Mathematics for AI

Adams Wai Kin Kong
School of Computer Science and Engineering
Nanyang Technological University, Singapore
adamskong@ntu.edu.sg

The Matrix Calculus You Need For Deep Leaning, Terence Parr and Jeremy Howard, July 2018
https://arxiv.org/pdf/1802.01528.pdf

Matrix Differentiation, Randal J Barnes,

https://atmos.washington.edu/~dennis/MatrixCalculus.pdf

Matrix Calculus

In the previous section, we can consider only functions with one-dimensional output, such as $y = f(x_1, x_2, \cdots, x_n)$ or $y = f(\boldsymbol{x})$, where $\boldsymbol{x}$ is a vector and x_i is the element of $\boldsymbol{x}$.

Let us consider that we have m functions depending on $\boldsymbol{x}$.

$y_1 = f_1(x_1, x_2, \cdots, x_n)$

$y_2 = f_2(x_1, x_2, \cdots, x_n)$

………….

$y_m = f_m(x_1, x_2, \cdots, x_n)$

They can be represented as $\boldsymbol{y} = F(\boldsymbol{x})$, where both $\boldsymbol{y}$ and $\boldsymbol{x}$ are vectors. In the lecture, you will learn how to differentiate F with respects to $\boldsymbol{x}$.

Jacobian

Let us consider an example $f(x, y) = 3x^2y$ and $\mathrm{g}(x, y) = 2x + 4y^2$. Their gradients are

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

$$\nabla g(x, y) = \left[\frac{\partial g(x, y)}{\partial x}, \frac{\partial g(x, y)}{\partial y}\right] = [2, 8y]$$

By organizing in a matrix form, we have the Jacobin matrix (or just the Jacobian)

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

Jacobian

Note that there are multiple ways to represent the Jacobian. In this note, the so-called numerator layout is used but many papers and software will use the denominator layout. It is just the transpose of the numerator layout Jacobian.

$$J=\begin{bmatrix} 6xy & 2 \\ 3x^2 & 8y \end{bmatrix}.$$

Jacobian

Let $\boldsymbol{x} = [x_1 \cdots x_n]^T$ be a n dimensional column vector and $\boldsymbol{y} = [y_1 \cdots y_m]^T$ be a m dimensional column vector. Given $y_i = f_i(\boldsymbol{x}), \forall i = 1, \dots, m$,. More clearly,

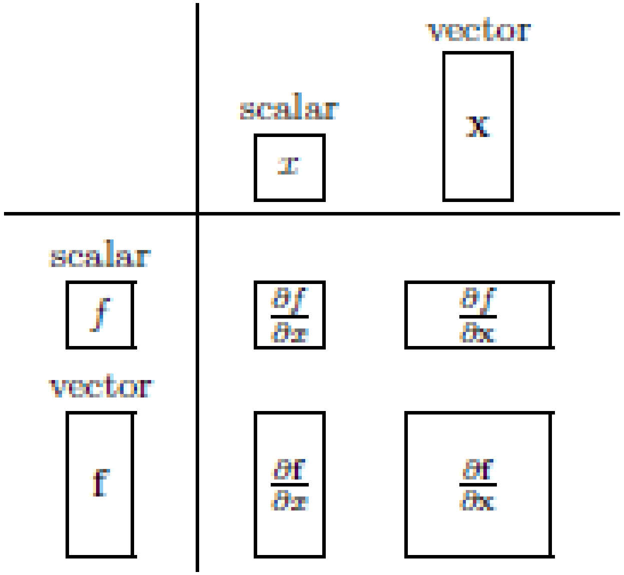
$$y_1 = f_1(\boldsymbol{x})$$
$$y_2 = f_2(\boldsymbol{x})$$
$$\cdots$$
$$y_m = f_m(\boldsymbol{x})$$

Representing these equations as $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$, where $\boldsymbol{f}: \Re^n \to \Re^m$.

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

Jacobian

Note that each $\frac{\partial}{\partial \boldsymbol{x}} f_i(\boldsymbol{x})$ is a row n-vector and n is the length of the vector. The Jacobian is a $m \times n$ matrix (row $\times$ column).



Example

Compute the Jacobian of identity function $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{x}$, with $x_i = f_i(\boldsymbol{x})$, where $i = 1 \cdots m$ and $\boldsymbol{x}$ is a m-dimensional vector.

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} = \begin{bmatrix} \nabla f_1(\boldsymbol{x}) \\ \nabla f_2(\boldsymbol{x}) \\ \vdots \\ \nabla f_m(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial \boldsymbol{x}} f_1(\boldsymbol{x}) \\ \frac{\partial}{\partial \boldsymbol{x}} f_2(\boldsymbol{x}) \\ \vdots \\ \frac{\partial}{\partial \boldsymbol{x}} f_m(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x_1} x_1 & \frac{\partial}{\partial x_2} x_1 & \cdots & \frac{\partial}{\partial x_m} x_1 \\ \frac{\partial}{\partial x_1} x_2 & \frac{\partial}{\partial x_2} x_2 & \cdots & \frac{\partial}{\partial x_m} x_2 \\ \vdots & \vdots & & \vdots \\ \frac{\partial}{\partial x_1} x_m & \frac{\partial}{\partial x_2} x_m & \cdots & \frac{\partial}{\partial x_m} x_m \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} = \boldsymbol{I}$$

where $\boldsymbol{I}$ is the identity matrix. Note that $\frac{\partial}{\partial x_i} x_j = 0$ if $j \neq i$.

Derivatives of vector element-wise binary operators

Element-wise binary operations on vectors, such as vector addition w + x, are important for deep learning training. The term, element-wise binary operations means applying an operator to the first item of each vector to get the first item of the output, then to the second items of the inputs for the second item of the output, and so forth. For examples

$$\boldsymbol{x} + \boldsymbol{y} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} + \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} x_1 + y_1 \\ \vdots \\ x_m + y_m \end{bmatrix}$$

$max(\boldsymbol{x}, \mathbf{0}) = \begin{bmatrix} max(x_1, 0) \\ \vdots \\ max(x_m, 0) \end{bmatrix}$, where $\mathbf{0}$ is a zero vector.

Derivatives of vector element-wise binary operators

In this note, the element-wise binary operator is denoted as $\odot$ and $\boldsymbol{y} = \odot(\boldsymbol{w}, \boldsymbol{x})$, where $\boldsymbol{y}, \boldsymbol{w}$ and $\boldsymbol{x}$ are column vectors with the same dimension, n. Using a matrix from to display the equation, we have

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \odot_1(\boldsymbol{w}, \boldsymbol{x}) \\ \odot_2(\boldsymbol{w}, \boldsymbol{x}) \\ \vdots \\ \odot_n(\boldsymbol{w}, \boldsymbol{x}) \end{bmatrix}$$

Derivatives of vector element-wise binary operators

The Jacobian with respect to **w** is the square matrix:

$$\boldsymbol{J_w} = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{w}} = \begin{bmatrix} \frac{\partial}{\partial w_1} \odot_1(\boldsymbol{w}, \boldsymbol{x}) & \frac{\partial}{\partial w_2} \odot_1(\boldsymbol{w}, \boldsymbol{x}) & \cdots & \frac{\partial}{\partial w_n} \odot_1(\boldsymbol{w}, \boldsymbol{x}) \\ \frac{\partial}{\partial w_1} \odot_2(\boldsymbol{w}, \boldsymbol{x}) & \frac{\partial}{\partial w_2} \odot_2(\boldsymbol{w}, \boldsymbol{x}) & \cdots & \frac{\partial}{\partial w_n} \odot_2(\boldsymbol{w}, \boldsymbol{x}) \\ \vdots & \vdots & & \vdots \\ \frac{\partial}{\partial w_1} \odot_n(\boldsymbol{w}, \boldsymbol{x}) & \frac{\partial}{\partial w_2} \odot_n(\boldsymbol{w}, \boldsymbol{x}) & \cdots & \frac{\partial}{\partial w_n} \odot_n(\boldsymbol{w}, \boldsymbol{x}) \end{bmatrix}$$

The Jacobian with respect to **x** is the square matrix:

$$\boldsymbol{J_x} = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial}{\partial x_1} \odot_1(\boldsymbol{w}, \boldsymbol{x}) & \frac{\partial}{\partial x_2} \odot_1(\boldsymbol{w}, \boldsymbol{x}) & \cdots & \frac{\partial}{\partial x_n} \odot_1(\boldsymbol{w}, \boldsymbol{x}) \\ \frac{\partial}{\partial x_1} \odot_2(\boldsymbol{w}, \boldsymbol{x}) & \frac{\partial}{\partial x_2} \odot_2(\boldsymbol{w}, \boldsymbol{x}) & \cdots & \frac{\partial}{\partial x_n} \odot_2(\boldsymbol{w}, \boldsymbol{x}) \\ \vdots & \vdots & & \vdots \\ \frac{\partial}{\partial x_1} \odot_n(\boldsymbol{w}, \boldsymbol{x}) & \frac{\partial}{\partial x_2} \odot_n(\boldsymbol{w}, \boldsymbol{x}) & \cdots & \frac{\partial}{\partial x_n} \odot_n(\boldsymbol{w}, \boldsymbol{x}) \end{bmatrix}$$

Derivatives of vector element-wise binary operators

Element-wise operations imply that y_i only depends on x_i and w_i. In other words, $y_i = \odot_i(w_i, x_i)$. Here I reuse the same notations for inputs with different dimensions, i.e., $\odot_i(w_i, x_i)$ and $\odot_i(\boldsymbol{w}, \boldsymbol{x})$. Therefore, if $i \neq j$

$$\frac{\partial}{\partial w_j} \odot_i(\boldsymbol{w}, \boldsymbol{x}) = \frac{\partial}{\partial w_j} \odot_i(w_i, x_i) = 0$$

and

$$\frac{\partial}{\partial x_j} \odot_i(\boldsymbol{w}, \boldsymbol{x}) = \frac{\partial}{\partial x_j} \odot_i(w_i, x_i) = 0$$

Derivatives of vector element-wise binary operators

The Jacobian matrixes become diagonal matrixes:

$$J_w = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{w}} = \begin{bmatrix} \frac{\partial}{\partial w_1} \bigodot_1(\boldsymbol{w}, \boldsymbol{x}) & 0 & \cdots & 0 \\ 0 & \frac{\partial}{\partial w_2} \bigodot_2(\boldsymbol{w}, \boldsymbol{x}) & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \frac{\partial}{\partial w_n} \bigodot_n(\boldsymbol{w}, \boldsymbol{x}) \end{bmatrix}$$

$$J_x = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial}{\partial x_1} \bigodot_1(\boldsymbol{w}, \boldsymbol{x}) & 0 & \cdots & 0 \\ 0 & \frac{\partial}{\partial x_2} \bigodot_2(\boldsymbol{w}, \boldsymbol{x}) & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \frac{\partial}{\partial x_n} \bigodot_n(\boldsymbol{w}, \boldsymbol{x}) \end{bmatrix}$$

Derivatives of vector element-wise binary operators

More succinctly, we can write

$$J_{\boldsymbol{w}} = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{w}} = diag\left(\frac{\partial}{\partial w_1} \odot_1 (w_1, x_1), \cdots \frac{\partial}{\partial w_n} \odot_n (w_n, x_n)\right)$$

$$J_{\boldsymbol{x}} = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} = diag\left(\frac{\partial}{\partial x_1} \odot_1 (w_1, x_1), \cdots \frac{\partial}{\partial x_n} \odot_n (w_n, x_n)\right)$$

Examples

Operators	Partial with respect to $\boldsymbol{w}$
+	$\frac{\partial(\boldsymbol{w}+\boldsymbol{x})}{\partial \boldsymbol{w}} = \operatorname{diag}\left(\cdots \frac{\partial(w_i + x_i)}{\partial w_i} \cdots\right) = \operatorname{diag}(\vec{1}) = I$
-	$\frac{\partial(\boldsymbol{w}-\boldsymbol{x})}{\partial \boldsymbol{w}} = \operatorname{diag}\left(\cdots \frac{\partial(w_i - x_i)}{\partial w_i} \cdots\right) = \operatorname{diag}(\vec{1}) = I$
$\otimes$	$\frac{\partial(\boldsymbol{w}\otimes \boldsymbol{x})}{\partial \boldsymbol{w}} = \operatorname{diag}\left(\cdots \frac{\partial(w_i \times x_i)}{\partial w_i} \cdots\right) = \operatorname{diag}(\boldsymbol{x})$
$\oslash$	$\frac{\partial(\boldsymbol{w}\oslash \boldsymbol{x})}{\partial \boldsymbol{w}} = \operatorname{diag}\left(\cdots \frac{\partial(w_i / x_i)}{\partial w_i} \cdots\right) = \operatorname{diag}(\cdots 1/x_i \cdots)$

Examples

Operators	Partial with respect to $\boldsymbol{x}$
+	$\frac{\partial(\boldsymbol{w}+\boldsymbol{x})}{\partial\boldsymbol{x}} = \mathrm{diag}\left(\cdots\frac{\partial(w_i+x_i)}{\partial x_i}\cdots\right) = \mathrm{diag}(\vec{1}) = I$
-	$\frac{\partial(\boldsymbol{w}-\boldsymbol{x})}{\partial\boldsymbol{x}} = \mathrm{diag}\left(\cdots\frac{\partial(w_i-x_i)}{\partial x_i}\cdots\right) = \mathrm{diag}(-\vec{1}) = -I$
$\otimes$	$\frac{\partial(\boldsymbol{w}\otimes\boldsymbol{x})}{\partial\boldsymbol{x}} = \mathrm{diag}\left(\cdots\frac{\partial(w_i\times x_i)}{\partial x_i}\cdots\right) = \mathrm{diag}(\boldsymbol{w})$
$\oslash$	$\frac{\partial(\boldsymbol{w}\oslash\boldsymbol{x})}{\partial\boldsymbol{x}} = \mathrm{diag}\left(\cdots\frac{\partial(w_i/x_i)}{\partial x_i}\cdots\right) = \mathrm{diag}\left(\cdots-\frac{w_i}{x_i^2}\cdots\right)$

Derivatives involving scalar expansion

When we multiple or add scalars to vector, we in fact expand the scalar to a vector and perform an element-wise operation. For example, $\boldsymbol{y} = \boldsymbol{x} + z$ means $\boldsymbol{y} = \boldsymbol{x} + z\vec{\mathbf{1}}$, where z is a scalar and $\boldsymbol{y}$ and $\boldsymbol{x}$ are vectors.

If $\boldsymbol{y} = \odot(\boldsymbol{x}, z\vec{\mathbf{1}})$, the partial derivatives with respect to $\boldsymbol{x}$ is

$$\boldsymbol{J_x} = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} = diag\left(\cdots \frac{\partial}{\partial x_i} \odot_i (x_i, z), \cdots\right)$$

and the partial derivatives with respect to z is

$$\frac{\partial \boldsymbol{y}}{\partial z} = \frac{\partial}{\partial z} \odot(\boldsymbol{x}, z)$$

Example

Let us consider the case, $\boldsymbol{y} = \boldsymbol{x} + z$. Therefore $\frac{\partial}{\partial x_i}(x_i + z) = 1$ and $\frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{x} + z) = \boldsymbol{I}$.

Computing the partial derivative with respect to the scalar parameter z, the result is a column vector, not a matrix.

$\frac{\partial}{\partial z}(x_i + z) = 1$ and $\frac{\partial}{\partial z}(\boldsymbol{x} + z) = \vec{\mathbf{1}}$

Example

Let us consider the case, $\boldsymbol{y} = z\boldsymbol{x}$, where **z** is a scalar and **y** and ***x*** are vectors. Since $\frac{\partial}{\partial x_i} z x_i = z$,

$$\frac{\partial}{\partial \boldsymbol{x}} z\boldsymbol{x} = \mathrm{diag}\left(\mathrm{z}\vec{1}\right) = z\boldsymbol{I}$$

Since $\frac{\partial}{\partial z} z x_i = x_i$, $\frac{\partial}{\partial \boldsymbol{z}} z\boldsymbol{x} = \boldsymbol{x}$

The Chain Rules

You have learned the single-variable chain rule,

$$\frac{dy}{dx} = \frac{dy}{du}\frac{du}{dx}.$$

Forward differentiation is from *x* to *y*. It means that we first compute $\frac{du}{dx}$ and then $\frac{dy}{du}$. Then, we multiple them together. Backward differentiation is from *y* to *x*. It means that we first compute $\frac{dy}{du}$ and then $\frac{du}{dx}$. Then, we multiple them together. Mathematically, they are the same, but in deep learning training, they are different. Once is from the top of network and the other is from the bottom of the network.

Example

Compute $\frac{dy}{dx}$ if $y = f(x) = \ln(\sin(x^3)^2)$

Intermediate variables	Derivatives
$u_1 = f_1(x) = x^3$	$\frac{d}{dx}u_1 = 3x^2$
$u_2 = f_2(u_1) = \sin(u_1)$	$\frac{d}{du_1}u_2 = \cos(u_1)$
$u_3 = f_3(u_2) = u_2^2$	$\frac{d}{du_2}u_3 = 2u_2$
$y = u_4 = f_4(u_3) = \ln(u_3)$	$\frac{d}{du_3}u_4 = \frac{1}{u_3}$

Apply $\frac{dy}{dx} = \frac{du_4}{du_3}\frac{du_3}{du_2}\frac{du_2}{du_1}\frac{du_1}{dx} = \frac{6u_2x^2\cos(u_1)}{u_3}$

Substitute $\frac{dy}{dx} = \frac{6\sin(u_1)x^2\cos(x^3)}{u_2^2} = \frac{6\sin(x^3)x^2\cos(x^3)}{\sin(u_1)^2} = \frac{6x^2\cos(x^3)}{\sin(x^3)}$

Single-variable total-derivative chain rule

4 The Chain Rule (General Version) Suppose that u is a differentiable function of the n variables $x_1, x_2, \ldots, x_n$ and each x_j is a differentiable function of the m variables $t_1, t_2, \ldots, t_m$. Then u is a function of $t_1, t_2, \ldots, t_m$ and

$$\frac{\partial u}{\partial t_i} = \frac{\partial u}{\partial x_1}\frac{\partial x_1}{\partial t_i} + \frac{\partial u}{\partial x_2}\frac{\partial x_2}{\partial t_i} + \cdots + \frac{\partial u}{\partial x_n}\frac{\partial x_n}{\partial t_i}$$

for each $i = 1, 2, \ldots, m$.

- Note that u is a function with one dimensional output.

Vector chain rule

Vector chain rule is used to compute the derivative of vector function, $\boldsymbol{y} = f(\boldsymbol{x})$, where both $\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors. In the following example, x is considered as 1 by 1 vector. For example,

$$\begin{bmatrix} y_1(x) \\ y_2(x) \end{bmatrix} = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} = \begin{bmatrix} \ln(x^2) \\ \sin(3x) \end{bmatrix}$$

Let us introduce two intermediate variables $g_1(x) = x^2$ and $g_2(x) = 3x$ and represent $\boldsymbol{y} = f\big(\boldsymbol{g}(x)\big)$. More clearly,

$$\begin{bmatrix} g_1(x) \\ g_2(x) \end{bmatrix} = \begin{bmatrix} x^2 \\ 3x \end{bmatrix} \text{ and } \begin{bmatrix} f_1(\boldsymbol{g}) \\ f_2(\boldsymbol{g}) \end{bmatrix} = \begin{bmatrix} \ln(g_1) \\ \sin(g_2) \end{bmatrix}$$

Vector chain rule

$$\frac{\partial \boldsymbol{y}}{\partial x} = \begin{bmatrix} \frac{\partial f_1(\boldsymbol{g})}{\partial x} \\ \frac{\partial f_2(\boldsymbol{g})}{\partial x} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial g_1}\frac{\partial g_1}{\partial x} + \frac{\partial f_1}{\partial g_2}\frac{\partial g_2}{\partial x} \\ \frac{\partial f_2}{\partial g_1}\frac{\partial g_1}{\partial x} + \frac{\partial f_2}{\partial g_2}\frac{\partial g_2}{\partial x} \end{bmatrix} = \begin{bmatrix} \frac{2x}{g_1} + 0 \\ 0 + \cos(g_2)\,3 \end{bmatrix} = \begin{bmatrix} \frac{2}{x} \\ 3\cos(3x) \end{bmatrix}$$

Reordering the abstract form in red, we can obtain

$$\begin{bmatrix} \frac{\partial f_1}{\partial g_1}\frac{\partial g_1}{\partial x} + \frac{\partial f_1}{\partial g_2}\frac{\partial g_2}{\partial x} \\ \frac{\partial f_2}{\partial g_1}\frac{\partial g_1}{\partial x} + \frac{\partial f_2}{\partial g_2}\frac{\partial g_2}{\partial x} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial g_1} & \frac{\partial f_1}{\partial g_2} \\ \frac{\partial f_2}{\partial g_1} & \frac{\partial f_2}{\partial g_2} \end{bmatrix} \begin{bmatrix} \frac{\partial g_1}{\partial x} \\ \frac{\partial g_2}{\partial x} \end{bmatrix} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{g}}\frac{\partial \boldsymbol{g}}{\partial x}$$

That means that the Jacobian is the multiplication of two other Jacobians.

Vector chain rule

Let us check the results in the example

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{g}}\frac{\partial \boldsymbol{g}}{\partial x} = \begin{bmatrix} \frac{1}{g_1} & 0 \\ 0 & \cos(g_2) \end{bmatrix} \begin{bmatrix} 2x \\ 3 \end{bmatrix} = \begin{bmatrix} \frac{1}{g_1} 2x + 0 \\ 0 + \cos(g_2)\, 3 \end{bmatrix} = \begin{bmatrix} \frac{2}{x} \\ 3\cos(3x) \end{bmatrix}$$

The vector chain rule and the single-variable chain rule have the same form.

Vector chain rule	**Single-variable chain rule**
$\frac{\partial}{\partial x}\boldsymbol{f}(\boldsymbol{g}(x)) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{g}}\frac{\partial \boldsymbol{g}}{\partial x}$	$\frac{d}{dx}f(g(x)) = \frac{df}{dg}\frac{dg}{dx}$

Vector chain rule

In the previous slides, *x* is a scalar. To make this formula work for multiple parameters or vector *x*, we just have to change x to vector. Thus, both $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}$ and $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}$ are matrixes and the complete vector chain rule is

$$\frac{\partial}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{g}(\boldsymbol{x})) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{g}} \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}$$

(Note: matrix multiply doesn't commute. $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{g}} \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} \neq \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{g}}$)

The vector formula automatically takes into consideration the total derivative while maintaining the same notational simplicity.

Vector chain rule

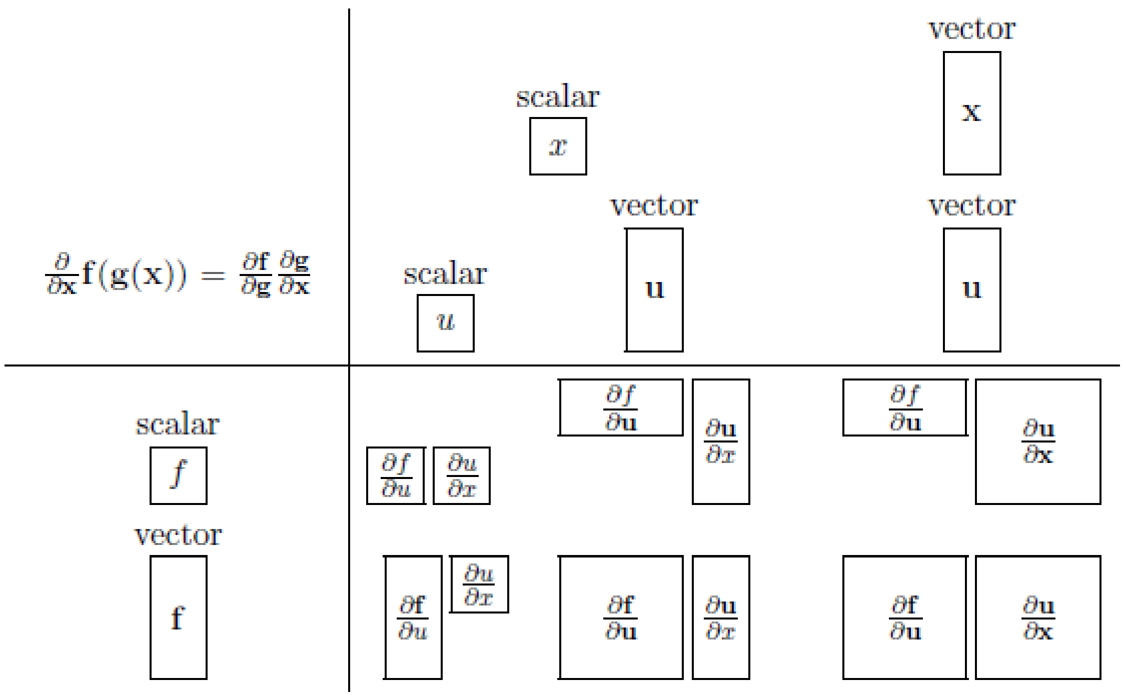
More clearly,

$$\frac{\partial}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{g}(\boldsymbol{x})) = \begin{bmatrix} \frac{\partial f_1}{\partial g_1} & \frac{\partial f_1}{\partial g_2} & \cdots & \frac{\partial f_1}{\partial g_k} \\ \frac{\partial f_2}{\partial g_1} & \frac{\partial f_2}{\partial g_2} & \cdots & \frac{\partial f_2}{\partial g_k} \\ & & \cdots & \\ \frac{\partial f_m}{\partial g_1} & \frac{\partial f_m}{\partial g_2} & \cdots & \frac{\partial f_m}{\partial g_k} \end{bmatrix} \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_n} \\ & & \cdots & \\ \frac{\partial g_k}{\partial x_1} & \frac{\partial g_k}{\partial x_2} & \cdots & \frac{\partial g_k}{\partial x_n} \end{bmatrix}$$

where m, n and k are the length of $\boldsymbol{f}, \boldsymbol{x}\ and\ \boldsymbol{g}$. The resulting Jacobian is $m \times n$. (an $m \times k\ matrix\ multiplied\ by$ a $k \times n$ matrix.)

Vector chain rule

The figure below summarizes the shapes of the Jacobian, where $\boldsymbol{u} = g(\boldsymbol{x})$.

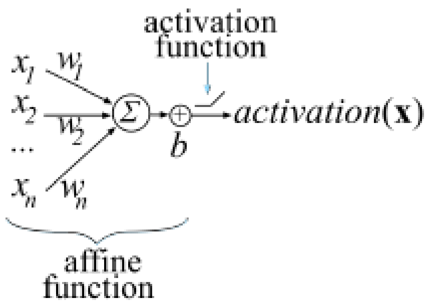


The gradient of neuron activation

Now, we consider a typical neuron activation for a signal neural network computation unit with respect to the model parameters, *w* and *b*:

$$activation(x) = \max(0, \boldsymbol{w} \cdot \boldsymbol{x} + b)$$

where **x** and **w** are vectors and 0 and b are scalar. In the neural network training, we need to compute $\frac{\partial}{\partial w} \max(0, \boldsymbol{w} \cdot \boldsymbol{x} + b)$ and $\frac{\partial}{\partial b} \max(0, \boldsymbol{w} \cdot \boldsymbol{x} + b)$.



The gradient of neuron activation

Let us consider $\frac{\partial}{\partial \boldsymbol{w}} \boldsymbol{w} \cdot \boldsymbol{x} + b$ and $\frac{\partial}{\partial b} \boldsymbol{w} \cdot \boldsymbol{x} + b$ first. Clearly

$$\frac{\partial}{\partial b} \boldsymbol{w} \cdot \boldsymbol{x} + b = 1$$

Since $\frac{\partial}{\partial w_i} \boldsymbol{w} \cdot \boldsymbol{x} + b = x_i$,

$$\frac{\partial}{\partial \boldsymbol{w}} \boldsymbol{w} \cdot \boldsymbol{x} + b = \boldsymbol{x}^T.$$

Now we consider $\max(0, z)$, where $z = \boldsymbol{w} \cdot \boldsymbol{x} + b$.

$$\frac{\partial}{\partial z} \max(0, z) = \begin{cases} 0 & z \leq 0 \\ \frac{dz}{dz} = 1 & z > 0 \end{cases}$$

When z=0, $\max(0, z)$ is mathematically non-differentiable but we define it as zero in network training.

The gradient of neuron activation

Using the vector chain rule

$$\frac{\partial \max(0, \boldsymbol{w} \cdot \boldsymbol{x} + b)}{\partial \boldsymbol{w}} = \frac{\partial \max(0, \boldsymbol{w} \cdot \boldsymbol{x} + b)}{\partial z} \frac{\partial z}{\partial \boldsymbol{w}}$$

$$= \begin{cases} 0 \dfrac{\partial z}{\partial \boldsymbol{w}} = \vec{\mathbf{0}}^T & z \leq 0 \\ 1 \dfrac{\partial z}{\partial \mathbf{w}} = \boldsymbol{x}^T & z > 0 \end{cases}$$

Substitute $\mathrm{z} = \boldsymbol{w} \cdot \boldsymbol{x} + b$ into the equation above.

$$\frac{\partial \max(0, \boldsymbol{w} \cdot \boldsymbol{x} + b)}{\partial \boldsymbol{w}} = \begin{cases} \vec{\mathbf{0}}^T & \boldsymbol{w} \cdot \boldsymbol{x} + b \leq 0 \\ \boldsymbol{x}^T & \boldsymbol{w} \cdot \boldsymbol{x} + b > 0 \end{cases}$$

The gradient of neuron activation

Now we consider the derivative of the neuron activation with respect to *b*.

$$\frac{\partial \max(0, \boldsymbol{w} \cdot \boldsymbol{x} + b)}{\partial b} = \begin{cases} 0 \dfrac{\partial z}{\partial b} = 0 & \boldsymbol{w} \cdot \boldsymbol{x} + b \leq 0 \\ 1 \dfrac{\partial z}{\partial b} = 1 & \boldsymbol{w} \cdot \boldsymbol{x} + b > 0 \end{cases}$$

The gradient of the neural network loss function

Training a neuron requires that we take the derivative of our loss or cost function with respect to the parameters of our model, w and b. We consider a simple L2 norm as a cost function and we have N training vectors, $\boldsymbol{x_1}, \boldsymbol{x_2} \cdots, \boldsymbol{x_N}$.

Let $\hat{y}_i = \max(0, \boldsymbol{w} \cdot \boldsymbol{x_i} + b)$ and the cost function

$$C(\boldsymbol{w}, b, \boldsymbol{X}, \boldsymbol{y}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \max(0, \boldsymbol{w} \cdot \boldsymbol{x_i} + b))^2$$

where y_i is the target output of the training vector $\boldsymbol{x_i}$, $\boldsymbol{y} = [y_1 \cdots y_N]^{\boldsymbol{T}}$ and $\boldsymbol{X} = [\boldsymbol{x_1}, \boldsymbol{x_2} \cdots, \boldsymbol{x_N}]$.

The gradient of the neural network loss function

To use the chain rules, the following intermediate variables are introduced.

$$u(\boldsymbol{w}, b, \boldsymbol{x_i}) = \max(0, \boldsymbol{w} \cdot \boldsymbol{x_i} + b)$$

$$v(y, u) = \mathrm{y} - \mathrm{u}$$

$$\mathrm{C}(v) = \frac{1}{N} \sum_{i=1}^{N} v^2$$

Note that u and v function of $\boldsymbol{x_i}$.

Using the previous results, we have

$$\frac{\partial u(\boldsymbol{w}, b, \boldsymbol{x})}{\partial \boldsymbol{w}} = \begin{cases} \vec{\boldsymbol{0}}^T & \boldsymbol{w} \cdot \boldsymbol{x} + b \leq 0 \\ \boldsymbol{x}^T & \boldsymbol{w} \cdot \boldsymbol{x} + b > 0 \end{cases}$$

And

$$\frac{\partial v(y, u)}{\partial \boldsymbol{w}} = \vec{\boldsymbol{0}}^T - \frac{\partial u}{\partial \boldsymbol{w}} = = \begin{cases} \vec{\boldsymbol{0}}^T & \boldsymbol{w} \cdot \boldsymbol{x} + b \leq 0 \\ -\boldsymbol{x}^T & \boldsymbol{w} \cdot \boldsymbol{x} + b > 0 \end{cases}$$

The gradient of the neural network loss function

$$\frac{\partial C(v)}{\partial \boldsymbol{w}} = \frac{\partial}{\partial \boldsymbol{w}} \frac{1}{N} \sum_{i=1}^{N} v^2 = \frac{1}{N} \sum_{i=1}^{N} 2v \frac{\partial v}{\partial \boldsymbol{w}} = \frac{1}{N} \sum_{i=1}^{N} 2v \frac{\partial (\mathrm{y}-\mathrm{u})}{\partial \boldsymbol{w}} = \frac{1}{N} \sum_{i=1}^{N} -2v \frac{\partial \mathrm{u}}{\partial \boldsymbol{w}}$$

Using the previous results

$$= \frac{1}{N} \sum_{i=1}^{N} \begin{cases} -2v\vec{\mathbf{0}}^T & \boldsymbol{w} \cdot \boldsymbol{x_i} + b \le 0 \\ -2v\boldsymbol{x_i^T} & \boldsymbol{w} \cdot \boldsymbol{x_i} + b > 0 \end{cases}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \begin{cases} \vec{\mathbf{0}}^T & \boldsymbol{w} \cdot \boldsymbol{x_i} + b \le 0 \\ -2(y_i - \max(0, \boldsymbol{w} \cdot \boldsymbol{x} + b))\boldsymbol{x_i^T} & \boldsymbol{w} \cdot \boldsymbol{x_i} + b > 0 \end{cases}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \begin{cases} \vec{\mathbf{0}}^T & \boldsymbol{w} \cdot \boldsymbol{x_i} + b \le 0 \\ -2(y_i - (\boldsymbol{w} \cdot \boldsymbol{x} + b))\boldsymbol{x_i^T} & \boldsymbol{w} \cdot \boldsymbol{x_i} + b > 0 \end{cases}$$

The gradient of the neural network loss function

$$\frac{\partial C(v)}{\partial b} = \frac{\partial}{\partial b}\frac{1}{N}\sum_{i=1}^{N} v^2 = \frac{1}{N}\sum_{i=1}^{N} 2v\frac{\partial v}{\partial b} = \frac{1}{N}\sum_{i=1}^{N} 2v\frac{\partial (y-u)}{\partial b} = \frac{1}{N}\sum_{i=1}^{N} -2v\frac{\partial u}{\partial b}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\begin{cases} -2v \times 0 & \boldsymbol{w}\cdot\boldsymbol{x_i} + b \le 0 \\ -2v \times 1 & \boldsymbol{w}\cdot\boldsymbol{x_i} + b > 0 \end{cases}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\begin{cases} 0 & \boldsymbol{w}\cdot\boldsymbol{x_i} + b \le 0 \\ 2 \times (\boldsymbol{w}\cdot\boldsymbol{x_i} + b - y_i) & \boldsymbol{w}\cdot\boldsymbol{x_i} + b > 0 \end{cases}$$

Some matrix differentiation formulas

Before providing some formulas for matrix differentiation, some matrix and vector multiplication formulas are given.

Let $a_{ij} \in \Re, \mathrm{i} = 1, 2 \cdots m, j = 1,2 \cdots n$ and

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

where is a $m \times n$ matrix. We also use

$$\boldsymbol{A} = [a_{ij}], \mathrm{i} = 1,2, \cdots m; j = 1,2 \cdots, n$$

to represent the matrix.

Some matrix differentiation formulas

Let A be $m \times n$, and B be $n \times p$, and let the product AB be

$$\boldsymbol{C} = \boldsymbol{AB}$$

C is an $m \times p$ matrix and its element at (i, j) is

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

for all $i = 1,2, \cdots m$ and $j = 1,2 \cdots , p$.

Some matrix differentiation formulas

Similarly, let $\boldsymbol{x} = [x_1, x_2 \cdots x_n]^T$ and $\mathbf{y} = [y_1, y_2 \cdots y_m]^T$. Then, the elements of $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{x}$ are

$$z_i = \sum_{k=1}^{n} a_{ik} x_k$$

and the elements of $\boldsymbol{w}^T = \boldsymbol{y}^T \boldsymbol{A}$ are

$$w_i = \sum_{k=1}^{m} a_{ki} y_k$$

Furthermore, the scalar resulting from the product $\alpha = \boldsymbol{y}^T \boldsymbol{A}\boldsymbol{x}$ can be computed by

$$\alpha = \sum_{j=1}^{m} \sum_{k=1}^{n} a_{jk} y_j x_k$$

Some matrix differentiation formulas

Given $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{x}$, then $\frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{z} = \boldsymbol{A}$

Proof: Using $z_i = \sum_{k=1}^{n} a_{ik} x_k$,

$$\frac{\partial}{\partial x_j} z_i = \sum_{k=1}^{n} a_{ik} \frac{\partial}{\partial x_j} x_k = a_{ij}$$

Thus, $\frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{z} = \boldsymbol{A}.$

Some matrix differentiation formulas

If $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{y}$ and $\boldsymbol{y}$ is a function of $\boldsymbol{x}$, then $\frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{z} = \boldsymbol{A}\,\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}$

Proof: Using $z_i = \sum_{k=1}^{n} a_{ik} y_k$

$$\frac{\partial}{\partial x_j} z_i = \sum_{k=1}^{n} a_{ik} \frac{\partial}{\partial x_j} y_k$$

The right hand side of the above equation is element (i, j) of $\boldsymbol{A}$ $\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}$.

Some matrix differentiation formulas

If $\alpha = \boldsymbol{y}^T \boldsymbol{A}\boldsymbol{x}$, where $\boldsymbol{y}$ is $m \times 1$, $\boldsymbol{x}$ is n$\times$ 1, $\boldsymbol{A}$ is $m \times n$, $\boldsymbol{A}$ is independent of $\boldsymbol{x}$ and $\boldsymbol{y}$ and $\boldsymbol{y}$ is independent of $\boldsymbol{x}$, then

$$\frac{\partial \alpha}{\partial \boldsymbol{x}} = \boldsymbol{y}^T \boldsymbol{A} \text{ and } \frac{\partial \alpha}{\partial \boldsymbol{y}} = \boldsymbol{x}^T \boldsymbol{A}^T$$

Some matrix differentiation formulas

Proof $\frac{\partial\alpha}{\partial\boldsymbol{x}} = \boldsymbol{y}^T\boldsymbol{A}$. Let $\boldsymbol{w}^T = \boldsymbol{y}^T\boldsymbol{A}$. Then, $\alpha = \boldsymbol{y}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{w}^T\boldsymbol{x}$.

$$\frac{\partial\alpha}{\partial x_i} = \sum_{j=1}^{n} w_j \frac{\partial x_j}{\partial x_i} = w_i$$

Thus, $\frac{\partial\alpha}{\partial\boldsymbol{x}} = \boldsymbol{w}^T = \boldsymbol{y}^T\boldsymbol{A}$.

Proof $\frac{\partial\alpha}{\partial\boldsymbol{y}} = \boldsymbol{x}^T\boldsymbol{A}^T$. Let $\boldsymbol{p} = \boldsymbol{A}\boldsymbol{x}$. Then, $\alpha = \boldsymbol{y}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}^T\boldsymbol{p}$. Since α is a scalar, $\alpha = \alpha^T = \boldsymbol{p}^T\boldsymbol{y}$. Using the result above,

$$\frac{\partial\alpha}{\partial\boldsymbol{y}} = \boldsymbol{p}^T = (\boldsymbol{A}\boldsymbol{x})^T = \boldsymbol{x}^T\boldsymbol{A}^T$$

Some matrix differentiation formulas

Considering the quadratic form $\alpha = \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}$, where $\boldsymbol{A}$ is $\mathrm{n} \times n$

$$\frac{\partial \alpha}{\partial \boldsymbol{x}} = \boldsymbol{x}^T (\boldsymbol{A} + \boldsymbol{A}^T)$$

By definition, $\alpha = \sum_{j=1}^{n} \sum_{i=1}^{n} a_{ij} x_i x_j$.

$\alpha = \sum_{j \neq k}^{n} x_j \sum_{i=1}^{n} a_{ij} x_i + x_k \sum_{i=1}^{n} a_{ik} x_i$

$= \sum_{j \neq k}^{n} x_j \sum_{i=1}^{n} a_{ij} x_i + x_k \sum_{i \neq k}^{n} a_{ik} x_i + a_{kk} x_k^2$

Some matrix differentiation formulas

$$\frac{\partial \alpha}{\partial x_k} = \sum_{j\neq k}^{n} x_j \sum_{i=1}^{n} a_{ij} \frac{\partial x_i}{\partial x_k} + \frac{\partial x_k}{\partial x_k} \left(\sum_{i\neq k}^{n} a_{ik} x_i\right) + 2a_{kk} x_k$$

$$= \sum_{j\neq k}^{n} a_{kj} x_j + \sum_{i\neq k}^{n} a_{ik} x_i + 2a_{kk} x_k$$

$$= \sum_{j=1}^{n} a_{kj} x_j + \sum_{i=1}^{n} a_{ik} x_i$$

Thus $\frac{\partial \alpha}{\partial \boldsymbol{x}} = \boldsymbol{x}^T (\boldsymbol{A} + \boldsymbol{A}^T)$

Some matrix differentiation formulas

If **A** is a symmetric matrix and $\alpha = \boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x}$, where $\boldsymbol{x}$ is $n \times 1$, **A** is $n \times n$ and **A** is not dependent on $\boldsymbol{x}$, then

$$\frac{\partial \alpha}{\partial \boldsymbol{x}} = \boldsymbol{2x^T A}$$

Using the result, $\frac{\partial \alpha}{\partial \boldsymbol{x}} = \boldsymbol{x}^T\left(\boldsymbol{A} + \boldsymbol{A}^T\right)$, the proof is straight forward.

Some matrix differentiation formulas

If $\alpha = \boldsymbol{x}^T \boldsymbol{x}$, where $\boldsymbol{x}$ is $n \times 1$, then $\frac{\partial \alpha}{\partial \boldsymbol{x}} = \mathbf{2}\boldsymbol{x}^T$

$\alpha = \boldsymbol{x}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{I} \boldsymbol{x}$, where $\boldsymbol{I}$ is an identity matrix and using the previous result $\frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}) = \boldsymbol{x}^T(\boldsymbol{A} + \boldsymbol{A}^T)$, the proof is straight forward.

Some matrix differentiation formulas

Let $\alpha = \boldsymbol{y}^T \boldsymbol{x}$, where $\boldsymbol{y}$ is $n \times 1$, $\boldsymbol{x}$ is $n \times 1$ and both $\boldsymbol{y}$ and $\boldsymbol{x}$ are functions of the vector $\boldsymbol{z}$. Then,

$$\frac{\partial \alpha}{\partial \boldsymbol{z}} = \boldsymbol{x}^T \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{z}} + \boldsymbol{y}^T \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{z}}$$

Proof: $\alpha = \sum_{j=1}^{n} x_j y_j$. $\frac{\partial \alpha}{\partial z_k} = \sum_{j=1}^{n} x_j \frac{\partial y_j}{\partial z_k} + y_j \frac{\partial x_j}{\partial z_k}$

Thus, $\frac{\partial \alpha}{\partial \boldsymbol{z}} = \boldsymbol{x}^T \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{z}} + \boldsymbol{y}^T \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{z}}$

Some matrix differentiation formulas

Let $\alpha = \boldsymbol{x}^T\boldsymbol{x}$, where $\boldsymbol{x}$ is $n \times 1$ and $\boldsymbol{x}$ is a function of the vector $\boldsymbol{z}$. Then,

$$\frac{\partial \alpha}{\partial \boldsymbol{z}} = \mathbf{2}\boldsymbol{x}^T \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{z}}$$

Using the results in the previous slide, the proof is straight forward.

Some matrix differentiation formulas

Let $\alpha = \boldsymbol{y}^T \boldsymbol{A}\boldsymbol{x}$, where $\boldsymbol{y}$ is $m \times 1$, $\boldsymbol{x}$ is $n \times 1$ and $\boldsymbol{A}$ is $m \times n$. Both $\boldsymbol{y}$ and $\boldsymbol{x}$ are functions of the vector $\boldsymbol{z}$ but $\boldsymbol{A}$ is independent of $\boldsymbol{z}$. Then,

$$\frac{\partial \alpha}{\partial \boldsymbol{z}} = \boldsymbol{x}^T A^T \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{z}} + \boldsymbol{y}^T A \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{z}}$$

Some matrix differentiation formulas

Proof: Let $w^T = \boldsymbol{y}^T\boldsymbol{A}$. $\alpha = \boldsymbol{w}^T\boldsymbol{x}$. Using the results in slide 48, we have

$$\frac{\partial \alpha}{\partial z} = \boldsymbol{x}^T \frac{\partial \boldsymbol{w}}{\partial z} + \boldsymbol{w}^T \frac{\partial \boldsymbol{x}}{\partial z}.$$

Substituting $w^T = \boldsymbol{y}^T\boldsymbol{A}$ in it, we have

$$\frac{\partial \alpha}{\partial \boldsymbol{z}} = \boldsymbol{x}^T \frac{\partial \boldsymbol{A}^T\boldsymbol{y}}{\partial \boldsymbol{z}} + \boldsymbol{y}^T\boldsymbol{A}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{z}}$$

$$= \boldsymbol{x}^T\boldsymbol{A}^T \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{z}} + \boldsymbol{y}^T\boldsymbol{A}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{z}}$$

Some matrix differentiation formulas

Let $\alpha = \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}$, where $\boldsymbol{x}$ is $n \times 1$, $\boldsymbol{A}$ is $n \times n$ and $\boldsymbol{x}$ is a function of the vector $\boldsymbol{z}$ but $\boldsymbol{A}$ is independent of $\boldsymbol{z}$. Then,

$$\frac{\partial \alpha}{\partial \boldsymbol{z}} = \boldsymbol{x}^T (\boldsymbol{A}^T + \boldsymbol{A}) \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{z}}$$

Using the result in the slide 50, it can be proven easily.

Some matrix differentiation formulas

Let $\alpha = \boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x}$, where $\boldsymbol{x}$ is $n \times 1$, $\boldsymbol{A}$ is a $n \times n$ symmetric matrix and $\boldsymbol{x}$ is a function of the vector $\boldsymbol{z}$ but $\boldsymbol{A}$ is independent of $\boldsymbol{z}$. Then,

$$\frac{\partial \alpha}{\partial \boldsymbol{z}} = 2\boldsymbol{x}^T \boldsymbol{A} \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{z}}$$

Using the result in the slide 52, it can be proven easily.

More formulas

K. B. Peteren and M. S. Pedersen, The Matrix Codebook

https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf