2022.01.19

The Matrix Calculus You Need For Deep Learning

-Terence Parr and Jeremy Howard

1. Introduction

미적분은 머신러닝, 특히 딥러닝에서 손실 함수를 최적화하는 데 많이 쓰인다.

선형대수에 관한 지식이 머신러닝을 돌리고 이러는 데에는 큰 필요가 없지만 머신러닝, 딥러닝 라이브러리가 어떻게 동작하는지 잘 이해하기 위해선 필수적이다.

예를 들어, 간단한 선형 활성화 함수도 내적을 요구한다.

신경망 구조는 많은 유닛들로 구성되는데 이 유닛들은 layers라고 불리는 유닛의 다수 집합체로 편성된다. 한 층의 활성화는 다른 층의 입력 값이 되고 마지막의 활성화는 최종 output이 될 수 있다.

뉴런을 training한다는 것은 가중치와 절편을 적절히 선택하는 것이다. 손실 함수를 최소화해야 한다. 이를 위해선 확률적 경사하강법이나 Adam 알고리즘을 사용하는데 이 방법 모두 편미분을 요구한다.

2. Review : Scalar derivative rules

Rule	$f(x)$	**Scalar derivative notation with respect to** x	**Example**
Constant	c	0	$\frac{d}{dx}99 = 0$
Multiplication by constant	cf	$c\frac{df}{dx}$	$\frac{d}{dx}3x = 3$
Power Rule	x^n	nx^{n-1}	$\frac{d}{dx}x^3 = 3x^2$
Sum Rule	$f + g$	$\frac{df}{dx} + \frac{dg}{dx}$	$\frac{d}{dx}(x^2 + 3x) = 2x + 3$
Difference Rule	$f - g$	$\frac{df}{dx} - \frac{dg}{dx}$	$\frac{d}{dx}(x^2 - 3x) = 2x - 3$
Product Rule	fg	$f\frac{dg}{dx} + \frac{df}{dx}g$	$\frac{d}{dx}x^2x = x^2 + x2x = 3x^2$
Chain Rule	$f(g(x))$	$\frac{df(u)}{du}\frac{du}{dx}$, let $u = g(x)$	$\frac{d}{dx}ln(x^2) = \frac{1}{x^2}2x = \frac{2}{x}$

3. Introduction to vector calculus and partial derivatives

$$\nabla f(x,y) = [\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y}] = [6yx, 3x^2]$$

4. Matrix calculus

$$J = \begin{bmatrix} \nabla f(x,y) \\ \nabla g(x,y) \end{bmatrix} = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} & \frac{\partial f(x,y)}{\partial y} \\ \frac{\partial g(x,y)}{\partial x} & \frac{\partial g(x,y)}{\partial y} \end{bmatrix} = \begin{bmatrix} 6yx & 3x^2 \\ 2 & 8y^7 \end{bmatrix}$$

4.1 Generalization of the Jacobian

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \nabla f_1(\mathbf{x}) \\ \nabla f_2(\mathbf{x}) \\ \dots \\ \nabla f_m(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial \mathbf{x}} f_1(\mathbf{x}) \\ \frac{\partial}{\partial \mathbf{x}} f_2(\mathbf{x}) \\ \dots \\ \frac{\partial}{\partial \mathbf{x}} f_m(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x_1} f_1(\mathbf{x}) & \frac{\partial}{\partial x_2} f_1(\mathbf{x}) & \dots & \frac{\partial}{\partial x_n} f_1(\mathbf{x}) \\ \frac{\partial}{\partial x_1} f_2(\mathbf{x}) & \frac{\partial}{\partial x_2} f_2(\mathbf{x}) & \dots & \frac{\partial}{\partial x_n} f_2(\mathbf{x}) \\ & & \dots & \\ \frac{\partial}{\partial x_1} f_m(\mathbf{x}) & \frac{\partial}{\partial x_2} f_m(\mathbf{x}) & \dots & \frac{\partial}{\partial x_n} f_m(\mathbf{x}) \end{bmatrix}$$

4.2 Derivatives of vector element-wise binary operators

We can generalize the element-wise binary operations with notation $\mathbf{y} = \mathbf{f}(\mathbf{w}) \bigcirc \mathbf{g}(\mathbf{x})$ where $m = n = |y| = |w| = |x|$. (Reminder: $|x|$ is the number of items in x.) The $\bigcirc$ symbol represents any element-wise operator (such as $+$) and not the $\circ$ function composition operator. Here's what equation $\mathbf{y} = \mathbf{f}(\mathbf{w}) \bigcirc \mathbf{g}(\mathbf{x})$ looks like when we zoom in to examine the scalar equations:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} f_1(\mathbf{w}) \bigcirc g_1(\mathbf{x}) \\ f_2(\mathbf{w}) \bigcirc g_2(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{w}) \bigcirc g_n(\mathbf{x}) \end{bmatrix}$$

where we write n (not m) equations vertically to emphasize the fact that the result of element-wise operators give $m = n$ sized vector results.

Using the ideas from the last section, we can see that the general case for the Jacobian with respect to $\mathbf{w}$ is the square matrix:

$$J_{\mathbf{w}} = \frac{\partial \mathbf{y}}{\partial \mathbf{w}} = \begin{bmatrix} \frac{\partial}{\partial w_1}(f_1(\mathbf{w}) \bigcirc g_1(\mathbf{x})) & \frac{\partial}{\partial w_2}(f_1(\mathbf{w}) \bigcirc g_1(\mathbf{x})) & \dots & \frac{\partial}{\partial w_n}(f_1(\mathbf{w}) \bigcirc g_1(\mathbf{x})) \\ \frac{\partial}{\partial w_1}(f_2(\mathbf{w}) \bigcirc g_2(\mathbf{x})) & \frac{\partial}{\partial w_2}(f_2(\mathbf{w}) \bigcirc g_2(\mathbf{x})) & \dots & \frac{\partial}{\partial w_n}(f_2(\mathbf{w}) \bigcirc g_2(\mathbf{x})) \\ & \dots & & \\ \frac{\partial}{\partial w_1}(f_n(\mathbf{w}) \bigcirc g_n(\mathbf{x})) & \frac{\partial}{\partial w_2}(f_n(\mathbf{w}) \bigcirc g_n(\mathbf{x})) & \dots & \frac{\partial}{\partial w_n}(f_n(\mathbf{w}) \bigcirc g_n(\mathbf{x})) \end{bmatrix}$$

and the Jacobian with respect to $\mathbf{x}$ is:

$$J_{\mathbf{x}} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial}{\partial x_1}(f_1(\mathbf{w}) \bigcirc g_1(\mathbf{x})) & \frac{\partial}{\partial x_2}(f_1(\mathbf{w}) \bigcirc g_1(\mathbf{x})) & \dots & \frac{\partial}{\partial x_n}(f_1(\mathbf{w}) \bigcirc g_1(\mathbf{x})) \\ \frac{\partial}{\partial x_1}(f_2(\mathbf{w}) \bigcirc g_2(\mathbf{x})) & \frac{\partial}{\partial x_2}(f_2(\mathbf{w}) \bigcirc g_2(\mathbf{x})) & \dots & \frac{\partial}{\partial x_n}(f_2(\mathbf{w}) \bigcirc g_2(\mathbf{x})) \\ & \dots & & \\ \frac{\partial}{\partial x_1}(f_n(\mathbf{w}) \bigcirc g_n(\mathbf{x})) & \frac{\partial}{\partial x_2}(f_n(\mathbf{w}) \bigcirc g_n(\mathbf{x})) & \dots & \frac{\partial}{\partial x_n}(f_n(\mathbf{w}) \bigcirc g_n(\mathbf{x})) \end{bmatrix}$$

4.3 Derivatives involving scalar expansion

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = diag\left(\ldots \frac{\partial}{\partial x_i}(f_i(x_i) \bigcirc g_i(z)) \ldots\right)$$

4.4 Vector Sum Reduction

Let $y = sum(\mathbf{f}(\mathbf{x})) = \sum_{i=1}^{n} f_i(\mathbf{x})$. Notice we were careful here to leave the parameter as a vector $\mathbf{x}$ because each function f_i could use all values in the vector, not just x_i. The sum is over the **results** of the function and not the parameter. The gradient ($1 \times n$ Jacobian) of vector summation is:

$$\begin{aligned}
\frac{\partial y}{\partial \mathbf{x}} &= \left[\frac{\partial y}{\partial x_1}, \frac{\partial y}{\partial x_2}, \ldots, \frac{\partial y}{\partial x_n}\right] \\
&= \left[\frac{\partial}{\partial x_1}\sum_i f_i(\mathbf{x}),\ \frac{\partial}{\partial x_2}\sum_i f_i(\mathbf{x}),\ \ldots,\ \frac{\partial}{\partial x_n}\sum_i f_i(\mathbf{x})\right] \\
&= \left[\sum_i \frac{\partial f_i(\mathbf{x})}{\partial x_1},\ \sum_i \frac{\partial f_i(\mathbf{x})}{\partial x_2},\ \ldots,\ \sum_i \frac{\partial f_i(\mathbf{x})}{\partial x_n}\right] \qquad \text{(move derivative inside } \textstyle\sum\text{)}
\end{aligned}$$

4.5 The Chain Rules

복잡한 식을 미분식이 계산하기 편하도록 부분 표현들로 나누는 전략이다.

Forward differentiation from x to y	**Backward differentiation from y to x**
$\frac{dy}{dx} = \frac{du}{dx}\frac{dy}{du}$	$\frac{dy}{dx} = \frac{dy}{du}\frac{du}{dx}$

5. The gradient of neuron activation

6. The gradient of the neural network loss function.

7. Summary