# Matrix Differentiation

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:

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

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.

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:

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

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

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

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

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

فرض کنید $a, x \in \mathbb{R}^n$ باشند. می‌خواهیم مشتق عبارت $f(x) = a^\top x$ را نسبت به $x$ پیدا کنیم. ابتدا تابع را به‌صورت مؤلفه‌ای می‌نویسیم:

$$f(x) = \sum_{i=1}^{n} a_i x_i$$

حال مشتق این تابع را نسبت به $x_j$ محاسبه می‌کنیم:

$$\frac{\partial f}{\partial x_j} = \frac{\partial}{\partial x_j} \sum_{i=1}^{n} a_i x_i = a_j$$

بنابراین، ماتریس ژاکوبین به‌صورت زیر خواهد بود:

$$J = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} = a^\top$$

چون $x^\top a = a^\top x$ است، مشتق آن نیز مشابه خواهد بود.

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

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

را بیابید.

$$f(x) = Ax$$

مولفه $i$ام این تابع برابر است با:

$$f_i(x) = \sum_{j=1}^{n} A_{ij} x_j$$

با مشتق‌گیری از این عبارت نسبت به $x_k$ داریم:

$$J_{ik} = \frac{\partial f_i}{\partial x_k} = A_{ij} \to J = A$$

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.

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.

$A \in \mathbb{R}^{n\times n}$ و $x \in \mathbb{R}^n$ باشد، تابع به‌صورت زیر تعریف می‌شود:

$$f(x) = x^\top Ax = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i A_{ij} x_j$$

حال مشتق این عبارت نسبت به $x_k$ را می‌گیریم:

$$\frac{\partial f}{\partial x_k} = \sum_{j=1}^{n} A_{kj}x_j + \sum_{i=1}^{n} A_{ik}x_i$$

که به صورت ماتریسی نوشته می‌شود:

$$\nabla f(x) = (A + A^\top)x \to J = x^\top(A^\top + A)$$

اگر ماتریس داده شده متقارن بود، یعنی $A = A^\top$، در این صورت گرادیان برابر با $2Ax$ می‌شد.

مشتق این تابع نسبت به $A$ برابر است با:

$$\frac{\partial x^\top Ax}{\partial A_{ij}} = x_i x_j \to \frac{\partial x^\top Ax}{\partial A} = xx^\top$$

ابعاد اصلی این خروجی $\mathbb{R}^{1\times n\times n}$ می‌باشد ولی برای سادگی آن را به صورت یک ماتریس $\mathbb{R}^{n\times n}$ در نظر می‌گیریم.

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

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

را محاسبه کنید.

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

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

را محاسبه کنید.

با استفاده از خاصیت مشتق‌گیری از ردگیری ماتریس داریم:

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

# Backpropagation

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

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

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

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

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:

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

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

با استفاده از مشتق ReLU و سیگموئید:

$$\frac{\partial l^{(21)}}{\partial W^{(21)}} = \mathbf{diag}(\mathbb{1}(l^{(21)} > 0)) \cdot l^{(1)\top}$$

$$\frac{\partial l^{(22)}}{\partial W^{(22)}} = \mathbf{diag}(\sigma(W^{(22)}l^{(1)})(1 - \sigma(W^{(22)}l^{(1)}))) \cdot l^{(1)\top}$$

در نتیجه:

$$\frac{\partial \mathcal{L}}{\partial W^{(21)}} = \frac{\partial \mathcal{L}}{\partial l^{(21)}} \cdot l^{(1)\top}$$

$$\frac{\partial \mathcal{L}}{\partial W^{(22)}} = \frac{\partial \mathcal{L}}{\partial l^{(22)}} \cdot l^{(1)\top}$$

با توجه به روابط قبلی، مشتق نسبت به $l^{(1)}$ به‌صورت زیر محاسبه می‌شود:

$$\frac{\partial \mathcal{L}}{\partial l^{(1)}} = (W^{(21)})^\top \frac{\partial \mathcal{L}}{\partial l^{(21)}} + (W^{(22)})^\top \frac{\partial \mathcal{L}}{\partial l^{(22)}}$$

چون داریم:

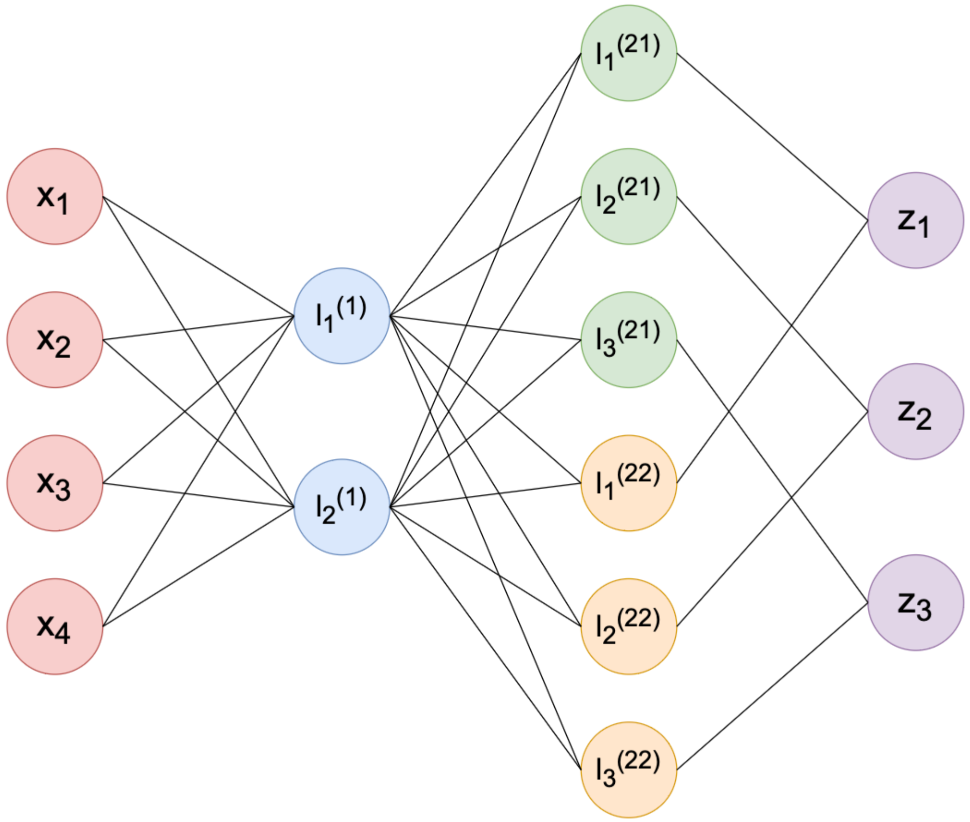
$$l^{(1)} = \text{ReLU}(W^{(1)}x)$$

پس:

$$\frac{\partial l^{(1)}}{\partial W^{(1)}} = \mathbf{diag}(\mathbb{1}(l^{(1)} > 0)) \cdot x^\top$$

در نتیجه:

$$\frac{\partial \mathcal{L}}{\partial W^{(1)}} = \frac{\partial \mathcal{L}}{\partial l^{(1)}} \cdot x^\top$$




$$l^{(21)} = \text{ReLU}(W^{(21)}l^{(1)})$$

$$l^{(22)} = \sigma(W^{(22)}l^{(1)})$$

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.

داریم:

$$l^{(1)} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad l^{(21)} = \begin{bmatrix} 3 \\ 0 \\ 1 \end{bmatrix}, \quad l^{(2)} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}, \quad z = \begin{bmatrix} 3 \\ 0.5 \\ 1 \end{bmatrix}$$

حال می‌توان گفت:

$$\textbf{softmax}(z) \approx \begin{bmatrix} 0.82 \\ 0.07 \\ 0.11 \end{bmatrix} \to \mathrm{CE}(\hat{y}, y) = -\log(\hat{y}_2) \approx 2.7$$

پس:

$$\frac{\partial \mathcal{L}}{\partial W^{(1)}} = \begin{bmatrix} -0.22 & -0.44 & -0.66 & -0.22 \\ 1.4 & 2.8 & 4.2 & 1.4 \end{bmatrix}$$

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

$$\frac{\partial \mathcal{L}}{\partial W^{(22)}} = \begin{bmatrix} 0 & 0 \\ -0.23 & -0.46 \\ 0 & 0 \end{bmatrix}$$

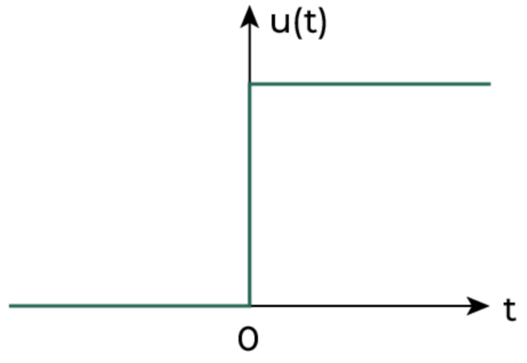
۱. زمانی که از روش Stochastic Gradient Descent استفاده می‌کنیم. ممکن است شاهد افزایش مقدار تابع هزینه پس از بروزرسانی وزن‌ها باشیم. چرا این اتفاق ممکن است رخ دهد؟ (دو دلیل را ذکر کنید.)

- نرخیادگیری بزرگ
- گرادیانهای نویزی

۲. برای گیت‌های منطقی AND و OR با سه ورودی، شبکه‌ی پرسپترون طراحی کنید.

۲. برای گیت‌های منطقی AND و OR با سه ورودی، شبکه‌ی پرسپترون طراحی کنید.

تابع فعال ساز: Step function




$$u(t) = \begin{cases} 1 & t > 0 \\ 0 & t > 0 \end{cases}$$

W1,w2, ... و bias باید طوری انتخاب بشود که به کمک تابع فعال ساز مقادیر جدول‌های روبرو تولید بشود

AND

| ورودی 1 | ورودی 2 | خروجی |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

OR

| ورودی 1 | ورودی 2 | خروجی |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

۳. در کلاس دیدیم با نگاه احتمالاتی به مسئله‌ی رگرشن و با فرض $p(y|f(x;W)) \sim \mathcal{N}(f(x;W), \sigma^۲)$ می‌توان بیشینه‌کردن درستنمایی را معادل با تابع‌هزینه‌ی میانگین مربعات خطا در نظر گرفت. حال فرض کنید $p(y|f(x;W)) \sim \mathcal{L}(f(x;W), b)$ آمده باشد.
رابطه‌ی توزیع لاپلاس به صورت زیر است:

$$\mathcal{L}(\mu, b) = \frac{۱}{۲b} exp(-\frac{|x-\mu|}{b}), \ b > ۰ \tag{۱}$$

در این صورت تابع هزینه‌ای که از بیشینه کردن درستنمایی بدست می‌آید به چه صورت است؟

۳. در کلاس دیدیم با نگاه احتمالاتی به مسئله‌ی رگرشن و با فرض $p(y|f(x;W)) \sim \mathcal{N}(f(x;W), \sigma^۲)$ می‌توان بیشینه‌کردن درستنمایی را معادل با تابع‌هزینه‌ی میانگین مربعات خطا در نظر گرفت. حال فرض کنید $p(y|f(x;W)) \sim \mathcal{L}(f(x;W), b)$ آمده باشد. رابطه‌ی توزیع لاپلاس به صورت زیر است:

$$\mathcal{L}(\mu, b) = \frac{۱}{۲b} exp(-\frac{|x-\mu|}{b}),\ b > ۰ \qquad (۱)$$

در این صورت تابع هزینه‌ای که از بیشینه کردن درستنمایی بدست می‌آید به چه صورت است؟

$D \sim L(f(x;\omega), b)$

Liklihood

$$\prod_{i=1}^{N} p(y^{(i)} | f(x^{(i)};\omega), b)$$

log liklihood

$$= \sum_{i=1}^{N} \log p(y^{(i)} | f(x^{(i)};\omega), b)$$

$$= \sum_{i=1}^{N} \log\left[\frac{1}{2b} \exp\left[-\frac{|y^{(i)} - f(x^{(i)};\omega)|}{b}\right]\right]$$

$$\sum_{i=1}^{N} -\frac{1}{b} |y^{(i)} - f(x^{(i)};\omega)|$$

تابع هزینه قدر مطلق