pre}_k^{(1)}}\, x_m.$$

به طور کلی:

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

$$\frac{\partial L}{\partial \ell_i^{(21)}} = (\hat{y}_i - y_i)\,\mathbf{1}\big(\ell_i^{(21)} \geq \ell_i^{(22)}\big), \quad \frac{\partial L}{\partial \ell_i^{(22)}} = (\hat{y}_i - y_i)\,\mathbf{1}\big(\ell_i^{(22)} > \ell_i^{(21)}\big),$$

$$\frac{\partial L}{\partial W_{i,j}^{(22)}} = (\hat{y}_i - y_i)\,\mathbf{1}\big(\ell_i^{(22)} > \ell_i^{(21)}\big)\,\ell_i^{(22)}\,\big(1 - \ell_i^{(22)}\big)\,\ell_j^{(1)},$$

$$\frac{\partial L}{\partial W_{i,j}^{(21)}} = (\hat{y}_i - y_i)\,\mathbf{1}\big(\ell_i^{(21)} \geq \ell_i^{(22)}\big)\,\mathbf{1}\big(\mathrm{pre}_i^{(21)} > 0\big)\,\ell_j^{(1)},$$

$$\frac{\partial L}{\partial \ell_j^{(1)}} = \sum_i \Big[\frac{\partial L}{\partial \ell_i^{(21)}}\,\mathbf{1}\big(\mathrm{pre}_i^{(21)} > 0\big)\,W_{i,j}^{(21)} + \frac{\partial L}{\partial \ell_i^{(22)}}\,\ell_i^{(22)}\,\big(1 - \ell_i^{(22)}\big)\,W_{i,j}^{(22)}\Big],$$

$$\frac{\partial L}{\partial W_{k,m}^{(1)}} = \Big[\frac{\partial L}{\partial \ell_k^{(1)}}\Big]\,\mathbf{1}\big(\mathrm{pre}_k^{(1)} > 0\big)\,x_m.$$

۳. مقدار خروجی و گرادیان‌ها را بر اساس مقداردهی اولیه‌ی زیر محاسبه کنید. فرض کنید که تابع هزینه‌ی مورد استفاده Cross Entropy است و داده‌ی ورودی به کلاس دوم تعلق دارد.

$$x = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix}, \quad W^{(1)} = \begin{bmatrix} -1 & 0 & 1 & -1 \\ 0 & -1 & 1 & 1 \end{bmatrix}, \quad W^{(21)} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \end{bmatrix}, \quad W^{(22)} = \begin{bmatrix} 2 & -1 \\ 4 & -2 \\ -2 & 1 \end{bmatrix}$$

مراحل مورد نیاز برای محاسبه‌ی Forward Pass را به طور کامل نمایش دهید. در پایان، مقدار $\hat{y}$ را محاسبه کرده و لاجیت‌ها را تا دو رقم اعشار گرد کنید.

سپس، با استفاده از قواعد به‌دست‌آمده در قسمت قبل، Backward Pass را اجرا کنید و گرادیان‌های وزن‌ها را تعیین کنید.

Forward Pass

الف) محاسبه $\ell^{(1)}$:

$$\mathrm{pre}^{(1)} = W^{(1)}\,x$$

با توجه به:

$$W^{(1)} = \begin{bmatrix} -1 & 0 & 1 & -1 \\ 0 & -1 & 1 & 1 \end{bmatrix}, \quad x = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix}$$

داریم:

$$\begin{aligned} \mathrm{pre}_1^{(1)} &= (-1)\cdot 1 + 0\cdot 2 + 1\cdot 3 + (-1)\cdot 1 = -1 + 0 + 3 - 1 = 1 \\ \mathrm{pre}_2^{(1)} &= 0\cdot 1 + (-1)\cdot 2 + 1\cdot 3 + 1\cdot 1 = 0 - 2 + 3 + 1 = 2 \end{aligned}$$

با اعمال تابع ReLU خواهیم داشت:

$$\ell^{(1)} = \text{ReLU}\big(\text{pre}^{(1)}\big) = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

ب) محاسبه $\ell^{(21)}$:

$$\text{pre}^{(21)} = W^{(21)}\,\ell^{(1)}$$

با توجه به:

$$W^{(21)} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \end{bmatrix}, \quad \ell^{(1)} = \begin{bmatrix} 1 \\ 2 \end{bmatrix},$$

داریم:

$$\begin{aligned} \text{pre}_1^{(21)} &= 1 \cdot 1 + 1 \cdot 2 = 3 \\ \text{pre}_2^{(21)} &= 1 \cdot 1 + (-1) \cdot 2 = 1 - 2 = -1 \\ \text{pre}_3^{(21)} &= (-1) \cdot 1 + 1 \cdot 2 = -1 + 2 = 1 \end{aligned}$$

با اعمال ReLU خواهیم داشت:

$$\ell^{(21)} = \text{ReLU}\big(\text{pre}^{(21)}\big) = \begin{bmatrix} 3 \\ 0 \\ 1 \end{bmatrix}$$

ج) محاسبه $\ell^{(22)}$:

$$\text{pre}^{(22)} = W^{(22)}\,\ell^{(1)}$$

با توجه به:

$$W^{(22)} = \begin{bmatrix} 2 & -1 \\ 4 & -2 \\ -2 & 1 \end{bmatrix}, \quad \ell^{(1)} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

داریم:

$$\begin{aligned} \text{pre}_1^{(22)} &= 2 \cdot 1 + (-1) \cdot 2 = 2 - 2 = 0, \\ \text{pre}_2^{(22)} &= 4 \cdot 1 + (-2) \cdot 2 = 4 - 4 = 0, \\ \text{pre}_3^{(22)} &= (-2) \cdot 1 + 1 \cdot 2 = -2 + 2 = 0. \end{aligned}$$

با اعمال تابع سیگموید $\sigma(z) = \frac{1}{1+e^{-z}}$ (که $\sigma(0) = 0.5$) داریم:

$$\ell^{(22)} = \sigma\big(\text{pre}^{(22)}\big) = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$$

د) محاسبه z:

لاجیت نهایی با انتخاب مؤلفه‌ای (عملگر max) از دو شاخه به صورت زیر تعریف می‌شود:

$$z_i = \max\big(\ell_i^{(21)},\, \ell_i^{(22)}\big).$$

بنابراین:

$$\begin{aligned} z_1 &= \max(3,\, 0.5) = 3, \\ z_2 &= \max(0,\, 0.5) = 0.5, \\ z_3 &= \max(1,\, 0.5) = 1. \end{aligned}$$

گرد کردن لاجیت‌ها به دو رقم اعشار:

$$z \approx \begin{bmatrix} 3.00 \\ 0.50 \\ 1.00 \end{bmatrix}$$

ه) محاسبه Softmax و $\hat{y}$:

فرمول تابع Softmax به صورت زیر است:

$$\hat{y}_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

با تقریب تا دو رقم اعشار داریم:

$$e^{3.00} \approx 20.09, \quad e^{0.50} \approx 1.65, \quad e^{1.00} \approx 2.72,$$

مجموع این اعداد برابر است با:

$$S = 20.09 + 1.65 + 2.72 \approx 24.46$$

طبق فرمول نوشته شده داریم:

$$\begin{aligned} \hat{y}_1 &\approx \tfrac{20.09}{24.46} \approx 0.82, \\ \hat{y}_2 &\approx \tfrac{1.65}{24.46} \approx 0.07, \\ \hat{y}_3 &\approx \tfrac{2.72}{24.46} \approx 0.11 \end{aligned}$$

پس ماتریس $\hat{y}$ به صورت زیر خواهد بود:

$$\hat{y} = \begin{bmatrix} 0.82 \\ 0.07 \\ 0.11 \end{bmatrix}$$

Backward Pass

تابع هزینه سه کلاسه Cross-Entropy به صورت زیر تعریف می‌شود:

$$L = -\sum_{i=1}^{3} y_i \, \log(\hat{y}_i)$$

طبق محاسبات قبلی داریم:

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

با توجه به $y=(0,1,0)^T$ و $\hat{y}\approx(0.82,\,0.07,\,0.11)^T$:

$$\frac{\partial L}{\partial z}\approx\begin{bmatrix}0.82-0\\0.07-1\\0.11-0\end{bmatrix}=\begin{bmatrix}0.82\\-0.93\\0.11\end{bmatrix}$$

الف) انتشار گرادیان از طریق عملگر max:

با توجه به انتخاب شاخه‌ها:

$$\begin{aligned}
\ell_1^{(21)}=3 \quad &> \quad \ell_1^{(22)}=0.5 \quad \Rightarrow \quad \frac{\partial L}{\partial \ell_1^{(21)}}=0.82, \quad \frac{\partial L}{\partial \ell_1^{(22)}}=0,\\
\ell_2^{(21)}=0 \quad &< \quad \ell_2^{(22)}=0.5 \quad \Rightarrow \quad \frac{\partial L}{\partial \ell_2^{(22)}}=-0.93, \quad \frac{\partial L}{\partial \ell_2^{(21)}}=0,\\
\ell_3^{(21)}=1 \quad &> \quad \ell_3^{(22)}=0.5 \quad \Rightarrow \quad \frac{\partial L}{\partial \ell_3^{(21)}}=0.11, \quad \frac{\partial L}{\partial \ell_3^{(22)}}=0.
\end{aligned}$$

اگر

$$\ell^{(21)}=\begin{bmatrix}3\\0\\1\end{bmatrix},\quad \ell^{(22)}=\begin{bmatrix}0.5\\0.5\\0.5\end{bmatrix},\quad g=\begin{bmatrix}0.82\\-0.93\\0.11\end{bmatrix},$$

آنگاه به‌صورت مؤلفه‌ای:

$$M_{21}=\begin{bmatrix}1\\0\\1\end{bmatrix},\quad M_{22}=\begin{bmatrix}0\\1\\0\end{bmatrix}.$$

پس

$$\frac{\partial L}{\partial \ell^{(21)}}=\begin{bmatrix}0.82\\-0.93\\0.11\end{bmatrix}\odot\begin{bmatrix}1\\0\\1\end{bmatrix}=\begin{bmatrix}0.82\\0\\0.11\end{bmatrix},\quad \frac{\partial L}{\partial \ell^{(22)}}=\begin{bmatrix}0.82\\-0.93\\0.11\end{bmatrix}\odot\begin{bmatrix}0\\1\\0\end{bmatrix}=\begin{bmatrix}0\\-0.93\\0\end{bmatrix}.$$

ب) گرادیان‌های شاخه $W^{(22)}$:

برای شاخه $\ell^{(22)}$ داریم:

$$\ell_i^{(22)}=\sigma\big(\mathrm{pre}_i^{(22)}\big),\quad \sigma'(z)=\ell^{(22)}(1-\ell^{(22)})$$

از آنجایی که $\mathrm{pre}_i^{(22)}=0$ و $\ell_i^{(22)}=0.5$ برای هر i، داریم:

$$\sigma'(0)=0.5(1-0.5)=0.25$$

بنابراین:

$$\frac{\partial L}{\partial \mathrm{pre}_i^{(22)}}=\frac{\partial L}{\partial \ell_i^{(22)}}\cdot 0.25$$

تنها برای $i=2$ که گرادیان غیر صفر است:

$$\frac{\partial L}{\partial \mathrm{pre}_2^{(22)}}=-0.93\times 0.25\approx -0.2325$$

گرادیان نسبت به وزن‌های $W^{(22)}$ (با توجه به $\mathrm{pre}_i^{(22)}=\sum_j W_{i,j}^{(22)}\,\ell_j^{(1)}$):

$$\frac{\partial L}{\partial W_{i,j}^{(22)}}=\frac{\partial L}{\partial \mathrm{pre}_i^{(22)}}\,\ell_j^{(1)}$$

بنابراین تنها برای ردیف $i = 2$ و $\ell^{(1)} = [1, 2]$:

$$\frac{\partial L}{\partial W^{(22)}_{2,1}} \approx -0.2325 \times 1 \approx -0.2325, \quad \frac{\partial L}{\partial W^{(22)}_{2,2}} \approx -0.2325 \times 2 \approx -0.4650$$

به صورت ماتریسی، این رابطه به شکل زیر نوشته می‌شود:

$$\frac{\partial L}{\partial W^{(22)}} = \left(\frac{\partial L}{\partial \mathrm{pre}^{(22)}}\right) \left(\ell^{(1)}\right)^T,$$

$$\frac{\partial L}{\partial W^{(22)}} = \begin{bmatrix} 0 \\ -0.2325 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -0.2325 & -0.465 \\ 0 & 0 \end{bmatrix}.$$

ج) گرادیان‌های شاخه $W^{(21)}$:

در شاخه $\ell^{(21)}$، داریم:

$$\ell^{(21)} = \mathrm{ReLU}\big(\mathrm{pre}^{(21)}\big), \quad \frac{\partial \ell^{(21)}_i}{\partial \mathrm{pre}^{(21)}_i} = \begin{cases} 1, & \mathrm{pre}^{(21)}_i > 0, \\ 0, & \text{o.w.} \end{cases}$$

از آنجایی که $\mathrm{pre}^{(21)}_1 = 3 > 0$ و $\mathrm{pre}^{(21)}_3 = 1 > 0$ (و $\mathrm{pre}^{(21)}_2 = -1$ پس غیرفعال است)، داریم:

$$\frac{\partial L}{\partial \mathrm{pre}^{(21)}_1} = 0.82, \quad \frac{\partial L}{\partial \mathrm{pre}^{(21)}_3} = 0.11, \quad \frac{\partial L}{\partial \mathrm{pre}^{(21)}_2} = 0$$

گرادیان نسبت به $W^{(21)}$ (با $\mathrm{pre}^{(21)}_i = \sum_j W^{(21)}_{i,j}\, \ell^{(1)}_j$):

$$\frac{\partial L}{\partial W^{(21)}_{i,j}} = \frac{\partial L}{\partial \mathrm{pre}^{(21)}_i}\, \ell^{(1)}_j$$

بنابراین:

$$\begin{aligned}
\text{برای}\, i = 1: &\quad \frac{\partial L}{\partial W^{(21)}_{1,1}} = 0.82 \times 1 = 0.82, \quad \frac{\partial L}{\partial W^{(21)}_{1,2}} = 0.82 \times 2 = 1.64, \\
\text{برای}\, i = 3: &\quad \frac{\partial L}{\partial W^{(21)}_{3,1}} = 0.11 \times 1 = 0.11, \quad \frac{\partial L}{\partial W^{(21)}_{3,2}} = 0.11 \times 2 = 0.22, \\
\text{برای}\, i = 2: &\quad \frac{\partial L}{\partial W^{(21)}_{2,j}} = 0.
\end{aligned}$$

$$\mathrm{pre}^{(21)}_i = \sum_j W^{(21)}_{i,j}\, \ell^{(1)}_j \quad \Longrightarrow \quad \frac{\partial \mathrm{pre}^{(21)}_i}{\partial W^{(21)}_{i,j}} = \ell^{(1)}_j$$

در فرمت ماتریسی، این مشتق به صورت زیر نوشته می‌شود:

$$\frac{\partial L}{\partial W^{(21)}} = \left(\frac{\partial L}{\partial \mathrm{pre}^{(21)}}\right) \left(\ell^{(1)}\right)^T$$

$$\frac{\partial L}{\partial W^{(21)}} = \begin{bmatrix} 0.82 \\ 0 \\ 0.11 \end{bmatrix} \begin{bmatrix} 1 & 2 \end{bmatrix} = \begin{bmatrix} 0.82 \times 1 & 0.82 \times 2 \\ 0 & 0 \\ 0.11 \times 1 & 0.11 \times 2 \end{bmatrix} = \begin{bmatrix} 0.82 & 1.64 \\ 0 & 0 \\ 0.11 & 0.22 \end{bmatrix}.$$

د) گرادیان نسبت به $\ell^{(1)}$:

از آنجایی که $\ell^{(1)}$ در هر دو شاخه 21 و 22 استفاده شده است:

$$\frac{\partial L}{\partial \ell^{(1)}} = \frac{\partial L}{\partial \ell^{(1)}}\Big|_{(21)} + \frac{\partial L}{\partial \ell^{(1)}}\Big|_{(22)},$$

که در آن:

$$\frac{\partial L}{\partial \ell_j^{(1)}}\Big|_{(21)} = \sum_i \frac{\partial L}{\partial \mathrm{pre}_i^{(21)}} W_{i,j}^{(21)},$$

و

$$\frac{\partial L}{\partial \ell_j^{(1)}}\Big|_{(22)} = \sum_i \frac{\partial L}{\partial \mathrm{pre}_i^{(22)}} W_{i,j}^{(22)}$$

با توجه به اینکه تنها شاخه 21 برای $i = 1, 3$ موثر است:

$$\begin{aligned} \text{برای} j = 1: &\quad 0.82 \cdot W_{1,1}^{(21)} + 0.11 \cdot W_{3,1}^{(21)} = 0.82 \cdot 1 + 0.11 \cdot (-1) = 0.82 - 0.11 = 0.71, \\ \text{برای} j = 2: &\quad 0.82 \cdot W_{1,2}^{(21)} + 0.11 \cdot W_{3,2}^{(21)} = 0.82 \cdot 1 + 0.11 \cdot 1 = 0.82 + 0.11 = 0.93 \end{aligned}$$

از شاخه 22 تنها $i = 2$ موثر است (چون تنها در آن گرادیان غیر صفر داریم):

$$\begin{aligned} \text{برای} j = 1: &\quad -0.2325 \cdot W_{2,1}^{(22)} = -0.2325 \cdot 4 = -0.93, \\ \text{برای} j = 2: &\quad -0.2325 \cdot W_{2,2}^{(22)} = -0.2325 \cdot (-2) = 0.465 \end{aligned}$$

بنابراین:

$$\begin{aligned} \tfrac{\partial L}{\partial \ell_1^{(1)}} &\approx 0.71 - 0.93 = -0.22, \\ \tfrac{\partial L}{\partial \ell_2^{(1)}} &\approx 0.93 + 0.465 = 1.395 \end{aligned}$$

این روابط به شکل کلی در قالب ماتریسی به صورت زیر نوشته می‌شوند:

$$\frac{\partial L}{\partial \ell^{(1)}} = \left(W^{(21)}\right)^T \frac{\partial L}{\partial \mathrm{pre}^{(21)}} + \left(W^{(22)}\right)^T \frac{\partial L}{\partial \mathrm{pre}^{(22)}}$$

بنابراین، به صورت ماتریسی:

$$\frac{\partial L}{\partial l^{(1)}} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 0.82 \\ 0 \\ 0.11 \end{bmatrix} + \begin{bmatrix} 2 & 4 & -2 \\ -1 & -2 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -0.2325 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.22 \\ 1.395 \end{bmatrix}.$$

ه) گرادیان‌های $W^{(1)}$:

با توجه به:

$$\ell^{(1)} = \mathrm{ReLU}\big(W^{(1)} x\big),$$

که $\mathrm{pre}_k^{(1)} > 0$ برای هر دو مؤلفه داریم، پس:

$$\frac{\partial L}{\partial W_{k,m}^{(1)}} = \frac{\partial L}{\partial \ell_k^{(1)}} x_m$$

برای $k = 1$:

$$\frac{\partial L}{\partial W_{1,\cdot}^{(1)}} = -0.22 \times [1, \quad 2, \quad 3, \quad 1] \approx [-0.22, \quad -0.44, \quad -0.66, \quad -0.22]$$

برای $k=2$:

$$\frac{\partial L}{\partial W^{(1)}_{2,\cdot}} = 1.395 \times \begin{bmatrix}1, & 2, & 3, & 1\end{bmatrix} \approx \begin{bmatrix}1.40, & 2.79, & 4.19, & 1.40\end{bmatrix}$$

با توجه به اینکه

$$\ell^{(1)} = \mathrm{ReLU}\big(W^{(1)} x\big)$$

و فرض می‌کنیم برای هر مؤلفه k داریم $\mathrm{pre}^{(1)}_k > 0$ (بنابراین مشتق ReLU برابر ۱ است)، آنگاه از قانون زنجیره‌ای می‌نویسیم:

$$\frac{\partial L}{\partial W^{(1)}_{k,m}} = \frac{\partial L}{\partial \ell^{(1)}_k} x_m.$$

به عبارت دیگر، به صورت ماتریسی:

$$\frac{\partial L}{\partial W^{(1)}} = \left(\frac{\partial L}{\partial \ell^{(1)}}\right) x^T$$

از قسمت‌های قبل داریم:

$$\frac{\partial L}{\partial \ell^{(1)}} = \begin{bmatrix}-0.22\\ 1.395\end{bmatrix} \quad \text{و} \quad x = \begin{bmatrix}1\\2\\3\\1\end{bmatrix}$$

بنابراین ضرب ماتریسی به صورت زیر محاسبه می‌شود:

$$\frac{\partial L}{\partial W^{(1)}} = \begin{bmatrix}-0.22\\ 1.395\end{bmatrix} \begin{bmatrix}1 & 2 & 3 & 1\end{bmatrix} = \begin{bmatrix}-0.22\times 1 & -0.22\times 2 & -0.22\times 3 & -0.22\times 1\\ 1.395\times 1 & 1.395\times 2 & 1.395\times 3 & 1.395\times 1\end{bmatrix}.$$

با انجام ضرب‌ها داریم:

$$\frac{\partial L}{\partial W^{(1)}} \approx \begin{bmatrix}-0.22 & -0.44 & -0.66 & -0.22\\ 1.40 & 2.79 & 4.19 & 1.40\end{bmatrix}.$$

پرسش ۳. Backtracking Line Search (۱۰ نمره)

در روش‌های بهینه‌سازی مانند Gradient Descent، مقدار مناسب برای t نقش مهمی در سرعت و همگرایی الگوریتم دارد. یکی از روش‌های رایج برای انتخاب مقدار t، روش Backtracking Line Search است که به‌صورت زیر تعریف می‌شود:

$$x_i = x_{i-1} - t_i \nabla f(x_{i-1})$$

Backtracking Line Search Algorithm:

Require: $\alpha \in (0, 0.5)$ (Sufficient decrease parameter)
Require: $\beta \in (0, 1)$ (Step size reduction factor)

Require: $t_0 > 0$ (Initial step size)

Initialize:
$t \leftarrow t_0$ (Set initial step size)

while $f(x + t\Delta x) > f(x) + \alpha t \nabla f(x)^T \Delta x$ **do**
$t \leftarrow \beta \cdot t$ (Reduce step size)
end while

return t

فرض کنید تابع f دارای مشتق دوم پیوسته بوده و کران ماتریسی زیر برای هسین آن برقرار باشد:

$$mI \preceq \nabla^2 f(x) \preceq MI$$

که در آن I ماتریس همانی است و نماد $\preceq$ نشان‌دهنده‌ی ترتیب جزئی بین ماتریس‌های متقارن (نیمه‌مثبت معین[۲]) می‌باشد. این شرط نشان می‌دهد که تمامی مقادیر ویژه‌ی $\nabla^2 f(x)$ در بازه‌ی $[m, M]$ قرار دارند.
به عبارت دیگر، از نامساوی $mI \preceq \nabla^2 f(x) \preceq MI$ نتیجه می‌شود که ماتریس $MI - \nabla^2 f(x)$ نیمه‌مثبت معین است، یعنی برای هر بردار v داریم:

$$v^T (MI - \nabla^2 f(x)) v \geq 0, \quad \forall v \in \mathbb{R}^n.$$

علاوه بر این، بردار Δx یک جهت کاهش در نقطه x است اما لزوماً برابر با $\nabla f(x)$ نیست.

۱. نشان دهید که اگر مقدار t در بازه‌ی زیر قرار گیرد، شرط توقف در Backtracking برقرار خواهد بود:

$$0 < t \leq -\frac{\nabla f(x)^T \Delta x}{M \|\Delta x\|_2^2}$$

راهنمایی: از بسط تیلور مرتبه دوم برای تابع $f(x)$ در جهت Δx استفاده کنید. سپس با استفاده از کران بالای $\nabla^2 f(x)$ نامساوی مناسب را استخراج کنید.

برای تحلیل تابع $f(x)$، از بسط تیلور مرتبه دوم استفاده می‌کنیم:

$$f(x) \approx f(a) + \nabla f(a)^T (x - a) + \frac{1}{2} (x - a)^T \nabla^2 f(a) (x - a)$$

با جایگذاری $x \to x + t\Delta x$ و $a \to x$، داریم:

$$f(x + t\Delta x) = f(x) + \nabla f(x)^T (x + t\Delta x - x) + \frac{1}{2} (x + t\Delta x - x)^T \nabla^2 f(x) (x + t\Delta x - x)$$

که ساده شده آن به‌صورت:

$$f(x + t\Delta x) = f(x) + t \nabla f(x)^T \Delta x + \frac{t^2}{2} \Delta x^T \nabla^2 f(x) \Delta x$$

است. با جایگذاری کران بالای ذکر شده در صورت سوال برای $\nabla^2 f(x)$:

$$\nabla^2 f(x) \preceq MI$$

[۲] Positive Semi-Definite (PSD) Ordering

و همچنین مقدار مناسب t، از نامساوی ذکر شده در صورت سوال:

$$t \leq -\frac{\nabla f(x)^T \Delta x}{M\|\Delta x\|^2}$$

رابطه ما به صورت زیر در خواهد آمد:

$$f(x+t\Delta x) \leq f(x) + t\nabla f(x)^T \Delta x - \frac{t}{2}\Delta x^T MI\Delta x(\frac{\nabla f(x)^T \Delta x}{M\|\Delta x\|^2})$$

بعد از فاکتورگیری خواهیم داشت:

$$f(x+t\Delta x) \leq f(x) + t\nabla f(x)^T \Delta x \left(1 - \frac{1}{2}\frac{\Delta x^T MI\Delta x}{M\|\Delta x\|^2}\right)$$

طبق خواص ضرب ماتریسی و از انجا که M اسکالر است، داریم:

$$\Delta x^T (MI)\, \Delta x \;=\; \Delta x^T \big(M\Delta x\big) \;=\; M\big(\Delta x^T \Delta x\big)$$

همچنین از آنجا که $\Delta x^T \Delta x = \|\Delta x\|^2$ داریم:

$$\Delta x^T (MI)\, \Delta x \;=\; M\,\|\Delta x\|^2$$

بنابراین:

$$\frac{\Delta x^T (MI)\, \Delta x}{M\,\|\Delta x\|^2} \;=\; \frac{M\|\Delta x\|^2}{M\|\Delta x\|^2} \;=\; 1$$

پس رابطه اصلی ما به صورت زیر در خواهد آمد:

$$f(x+t\Delta x) \leq f(x) + \frac{1}{2}t\nabla f(x)^T \Delta x$$

که اگر آلفا را برابر ۱/۲ در نظر بگیریم به شرط توقف نوشته شده در الگوریتم بالا می‌رسیم.

۲. با استفاده از نتیجه‌ی بخش قبل، یک کران بالا برای تعداد تکرارهای مورد نیاز در فرآیند Backtracking Line Search به‌دست آورید.
راهنمایی: مقدار t در هر تکرار با ضریب β کاهش می‌یابد. از این خاصیت استفاده کرده و لگاریتم بگیرید تا تعداد تکرارهای مورد نیاز را به‌دست آورید.

برای به‌دست آوردن کران بالا برای تعداد تکرارهای مورد نیاز در این فرآیند، از این خاصیت استفاده می‌کنیم که مقدار t در هر تکرار با ضریب β کاهش می‌یابد. بنابراین، مقدار t پس از k تکرار به‌صورت زیر خواهد بود:

$$t = \beta^k t_0$$

همچنین از قسمت قبل داریم:

$$t \leq -\frac{\nabla f(x)^T \Delta x}{M\|\Delta x\|^2}$$

پس:

$$\beta^k t_0 \leq -\frac{\nabla f(x)^T \Delta x}{M\|\Delta x\|^2}$$

حال با گرفتن لگاریتم طبیعی از دو طرف رابطه:

$$k \ln \beta + \ln t_0 \leq \ln(-\frac{\nabla f(x)^T \Delta x}{M\|\Delta x\|^2})$$

$$k \leq \frac{\ln(-\frac{\nabla f(x)^T \Delta x}{M\|\Delta x\|^2}) - \ln t_0}{\ln \beta}$$

پس کران بالای تعداد تکرارها برابر است با:

$$k \leq \frac{\ln(-\frac{\nabla f(x)^T \Delta x}{t_0 M\|\Delta x\|^2})}{\ln \beta}$$

پرسش ۴. Adam (۱۵ نمره)

در درس، الگوریتم بهینه‌سازی Adam معرفی شده است که با استفاده از میانگین‌های نمایی از گرادیان‌ها و مربع‌های آن‌ها، نرخ به‌روزرسانی پارامترها را بهبود می‌بخشد. الگوریتم Adam به صورت زیر ارائه شده است:

Adam Algorithm:

Require: α : Stepsize
Require: $\beta_1, \beta_2 \in [0, 1]$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters θ
Require: θ_0 : Initial parameter vector

Initialize:
$m_0 \leftarrow 0$ (Initialize 1^{st} moment vector)
$v_0 \leftarrow 0$ (Initialize 2^{nd} moment vector)
$t \leftarrow 0$ (Initialize timestep)

while θ_t not converged **do**
$t \leftarrow t + 1$
$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Obtain gradient at timestep t)
$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second moment estimate)
$\hat{m}_t \leftarrow m_t/(1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
$\hat{v}_t \leftarrow v_t/(1 - \beta_2^t)$ (Compute bias-corrected second moment estimate)
$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t/(\sqrt{\hat{v}_t} + \epsilon)$ (Update parameters)
end while
return θ_t (Resulting parameters)

۱. ابتدا توجه کنید که در الگوریتم Adam رابطه میانگین نمایی برای مربع گرادیان‌ها به صورت بازگشتی تعریف شده است:

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2.$$

حالا این رابطه را به صورت غیر بازگشتی بازنویسی کنید تا نشان داده شود که

$$v_t = (1 - \beta_2) \sum_{i=1}^{t} \beta_2^{t-i} g_i^2.$$

برای اثبات این رابطه به کمک استقرا، ابتدا نشان می‌دهیم که برای مقدار اولیه صدق می‌کند و سپس فرض می‌کنیم که برای مقدار $t = k$ برقرار است و نشان می‌دهیم که برای $t = k + 1$ نیز برقرار خواهد بود.
پایه: برای $t = 1$ از رابطه بازگشتی داریم:

$$v_1 = (1 - \beta_2) g_1^2$$

همچنین از رابطه‌ی غیر بازگشتی نیز داریم:

$$v_1 = (1 - \beta_2) \sum_{i=1}^{1} \beta_2^{1-i} g_i^2 = (1 - \beta_2) g_1^2$$

پس رابطه برای $t = 1$ برقرار است.

فرض استقرا: فرض می‌کنیم که رابطه برای $t = k$ برقرار است:

$$v_k = (1 - \beta_2) \sum_{i=1}^{k} \beta_2^{k-i} g_i^2$$

گام استقرا: نشان می‌دهیم که رابطه برای $t = k + 1$ نیز برقرار است. از رابطه بازگشتی داریم:

$$v_{k+1} = \beta_2 v_k + (1 - \beta_2) g_{k+1}^2$$

با جایگذاری فرض استقرا:

$$v_{k+1} = \beta_2 \left((1 - \beta_2) \sum_{i=1}^{k} \beta_2^{k-i} g_i^2 \right) + (1 - \beta_2) g_{k+1}^2$$

با پخش کرد β_2روی پرانتز داریم:

$$v_{k+1} = (1 - \beta_2) \sum_{i=1}^{k} \beta_2^{(k+1)-i} g_i^2 + (1 - \beta_2) g_{k+1}^2$$

که می‌توان آن را به صورت زیر بازنویسی کرد:

$$v_{k+1} = (1 - \beta_2) \sum_{i=1}^{k+1} \beta_2^{(k+1)-i} g_i^2$$

بنابراین، رابطه غیر بازگشتی برای میانگین نمایی مربع گرادیان‌ها به کمک استقرا اثبات شد:

$$v_t = (1 - \beta_2) \sum_{i=1}^{t} \beta_2^{t-i} g_i^2.$$

۲. فرض کنید گرادیان‌ها به صورت مستقل و با توزیع یکسان ($i.i.d.$) باشند. از رابطه‌ی غیر بازگشتی بدست آمده، امید ریاضی v_t را محاسبه کنید. منظور از $i.i.d.$ این است که

$$\mathbb{E}[v_t] = (1 - \beta_2) \sum_{i=1}^{t} \beta_2^{t-i} \, \mathbb{E}[g_i^2].$$

نتیجه نهایی نشان می‌دهد که
$$\mathbb{E}[v_t] = \mathbb{E}[g_t^2]\,(1-\beta_2^t).$$
در این بخش توضیح دهید که این نتیجه چه مفهومی دارد و چرا نشان‌دهنده بایاس در تخمین مربع گرادیان‌ها است.

همانطور که گفته شد، با گرفتن امید ریاضی از دو طرف معادله غیر بازگشتی که قبلاً به دست آمده و استفاده از خطی بودن امید ریاضی، خواهیم داشت:

$$\mathbb{E}[v_t] = (1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i}\mathbb{E}[g_i^2]$$

چون g_i ها به صورت .i.i.d در نظر گرفته شده‌اند، مقدار امید ریاضی مربع آن‌ها با یکدیگر برابر است. بنابراین می‌توان این مقدار ثابت را از سیگما خارج کرد:

$$\mathbb{E}[v_t] = (1-\beta_2)\mathbb{E}[g_t^2]\sum_{i=1}^{t}\beta_2^{t-i}$$

مجموع بالا یک سری هندسی است که مقدار آن برابر است با:

$$\sum_{i=0}^{t-1}\beta_2^i = \frac{1-\beta_2^t}{1-\beta_2}$$

در نتیجه، مقدار امید ریاضی v_t به‌صورت زیر به دست می‌آید:

$$\mathbb{E}[v_t] \;= \mathbb{E}[g_t^2]\,(1-\beta_2^t)$$

توضیح نتیجه: این نتیجه نشان می‌دهد که v_t مقدار $\mathbb{E}[g_t^2]$ را به‌درستی تخمین می‌زند، اما یک ضریب $(1-\beta_2^t)$ دارد که باعث ایجاد بایاس در تخمین مقدار واقعی $\mathbb{E}[g_t^2]$ می‌شود. این بایاس در مراحل ابتدایی آموزش محسوس‌تر است، اما با گذشت زمان و افزایش t، مقدار β_2^t به صفر میل می‌کند و تخمین دقیق‌تر می‌شود.

۳. بر اساس نتایج بخش قبل توضیح دهید که چرا اعمال مرحله‌ی Bias Correction در الگوریتم Adam ضروری است. به طور خلاصه بیان کنید که در ابتدای اجرای الگوریتم، مقادیر v_t (و همچنین m_t) به دلیل مقدار اولیه صفر به سمت صفر سوگیری دارند و Bias Correction باعث می‌شود که این سوگیری اصلاح شده و تخمین‌های به‌دست آمده دقیق‌تر باشند. همچنین توضیح دهید که تاثیر مقدار β_2 چگونه این سوگیری به سمت صفر را تشدید می‌کند.

همان‌طور که در بخش‌های قبل دیدیم، در ابتدای اجرای الگوریتم Adam مقادیر m_t و v_t (میانگین و میانگین مربعات گرادیان) به دلیل مقدار اولیه‌ی صفر به‌سمت صفر سوگیری دارند. در واقع، از رابطه‌های:

$$m_t = \beta_1\,m_{t-1} + (1-\beta_1)\,g_t, \quad v_t = \beta_2\,v_{t-1} + (1-\beta_2)\,g_t^2,$$

نتیجه می‌شود که در گام‌های اولیه، به‌علت ضرب‌شدن مکرر در ضرایب β_1 و β_2 و شروع از صفر، میانگین‌ها کوچکتر از مقدار واقعی خود تخمین زده می‌شوند. این پدیده یک بایاس (Bias) را در تخمین ایجاد می‌کند. برای رفع این بایاس، Adam مرحله‌ای را انجام می‌دهد که در آن تخمین‌های m_t و v_t را به‌صورت زیر اصلاح می‌کند:

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1-\beta_2^t}.$$

این کسرها باعث می‌شوند که عوامل $(1-\beta_1^t)$ و $(1-\beta_2^t)$ که در مراحل ابتدایی کوچک‌اند و موجب کوچک‌ترشدن تخمین‌ها شده بودند، حذف شوند. به این ترتیب، Bias Correction مقدار واقعی‌تری از

میانگین و میانگین مربعات گرادیان را ارائه می‌دهد.

تأثیر β_2 در شدت بایاس: پارامتر β_2 مشخص می‌کند که تا چه حد به مقادیر قبلی v_t اتکا شود. هرچه β_2 به ۱ نزدیک‌تر باشد، اثر داده‌های قدیمی‌تر بیشتر حفظ می‌شود و گرایش به سمت صفر در مراحل ابتدایی شدیدتر خواهد بود. در مقابل، هرچه β_2 کوچکتر باشد، تأثیر گرادیان‌های اخیر قوی‌تر و تخمین سریع‌تر به مقدار واقعی خود نزدیک می‌شود. در هر حال، مرحله‌ی Bias Correction باعث می‌شود که حتی با β_2 بزرگ هم در نهایت بایاس از بین برود (چراکه $(1-\beta_2^t)$ به‌تدریج به ۱ میل می‌کند).

به‌طور خلاصه، Bia Correction برای حذف بایاس ناشی از مقادیر اولیه‌ی صفر در m_t و v_t ضروری است و اجازه می‌دهد تخمین‌های به‌دست‌آمده در گام‌های اولیه، بیانگر بهتری از میانگین و میانگین مربعات گرادیان باشند. همچنین مقدار β_2 نقش تعیین‌کننده‌ای در میزان اتکا به تاریخچه‌ی گرادیان‌ها و شدت این بایاس دارد.

۴. در الگوریتم Adam، به‌روزرسانی هر وزن بر مبنای مقیاس کردن گرادیان‌ها با معکوس «نرم‌دو» گرادیان‌های فعلی و قبلی انجام می‌شود. حال با جایگزینی «نرم‌دو» با «نرم‌بی‌نهایت» (به این صورت که توان β_2 را برابر p در نظر گرفته و سپس $p \to \infty$ را اعمال کنید) می‌توانید الگوریتم بهینه‌سازی جدیدی به نام AdaMax را به دست آورید. راهنمایی: به مقاله اصلی Adam مراجعه کنید. در آن توضیح داده شده که

$$u_t = \lim_{p\to\infty}\left((1-\beta_2)\sum_{i=1}^{t}\beta_2^{p(t-i)}|g_i|^p\right)^{1/p}$$

که منجر به رابطه‌ی بازگشتی

$$u_t = \max\big(\beta_2 \cdot u_{t-1},\ |g_t|\big)$$

می‌شود. الگوریتم AdaMax به صورت زیر ارائه شده است:

AdaMax Algorithm:

Require: α : Stepsize
Require: $\beta_1, \beta_2 \in [0,1]$
Require: $f(\theta)$: Stochastic objective function
Require: θ_0 : Initial parameter vector

Initialize:
$m_0 \leftarrow 0$
$u_0 \leftarrow 0$ (Initialize the exponentially weighted infinity norm)
$t \leftarrow 0$

while θ_t not converged **do**
$t \leftarrow t+1$
$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$
$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1-\beta_1)\cdot g_t$
$u_t \leftarrow \max\big(\beta_2 \cdot u_{t-1},\ |g_t|\big)$
$\theta_t \leftarrow \theta_{t-1} - \Big(\alpha/(1-\beta_1^t)\Big)\cdot m_t/u_t$
end while
return θ_t

---

۵. الگوریتم حاصل را با الگوریتم Adam مقایسه کنید. به صورت مستقیم بیان کنید که در چه شرایطی استفاده از الگوریتم حاصل‌شده نسبت به Adam بهتر عمل می‌کند؟

در الگوریتم Adam از میانگین مربع گرادیان‌ها (یعنی v_t) برای تنظیم اندازه‌گام استفاده می‌شود؛ اما در AdaMax به‌جای آن از نرمِ بی‌نهایت گرادیان‌های گذشته استفاده می‌شود:

$$u_t = \max\big(\beta_2\, u_{t-1},\ |g_t|\big)$$

این تغییر باعث می‌شود که نوسان اندازه‌گام کمتر به مقادیر خاص گرادیان حساس باشد و به‌نوعی کلی‌تر عمل کند، زیرا نرم بی‌نهایت فقط به بزرگترین مؤلفه گرادیان‌ها توجه می‌کند.

مقایسه و شرایط استفاده:

– **میزان پایداری:** در مسائلی که مؤلفه‌های گرادیان دارای توزیع غیرهمگن باشند و ممکن است یکی از مؤلفه‌ها ناگهان مقدار بزرگی پیدا کند، استفاده از نرم بی‌نهایت می‌تواند به **پایداری بیشتر** کمک کند؛ چراکه فقط بزرگترین مقدار گرادیان را دنبال می‌کند.

– **نوسان کمتر:** AdaMax معمولاً در مسائلی که دامنه گرادیان بسیار وسیع و ناهمگن است، **نوسانات** کمتری در اندازه‌گام ایجاد می‌کند.

– **نقطه ضعف احتمالی:** اگر مؤلفه‌های کوچک گرادیان هم اهمیت زیادی داشته باشند، تمرکز صرف بر بزرگترین مؤلفه گرادیان ممکن است باعث شود که بعضی جهت‌های مهم اما با اندازه کوچک، به‌خوبی در به‌روزرسانی وزن‌ها لحاظ نشوند.

– **شرایط برتری:** در شبکه‌های عمیق با ابعاد بالا و گرادیان‌هایی که گاهی اوقات مؤلفه‌های بسیار بزرگی دارند (و بقیه مؤلفه‌ها کوچک هستند)، AdaMax ممکن است از Adam **عملکرد بهتری** داشته باشد، زیرا کمتر به اندازه‌های نسبی بین مؤلفه‌های گرادیان حساس است.

– **سرعت همگرایی:** هرچند AdaMax و Adam هر دو جزو روش‌های تطبیقی مؤثر هستند، اما در عمل ممکن است AdaMax همگرایی کمی کندتری نسبت به Adam داشته باشد..

پرسش ۵. بهینه‌سازی مرتبه دوم (۱۰ نمره)

برای یک تابع دلخواه $f(x)$ می‌توان از سری Taylor Series مرتبه دوم در نزدیکی نقطه x_0 استفاده کرد:

$$f(x) \approx f(x_0) + (x - x_0)^\top \nabla f(x_0) + \frac{1}{2}(x - x_0)^\top H (x - x_0),$$

که در اینجا:

- $\nabla f(x_0)$ بردار gradient تابع در نقطه x_0 است.
- H ماتریس هسین[۳] تابع f در نقطه x_0 است که به صورت

$$H_{i,j} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$$

تعریف می‌شود.

[۳] Hessian Matrix

۱. فرض کنید نرخ یادگیری ϵ در الگوریتم Descent Gradient به صورت $x = x_0 - \epsilon\, g,$ به کار می‌رود، که در آن $g = \nabla f(x_0)$ می‌باشد. طبق بسط تیلور، مقدار $f(x_0 - \epsilon\, g)$ را به صورت تقریبی تا جمله مرتبه سوم (یعنی با باقی‌مانده‌ای از مرتبه $O(\epsilon^3)$) بنویسید.

با قرار دادن $x = x_0 - \epsilon g$ در بسط تیلور داریم:

$$x - x_0 = -\,\epsilon\, g$$

بنابراین:

$$f\big(x_0 - \epsilon g\big) \;\approx\; f(x_0) \;+\; \underbrace{(-\,\epsilon\, g)^{\top}\nabla f(x_0)}_{\text{جمله مرتبه اول}} \;+\; \frac{1}{2}\, \underbrace{(-\,\epsilon\, g)^{\top}\, H\, (-\,\epsilon\, g)}_{\text{جمله مرتبه دوم}} \;+\; O(\epsilon^3)$$

$$= f(x_0) \;-\; \epsilon\, g^{\top}\nabla f(x_0) \;+\; \frac{1}{2}\,\epsilon^2\, g^{\top}\, H\, g \;+\; O(\epsilon^3)$$

در نتیجه، بر اساس بسط تیلور تا جمله مرتبه دوم (به‌همراه باقی‌مانده از مرتبه $O(\epsilon^3)$)، داریم:

$$f\big(x_0 - \epsilon g\big) \approx f(x_0) \;-\; \epsilon\, g^{\top}\nabla f(x_0) \;+\; \frac{1}{2}\,\epsilon^2\, g^{\top}\, H\, g \;+\; O(\epsilon^3)$$

۲. در صورتی که $g^{\top}Hg$ منفی یا صفر باشد، افزایش یا کاهش نرخ یادگیری ϵ چه تأثیری بر مقدار تقریبی $f(x)$ خواهد داشت؟ چرا استفاده از این روش بهینه‌سازی در چنین شرایطی مناسب نیست؟

تأثیر ϵ بر مقدار تابع به شکل خطی (مرتبه اول) و مربعی (مرتبه دوم) ظاهر می‌شود. در صورتی‌که ϵ بسیار کوچک یا صفر باشد، تغییر در $f\big(x_0 - \epsilon\, g\big)$ ناچیز می‌شود و فرایند یادگیری (به‌روزرسانی پارامترها) عملاً متوقف یا بسیار کُند خواهد شد. از سوی دیگر، اگر ϵ بیش‌ازحد بزرگ باشد، ممکن است مقدار تابع افزایش یابد یا دچار نوسانات شدید شود.

- اگر $g^{\top}H\, g \le 0$ باشد، هسین در جهت g **نیمه‌منفی** یا Negative semi-definite است. این بدان معناست که در جهت بردار گرادیان، تابع رفتار محدبی (convex) ندارد و حرکت در آن جهت لزوماً منجر به کاهش مقدار تابع نخواهد شد.
- در روش‌های گرادیان نزولی ساده که از تقریب درجه دوم (یا همان بسط تیلور) استفاده می‌کنند، فرض بر این است که H در ناحیه مورد نظر **مثبت معین** است تا تضمین کند حرکت در جهت $-g$ مقدار تابع را کم می‌کند.
- هنگامی که $g^{\top}H\, g \le 0$ باشد یا هسین نامعین (indefinite) باشد، هیچ تضمینی برای کاهش f وجود ندارد و ممکن است الگوریتم به‌جای رسیدن به کمینه، به ناحیه‌ای نامناسب هدایت شود یا دچار رفتار ناپایدار گردد.

۳. اگر $g^{\top}Hg$ مثبت باشد، با مشتق‌گیری از عبارت ارائه‌شده نسبت به ϵ و قرار دادن آن برابر صفر، نرخ یادگیری بهینه $\epsilon^{\star}$ را به دست آورید.

از تقریب بسط تیلور مرتبه دوم داریم:

$$f(x_0 - \epsilon g) \approx f(x_0) - \epsilon g^{\top}\nabla f(x_0) + \frac{1}{2}\epsilon^2 g^{\top}Hg + O(\epsilon^3)$$

برای یافتن مقدار بهینه ϵ^*، تابع $f(x)$ را نسبت به ϵ مشتق گرفته و مقدار بهینه آن را تعیین می‌کنیم:

$$\frac{d}{d\epsilon}\left(f(x_0-\epsilon g)\right) = -g^\top \nabla f(x_0) + \epsilon g^\top H g$$

با قرار دادن مشتق برابر صفر، مقدار ϵ^* به‌دست می‌آید:

$$\epsilon^* = \frac{g^\top g}{g^\top H g}$$

از آنجا که در فرض مسئله $g^\top H g > 0$ است، مقدار ϵ^* مثبت خواهد بود.

۴. با توجه به رابطه به دست آمده برای $\epsilon^\star$، توضیح دهید که مقدار نرخ یادگیری بهینه $\epsilon^\star$ در چه شرایطی بیشینه و کمینه خواهد شد.
راهنمایی: Hx را به‌صورت ترکیب خطی از بردارهای ویژه‌ی H و مقدار ویژه‌های متناظر آن بنویسید.

برای تحلیل مقدار ϵ^*، از فرم مقدار ویژه هسین H استفاده می‌کنیم. فرض کنیم H دارای مقادیر ویژه $\lambda_1, \lambda_2, \ldots, \lambda_n$ و بردارهای ویژه متناظر $v_1, v_2, \ldots, v_n$ باشد، بنابراین هر بردار دلخواه مانند g را می‌توان به‌صورت ترکیب خطی از بردارهای ویژه نوشت:

$$g = \sum_{i=1}^{n} \alpha_i v_i$$

با اعمال H بر روی g داریم:

$$Hg = H\sum_{i=1}^{n} \alpha_i v_i = \sum_{i=1}^{n} \alpha_i H v_i = \sum_{i=1}^{n} \alpha_i \lambda_i v_i$$

بنابراین، مقدار $g^\top H g$ به‌صورت زیر محاسبه می‌شود:

$$g^\top H g = \sum_{i=1}^{n} \alpha_i^2 \lambda_i$$

از طرفی، چون $\nabla f(x_0)$ هم یک ترکیب خطی از بردارهای ویژه است، مقدار $g^\top \nabla f(x_0)$ نیز ترکیبی از ضرایب α_i و مقادیر ویژه خواهد بود. در نتیجه، مقدار ϵ^* به مقادیر ویژه H بستگی دارد:

$$\epsilon^* = \frac{\sum_{i=1}^{n} \alpha_i^2}{\sum_{i=1}^{n} \alpha_i^2 \lambda_i}$$

- مقدار ϵ^* زمانی بیشینه خواهد شد که مقدار مخرج (یعنی $\sum_{i=1}^{n} \alpha_i^2 \lambda_i$) کمینه باشد، که این حالت زمانی رخ می‌دهد که بیشترین بخش بردار g در راستای کوچک‌ترین مقدار ویژه $\lambda_{\min}$ قرار بگیرد.
- برعکس، مقدار ϵ^* زمانی کمینه می‌شود که g در راستای بزرگ‌ترین مقدار ویژه $\lambda_{\max}$ قرار بگیرد.

بنابراین، مقدار بهینه نرخ یادگیری ϵ^* بین دو کران وابسته به مقادیر ویژه ماتریس هسین قرار دارد:

$$\frac{1}{\lambda_{\max}} \le \epsilon^* \le \frac{1}{\lambda_{\min}}.$$

پرسش ۶. Regularization (۲۰ نمره)

اگر برای تابعی نظیر $f:\mathbb{R}^n \to \mathbb{R}$ داشته باشیم:

$$\|f(x_1) - f(x_2)\| \leq L\|x_1 - x_2\|, \qquad \forall x_1,\ x_2 \in \mathbb{R}^n$$

می‌گوییم این تابع L-Lipschitz است. برای درک بهتر این مفهوم $x_1 = x_2 + \delta$ را در نظر داشته باشید. اکنون می‌توان ادعا کرد که برای هر x_2 و به ازای هر δ دلخواه داریم:

$$\|f(x_2 + \delta) - f(x_2)\| \leq L\|\delta\|$$

یعنی اگر به ازای هر مقدار ورودی نظیر x، آن را به اندازه δ تغییر دهیم، میزان تغییرات در خروجی تابع کمتر از $L\|\delta\|$ خواهیم داشت و به بیانی یک کران بالا برای میزان تغییرات تعریف کردیم.
می‌دانیم که در هر لایه از یک شبکه عصبی داریم:

$$x^{(\ell+1)} = h^{(\ell)}(W^{(\ell)}x^{(\ell)})$$

که در اینجا $h(\cdot)$ یک تابع فعال‌سازی[۴] دلخواه است. حال به پرسش‌های زیر پاسخ دهید.

۱. با نوشتن رابطه فوق برای ضرب ماتریسی Wx نشان دهید که ثابت Lipschitz برای این ماتریس برابر با نرم الحاقی[۵] آن است. یعنی:

$$L = \max_{\delta \neq 0} \frac{\|W\delta\|}{\|\delta\|} = \|W\|$$

از تعریف صورت سوال برای تابع L-Lipschitz برای تابع $f(x) = Wx$، داریم:

$$\|Wx_1 - Wx_2\| \leq L\|x_1 - x_2\|$$

با قرار دادن $\delta = x_1 - x_2$ داریم:

$$\|W\delta\| \leq L\|\delta\| \quad \text{برای هر } \delta \in \mathbb{R}^n$$

بهترین مقدار L، یعنی کوچک‌ترین عددی که برای همه $\delta \neq 0$ نابرابری بالا برقرار باشد، به صورت زیر تعریف می‌شود:

$$L = \sup_{\delta \neq 0} \frac{\|W\delta\|}{\|\delta\|}$$

بر اساس تعریف، $\|W\|$ به عنوان نرم الحاقی یا طیفی W به صورت زیر تعریف می‌شود:

$$\|W\| = \sup_{\delta \neq 0} \frac{\|W\delta\|}{\|\delta\|}$$

بنابراین ثابت لیپ‌شیتز (L) تابع $f(x) = Wx$ برابر با $\|W\|$ است.

[۴] Activation Function
[۵] Spectral Norm

۲. با استفاده از تجزیه مقادیر تکین[۶] این ماتریس نشان دهید که ثابت Lipschitz این ماتریس برابر است با بزرگ‌ترین مقدار تکین آن.

فرض کنید ماتریس W دارای تجزیه مقدار تکین به صورت زیر باشد:

$$W = U\Sigma V^\top,$$

که در آن:

- U و V ماتریس‌های اورتوگونال هستند ($U^\top U = I$ و $V^\top V = I$).
- Σ یک ماتریس قطری است که مقادیر تکین $\sigma_1, \sigma_2, \dots, \sigma_r$ را دربردارد، به طوری که:

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$$

برای هر بردار غیرصفر $\delta \in \mathbb{R}^n$ داریم:

$$\|W\delta\| = \|U\Sigma V^\top \delta\|$$

از آنجا که U یک ماتریس اورتوگونال است:

$$\|UX\|^2 = (UX)^\top (UX) = X^\top U^\top U X = X^\top I X = \|X\|^2$$

پس:

$$\|UX\| = \|X\|$$

در نتیجه:

$$\|U\Sigma V^\top \delta\| = \|\Sigma V^\top \delta\|$$

حال، با تعریف $\xi = V^\top \delta$، می‌توانیم رابطه اول را به صورت زیر بنویسیم:

$$\|W\delta\| = \|\Sigma \xi\|$$

از آنجا که V نیز یک ماتریس اورتوگونال است، این تبدیل نیز نُرم را حفظ می‌کند:

$$\|\xi\| = \|V^\top \delta\| = \|\delta\|$$

با جایگذاری $\|\xi\| = \|\delta\|$ در رابطه قبل، به دست می‌آید:

$$\frac{\|W\delta\|}{\|\delta\|} = \frac{\|\Sigma\xi\|}{\|\xi\|}$$

با توجه به ساختار قطری Σ، اگر $\xi = (\xi_1, \xi_2, \dots, \xi_r)^\top$ باشد، خواهیم داشت:

$$\|\Sigma\xi\|^2 = \sum_{i=1}^{r} \sigma_i^2\, \xi_i^2 \quad \text{و} \quad \|\xi\|^2 = \sum_{i=1}^{r} \xi_i^2$$

پس می‌توان نوشت:

$$\frac{\|W\delta\|}{\|\delta\|} = \sqrt{\frac{\sum_{i=1}^{r} \sigma_i^2\, \xi_i^2}{\sum_{i=1}^{r} \xi_i^2}}$$

[۶] Singular Value Decomposition (SVD)

چون σ_1 بزرگ‌ترین مقدار تکین است، بیشینهٔ این نسبت زمانی حاصل می‌شود که تمام "وزن" ξ در مؤلفهٔ متناظر با σ_1 متمرکز باشد. بنابراین، اگر:

$$\xi = (1, 0, \dots, 0)^\top,$$

داشته باشیم:

$$\frac{\|W\delta\|}{\|\delta\|} = \frac{\|\Sigma\xi\|}{\|\xi\|} = \frac{\sqrt{\sigma_1^2 \cdot 1^2 + 0 + \cdots + 0}}{1} = \sigma_1$$

پس داریم:

$$L = \sup_{\delta \neq 0} \frac{\|W\delta\|}{\|\delta\|} = \sigma_1$$

۳. در صورتی که پس از ضرب ماتریس از یک تابع فعال‌سازی نظیر $h(\cdot)$ استفاده کنیم، ثابت Lipschitz آن چه تغییری خواهد کرد؟ (کافیست برای توابع داده شده مقدار جدید را برای $h(Wx)$ بدست آورید.)

$$h_1(z) = \mathrm{ReLU}(z) = \max(0, z)$$
$$h_2(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$
$$h_3(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

فرض کنید توابع $g : \mathbb{R}^n \to \mathbb{R}^m$ و $f : \mathbb{R}^m \to \mathbb{R}^p$ به ترتیب دارای ضرایب لیپ‌شیتز L_g و L_f باشند، یعنی:

$$\|g(x_1) - g(x_2)\| \leq L_g \|x_1 - x_2\|, \quad \forall x_1, x_2 \in \mathbb{R}^n,$$

$$\|f(y_1) - f(y_2)\| \leq L_f \|y_1 - y_2\|, \quad \forall y_1, y_2 \in \mathbb{R}^m.$$

هدف ما یافتن ضریب لیپ‌شیتز ترکیب $f(g(x))$ است. برای هر دو نقطه $x_1, x_2 \in \mathbb{R}^n$ با اعمال خاصیت لیپ‌شیتز برای f داریم:

$$\|f(g(x_1)) - f(g(x_2))\| \leq L_f \|g(x_1) - g(x_2)\|$$

و سپس استفاده از خاصیت لیپ‌شیتز برای g داریم:

$$\|g(x_1) - g(x_2)\| \leq L_g \|x_1 - x_2\|$$

در نتیجه:

$$\|f(g(x_1)) - f(g(x_2))\| \leq L_f L_g \|x_1 - x_2\|$$

پس، ترکیب $f(g(x))$ نیز لیپ‌شیتز است و ضریب آن برابر است با:

$$L_{f \circ g} = L_f L_g$$

با توجه به خاصیت اثبات‌شده، ثابت لیپ‌شیتز تابع ترکیبی $h(Wx)$ برابر است با:

$$L = L_h \cdot L_W$$

که در آن:

- L_W ثابت لیپ‌شیتز ماتریس W است که در مرحله‌ی قبل نشان دادیم برابر با بزرگ‌ترین مقدار تکین σ_1 است، یعنی:

$$L_W = \sigma_1$$

- L_h ثابت لیپ‌شیتز تابع فعال‌سازی $h(\cdot)$ است که در ادامه برای توابع مختلف محاسبه می‌شود.

ثابت لیپ‌شیتز تابع فعال‌سازی $h(x)$ برابر است با بیشترین مقدار مشتق آن در دامنه‌ی مورد نظر، زیرا:

$$|h(x_1) - h(x_2)| \leq L_h |x_1 - x_2|, \quad \forall x_1, x_2$$

چون تابع $h(x)$ در ناحیه موردنظر مشتق‌پذیر است، از قضیه مقدار میانگین استفاده می‌کنیم:

$$h(x_1) - h(x_2) = h'(\xi)(x_1 - x_2),$$

که در آن ξ عددی بین x_1 و x_2 است. با گرفتن قدر مطلق از دو طرف رابطه:

$$|h(x_1) - h(x_2)| = |h'(\xi)| \cdot |x_1 - x_2|$$

با جایگذاری سمت راست عبارت بالا در نامساوی حاصل از خاصیت لیپ‌شیتز تابع h(x) داریم:

$$|h'(\xi)| \cdot |x_1 - x_2| \leq L_h |x_1 - x_2|$$

با حذف قدر مطلق تفاضل از رابطه بالا، داریم:

$$|h'(\xi)| \leq L_h$$

در نتیجه:

$$L_h = \sup_{\xi} |h'(\xi)|$$

پس برای توابع مختلف داریم:

- تابع **ReLU** $h(x) = \max(0, x)$ دارای مشتق زیر است:

$$h'(x) = \begin{cases} 1 & x > 0, \\ 0 & x \leq 0 \end{cases}$$

بنابراین:

$$L_h = \sup_{x} |h'(x)| = 1$$

- تابع **tanh** $h(x) = \tanh(x)$ دارای مشتق:

$$h'(x) = 1 - \tanh^2(x)$$

حداکثر مقدار این مشتق زمانی حاصل می‌شود که $\tanh^2(x) = 0$، یعنی وقتی $x = 0$. پس:

$$L_h = \sup_{x} |h'(x)| = 1$$

- تابع **Sigmoid** $h(x) = \frac{1}{1+e^{-x}}$ دارای مشتق:

$$h'(x) = h(x)(1 - h(x))$$

بیشترین مقدار این مشتق زمانی حاصل می‌شود که $h(x) = \frac{1}{2}$، یعنی $x = 0$. در این حالت:

$$h'(0) = \frac{1}{4}$$

بنابراین:

$$L_h = \sup_x |h'(x)| = \frac{1}{4}$$

با توجه به رابطه $L = L_h \cdot L_W$، داریم:

$$L = \begin{cases} \sigma_1, & h(x) = \text{tanh یا ReLU}, \\ \frac{\sigma_1}{4}, & h(x) = \text{sigmoid}. \end{cases}$$

یعنی:

- برای ReLU و tanh، ثابت لیپ‌شیتز تغییری نمی‌کند و برابر با σ_1 باقی می‌ماند.
- برای sigmoid، ثابت لیپ‌شیتز با ضریب $\frac{1}{4}$ کاهش می‌یابد و برابر $\frac{\sigma_1}{4}$ می‌شود.

۴. با استفاده از نتایج بخش‌های قبل، با فرض استفاده از توابع فعال‌سازی ReLU در تمامی لایه‌های یه شبکه عصبی دارای n لایه، کران بالایی برای ثابت Lipschitz آن ارائه دهید.

یک شبکه عصبی عمیق دارای n لایه را در نظر بگیرید که در هر لایه، ابتدا یک تبدیل خطی با ماتریس وزن W_i انجام شده و سپس تابع فعال‌سازی ReLU اعمال می‌شود:

$$x^{(i)} = h(W_i x^{(i-1)}), \quad \text{برای } i = 1, 2, \dots, n$$

که در آن $x^{(i)}$ خروجی لایه‌ی iـام و $h(\cdot)$ تابع فعال‌سازی ReLU است. خروجی نهایی شبکه به‌صورت:

$$f(x) = h(W_n h(W_{n-1} \dots h(W_1 x)))$$

همان‌طور که در بخش‌های قبل نشان داده شد، ثابت لیپ‌شیتز یک تبدیل W_i برابر است با:

$$L_{W_i} = \sigma_1^{(i)},$$

که $\sigma_1^{(i)}$ بزرگ‌ترین مقدار تکین ماتریس W_i است. همچنین ثابت لیپ‌شیتز تابع ReLU برابر با $L_h = 1$ است، بنابراین برای هر لایه‌ی i خواهیم داشت:

$$L_i = L_h \cdot L_{W_i} = \sigma_1^{(i)}$$

ثابت لیپ‌شیتز یک ترکیب متوالی از توابع به‌صورت زیر محاسبه می‌شود:

$$L = L_n \cdot L_{n-1} \cdots L_2 \cdot L_1$$

با جایگذاری $L_i = \sigma_1^{(i)}$ برای هر لایه، نتیجه می‌گیریم که:

$$L = \sigma_1^{(n)} \cdot \sigma_1^{(n-1)} \cdots \sigma_1^{(2)} \cdot \sigma_1^{(1)}$$

چون هر مقدار تکین $\sigma_1^{(i)}$ مقدار غیرمنفی است، کران بالای ثابت لیپ‌شیتز شبکه به‌صورت زیر به‌دست می‌آید:

$$L \leq \prod_{i=1}^{n} \sigma_1^{(i)}$$

این مقدار نشان می‌دهد که هرچه تعداد لایه‌ها بیشتر شود، مقدار L می‌تواند رشد نمایی داشته باشد، مگر این‌که مقدار تکین بزرگ‌ترین مقدار ماتریس‌های وزنی محدود باشد.

۵. در صورتی که داده‌های ما دارای مقداری نویز باشد، یعنی $\tilde{x} = x + \epsilon$ باشد، توضیح دهید چگونه Weight Decay می‌تواند به ما در رسیدن به نتایج بهتر کمک کند؟ (راهنمایی: با استفاده از نتایج بخش‌های قبل و با مقایسه پیش‌بینی مدل برای $y = Wx$ و $\tilde{y} = W\tilde{x}$ به این سوال پاسخ دهید.)

فرض کنید داده‌های ورودی دارای نویز باشند، یعنی به‌جای مشاهده‌ی مقدار واقعی x، داده‌ی مشاهده‌شده برابر است با:

$$\tilde{x} = x + \epsilon$$

که در آن ϵ نویز موجود در داده‌ها است. در این حالت، خروجی مدل برای داده‌های واقعی و نویزی به‌ترتیب برابر خواهد بود با:

$$y = Wx, \quad \tilde{y} = W\tilde{x} = W(x + \epsilon) = Wx + W\epsilon$$

بنابراین، خطای ناشی از نویز در خروجی مدل برابر است با:

$$\tilde{y} - y = W\epsilon$$

همان‌طور که در نتایج قبلی نشان دادیم، ثابت لیپ‌شیتز ماتریس W برابر با بزرگ‌ترین مقدار تکین آن است:

$$L = \sigma_1(W)$$

این ثابت تعیین می‌کند که چگونه تغییرات در ورودی (ϵ) به خروجی انتقال می‌یابد:

$$\|W\epsilon\| \leq \sigma_1(W)\|\epsilon\|$$

اگر مقدار $\sigma_1(W)$ بزرگ باشد، اثر نویز ϵ در خروجی مدل تقویت شده و منجر به پیش‌بینی‌های ناپایدار می‌شود.

Weight Decay یک تکنیک منظم‌سازی است که به تابع هزینه یک جریمه اضافه می‌کند تا از بزرگ شدن مقادیر وزن‌ها جلوگیری شود:

$$\mathcal{L}_{\text{reg}} = \mathcal{L} + \lambda\|W\|_F^2$$

افزودن این جریمه باعث کاهش مقدار عناصر ماتریس W شده و در نتیجه مقدار تکین بزرگ‌ترین مقدار ماتریس W کاهش می‌یابد. با توجه به رابطه‌ی قبلی:

$$\|W\epsilon\| \leq \sigma_1(W)\|\epsilon\|$$

مشاهده می‌شود که کاهش مقدار $\sigma_1(W)$ باعث کاهش حساسیت مدل به نویز خواهد شد.

Weight Decay با کاهش مقدار تکین بزرگ‌ترین مقدار ماتریس وزن W، ثابت لیپ‌شیتز مدل را کاهش می‌دهد. این کاهش باعث می‌شود که مدل به نویز حساسیت کمتری داشته باشد و خروجی‌های پایدارتر و تعمیم‌پذیرتری ارائه دهد. بنابراین، Weight Decay یک روش مؤثر برای مقابله با نویز در داده‌ها و جلوگیری از بیش‌برازش است.