

CPEN 455 24W2 TUTORIAL  
LINEAR ALGEBRA & MATRIX CALCULUS<sup>1</sup>

**Qi Yan**

University of British Columbia

January 20, 2025

---

<sup>1</sup>Based on materials from Murphy, 2022; Parr and Howard, 2018.

# CONTENTS

<b>1</b>	<b>Linear algebra</b>	<b>2</b>
1.1	Vector and matrix fundamentals	2
1.2	Vector norms	7
<b>2</b>	<b>Matrix calculus</b>	<b>8</b>
2.1	Notations	8
2.2	Scalar derivative rules	9
2.3	Vector calculus and partial derivatives	11
2.4	Generalization of the Jacobian	16
2.5	Geometric understanding of Jacobians	20
2.6	Vector sum reduction	23
2.7	The Chain rules	25
<b>3</b>	<b>Common results</b>	<b>30</b>
3.1	Gradients and Jacobians	30
3.2	Scalar expansion	31
3.3	Vector reductions	32
3.4	Chain rules	33
<b>4</b>	<b>References</b>	<b>34</b>

# LINEAR ALGEBRA

## VECTOR AND MATRIX FUNDAMENTALS

- ▶ A **vector**  $\mathbf{x} \in \mathbb{R}^n$  is a list of  $n$  numbers, usually written as a column vector:

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

- ▶  $x_i$  or  $\mathbf{x}[i]$  is the  $i^{\text{th}}$  element of vector  $\mathbf{x}$  (scalar).
- ▶ The vector of all ones is denoted  $\mathbf{1}$ . The vector of all zeros is denoted  $\mathbf{0}$ .
- ▶ The unit vector  $\mathbf{e}_i$  is a vector of all 0's, except entry  $i$ , which has value 1:

$$\mathbf{e}_i = (0, \dots, 0, 1, 0, \dots, 0)$$

This is also called a one-hot vector.

- ▶ The **dot product**  $\mathbf{w} \cdot \mathbf{x}$  is the summation of the element-wise multiplication of the elements:  
 $\sum_i (w_i x_i) = \text{sum}(\mathbf{w} \otimes \mathbf{x})$ . Or, you can look at it as  $\mathbf{w}^T \mathbf{x}$ .

# LINEAR ALGEBRA

## VECTOR AND MATRIX FUNDAMENTALS

A **matrix**  $A \in \mathbb{R}^{m \times n}$  with  $m$  rows and  $n$  columns is a 2d array of numbers, arranged as follows:

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

If  $m = n$ , the matrix is said to be **square**.

- ▶  $A_{ij}$  or  $A[i, j]$ : the entry of  $A$  in the  $i$ th row and  $j$ th column.
- ▶  $A[i, :]$ : the  $i$ th row and  $A[:, j]$ : the  $j$ th column.
- ▶ We treat all vectors as column vectors by default (e.g.,  $A[:, j]$ ).

We can view a matrix as a set of columns stacked along the horizontal axis:

$$A = \left[ \begin{array}{c|c|c|c} | & | & \cdots & | \\ A[:, 1] & A[:, 2] & \cdots & A[:, n] \\ | & | & \cdots & | \end{array} \right] = [A[:, 1], A[:, 2], \dots, A[:, n]].$$

We can also view a matrix as a set of rows stacked along the vertical axis:

$$A = \left[ \begin{array}{c} - & A[1, :] & - \\ - & A[2, :] & - \\ & \vdots & \\ - & A[m, :] & - \end{array} \right] = [A[1, :]; A[2, :]; \dots; A[m, :]].$$

# LINEAR ALGEBRA

## VECTOR AND MATRIX FUNDAMENTALS

- ▶ The **transpose** of a matrix results from “flipping” the rows and columns. Given a matrix  $A \in \mathbb{R}^{m \times n}$ , its transpose, written  $A^T \in \mathbb{R}^{n \times m}$ , is defined as

$$(A^T)[i, j] = A[j, i]$$

- ▶ The following properties of transposes are easily verified:

$$(A^T)^T = A$$

$$(AB)^T = B^T A^T$$

$$(A + B)^T = A^T + B^T$$

- ▶ If a square matrix satisfies  $A = A^T$ , it is called **symmetric**. We denote the set of all symmetric matrices of size  $n$  as  $\mathbb{S}^n$ .
- ▶  $I$  represents the square **identity matrix** of appropriate dimensions that is zero everywhere but the diagonal, which contains all ones. We may also use lower script to indicate dimension, e.g.,  $I_M$ .
- ▶  $diag(\mathbf{x})$  constructs a matrix whose diagonal elements are taken from vector  $\mathbf{x}$ .

# LINEAR ALGEBRA

## VECTOR AND MATRIX FUNDAMENTALS

- ▶ A **tensor** (in machine learning) is a generalization of a 2d array to higher dimensions. For example, the entries of a 3d tensor are denoted by  $A[ijk]$ .

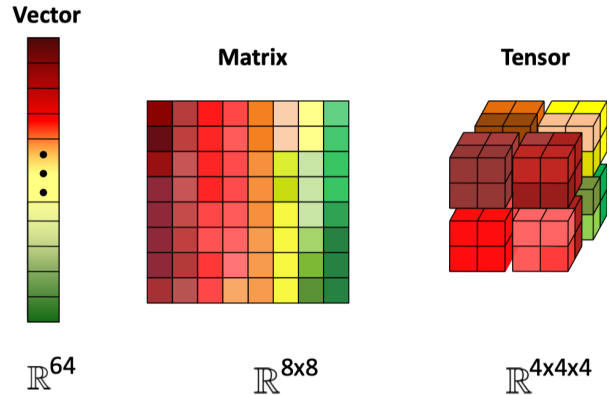


Image credit: Murphy, 2022

- ▶ The number of dimensions is known as the **order** or **rank** of the tensor.

# LINEAR ALGEBRA

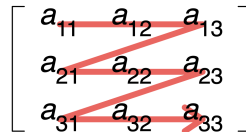
## VECTOR AND MATRIX FUNDAMENTALS

- ▶ We can **reshape** a matrix into a vector by stacking its columns on top of each other. This is denoted by

$$\text{vec}(A) = [A[:, 1]; \cdots ; A[:, n]] \in \mathbb{R}^{mn \times 1}$$

- ▶ Conversely, we can reshape a vector into a matrix. There are two choices: **row-major order** (used by Python/PyTorch) and **column-major order** (used by Matlab and R).

Row-major order



Column-major order

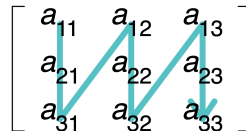


Image credit: Wikipedia<sup>2</sup>

<sup>2</sup>[https://en.wikipedia.org/wiki/Row-\\_and\\_column-major\\_order](https://en.wikipedia.org/wiki/Row-_and_column-major_order)

# LINEAR ALGEBRA

## VECTOR NORMS

A **norm** of a vector  $\|\mathbf{x}\|$  measures of the “length” of the vector. More formally, a norm is any function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  that satisfies 4 properties:

- ▶ For all  $\mathbf{x} \in \mathbb{R}^n$ ,  $f(\mathbf{x}) \geq 0$  (non-negativity).
- ▶  $f(\mathbf{x}) = 0$  if and only if  $\mathbf{x} = \mathbf{0}$  (definiteness).
- ▶ For all  $\mathbf{x} \in \mathbb{R}^n$ ,  $t \in \mathbb{R}$ ,  $f(t\mathbf{x}) = |t|f(\mathbf{x})$  (absolute value homogeneity).
- ▶ For all  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ ,  $f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y})$  (triangle inequality).

Consider the following common examples:

- ▶ **p-norm**  $\|\mathbf{x}\|_p = (\sum_{i=1}^n |x_i|^p)^{1/p}$ , for  $p \geq 1$ .
- ▶ **2-norm**  $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$ , also called *Euclidean norm*. Note that  $\|\mathbf{x}\|_2^2 = \mathbf{x}^\top \mathbf{x}$ .
- ▶ **1-norm**  $\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$ .
- ▶ **Max-norm**  $\|\mathbf{x}\|_\infty = \max_i |x_i|$ .
- ▶ **0-norm**  $\|\mathbf{x}\|_0 = \sum_{i=1}^n \mathbb{I}(|x_i| > 0)$ . This is a **pseudo norm**, since it does not satisfy homogeneity. It counts the number of non-zero elements in  $\mathbf{x}$ .



# MATRIX CALCULUS

## NOTATIONS

- ▶ **Differentiation**  $\frac{d}{dx}$  is an operator that maps a function of one parameter to another function. That means that  $\frac{d}{dx}f(x)$  maps  $f(x)$  to its derivative with respect to  $x$ , which is the same thing as  $\frac{df(x)}{dx}$ . Also, if  $y = f(x)$ , then  $\frac{dy}{dx} = \frac{df(x)}{dx}$ .
- ▶ The **partial derivative** of the function with respect to  $x$ ,  $\frac{\partial f(x)}{\partial x}$ , performs the usual scalar derivative holding all other variables constant.
- ▶ The **gradient** of  $f$  with respect to vector  $\mathbf{x}$ ,  $\nabla f(\mathbf{x})$ , organizes all of the partial derivatives for a specific scalar function.

$$\nabla f(\mathbf{x}) = \left[ \frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \dots, \frac{\partial f(x)}{\partial x_n} \right]$$

- ▶ The **Jacobian** organizes the gradients of multiple functions into a matrix by stacking them:

$$J = \begin{bmatrix} \nabla f_1(\mathbf{x}) \\ \nabla f_2(\mathbf{x}) \end{bmatrix}$$

- ▶ **We will cover more concrete examples later.**

# MATRIX CALCULUS

## SCALAR DERIVATIVE RULES

Rule	$f(x)$	Scalar derivative w.r.t. $x$	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	$fg$	$f \frac{dg}{dx} + g \frac{df}{dx}$	$\frac{d}{dx} (x^2 x) = x^2 + x(2x) = 3x^2$
Chain Rule	$f(g(x))$	$\frac{df(u)}{du} \frac{du}{dx}$ , let $u = g(x)$	$\frac{d}{dx} \log(x^2) = \frac{1}{x^2} 2x = \frac{2}{x}$

- ▶ More rules can be found on Wikipedia<sup>3</sup>.
- ▶ When a function has a single parameter,  $f(x)$ , you'll often see  $f'(x)$  used as shorthands for  $\frac{d}{dx} f(x)$ . We **recommend against** using this notation ( $f'(x)$ ) as it does not make clear the variable we're taking the derivative with respect to.

<sup>3</sup>[https://en.wikipedia.org/wiki/List\\_of\\_derivatives\\_and\\_integrals\\_in\\_alternative\\_calculi](https://en.wikipedia.org/wiki/List_of_derivatives_and_integrals_in_alternative_calculi)

# MATRIX CALCULUS

## SCALAR DERIVATIVE RULES

- ▶ You can think of  $\frac{d}{dx}$  as an operator. This helps to simplify complicated derivatives because the operator is distributive and lets us pull out constants.

### Example 2.1 (Viewing $\frac{d}{dx}$ as an operator)

For example, in the following equation, we can pull out the constant 9 and distribute the derivative operator across the elements in the parentheses.

$$\frac{d}{dx}9(x + x^2) = 9\frac{d}{dx}(x + x^2) = 9(1 + 2x) = 9 + 18x$$

- ▶ That procedure reduced the derivative of  $9(x + x^2)$  to a bit of arithmetic and the derivatives of  $x$  and  $x^2$ , which are much easier to solve than the original derivative.

# MATRIX CALCULUS

## VECTOR CALCULUS AND PARTIAL DERIVATIVES

- ▶ Neural network layers are not single functions of a single parameter,  $f(x)$ . We care about functions of multiple parameters such as  $f(x, y)$ . For example, what is the derivative of  $f(x, y) = xy$ ?
- ▶ We compute derivatives with respect to one variable ( $x$  or  $y$ ) at a time, giving us two different *partial derivatives* for this two-parameter function (one for  $x$  and one for  $y$ ).
- ▶ Instead of using operator  $\frac{d}{dx}$ , the partial derivative operator is  $\frac{\partial}{\partial x}$ . So,  $\frac{\partial}{\partial x}xy$  and  $\frac{\partial}{\partial y}xy$  are the partial derivatives of  $xy$ ; often, these are just called the *partials*<sup>4</sup>.

---

<sup>4</sup>For functions of a single parameter, operator  $\frac{\partial}{\partial x}$  is equivalent to  $\frac{d}{dx}$  (for sufficiently smooth functions). However, it's better to use  $\frac{d}{dx}$  to make it clear you're referring to a scalar derivative.

# MATRIX CALCULUS

## VECTOR CALCULUS AND PARTIAL DERIVATIVES

- ▶ The partial derivative with respect to  $x$  is just the usual scalar derivative, simply treating any other variable in the equation as a constant.

### Example 2.2 (Partial derivative)

Consider function  $f(x, y) = 3x^2y$ . The partial derivative with respect to  $x$  is written as  $\frac{\partial}{\partial x} 3x^2y$ . There are three constants from the perspective of  $\frac{\partial}{\partial x}$ : 3, 2, and  $y$ . Therefore,  $\frac{\partial}{\partial x} 3x^2y = 3 \cdot 2x \cdot y = 6xy$ .

The partial derivative with respect to  $y$  treats  $x$  like a constant:  $\frac{\partial}{\partial y} 3x^2y = 3x^2 \cdot \frac{\partial y}{\partial y} = 3x^2 \cdot 1 = 3x^2$ .

# MATRIX CALCULUS

## VECTOR CALCULUS AND PARTIAL DERIVATIVES

- ▶ To make it clear we are doing *vector calculus* and not just multivariate calculus, let's consider what we do with the partial derivatives  $\frac{\partial f(x,y)}{\partial x}$  and  $\frac{\partial f(x,y)}{\partial y}$  (another way to say  $\frac{\partial}{\partial x} f(x,y)$  and  $\frac{\partial}{\partial y} f(x,y)$ ).

### Example 2.3 (Partials and gradients)

Again, consider  $f(x, y) = 3x^2y$ . Instead of having them just floating around and not organized in any way, let's organize them into a horizontal vector. We call this vector the *gradient* of  $f(x, y)$  and write it as:

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

- ▶ **The gradient of  $f(x, y)$  is simply a vector of its partials.** Gradients deals with functions that map  $n$  scalar parameters to a single scalar.

# MATRIX CALCULUS

## VECTOR CALCULUS AND PARTIAL DERIVATIVES

- ▶ When we move from derivatives of one function to derivatives of many functions, we move from the world of vector calculus to matrix calculus.

### Example 2.4 (Another example on gradients)

Let's compute partial derivatives for two functions, both of which take two parameters. We can keep the same  $f(x, y) = 3x^2y$  from the last section, but let's also bring in  $g(x, y) = 2x + y^8$ . The gradient for  $g$  has two entries, a partial derivative for each parameter:

$$\frac{\partial g(x, y)}{\partial x} = \frac{\partial}{\partial x} 2x + \frac{\partial}{\partial x} y^8 = 2 \frac{\partial x}{\partial x} + 0 = 2 \times 1 = 2$$

and

$$\frac{\partial g(x, y)}{\partial y} = \frac{\partial}{\partial y} 2x + \frac{\partial}{\partial y} y^8 = 0 + 8y^7 = 8y^7$$

giving us gradient  $\nabla g(x, y) = [2, 8y^7]$ .

# MATRIX CALCULUS

## VECTOR CALCULUS AND PARTIAL DERIVATIVES

- ▶ Gradient vectors organize all of the partial derivatives for a specific scalar function. If we have two functions, we can also organize their gradients into a matrix by stacking the gradients. When we do so, we get the **Jacobian matrix** (or just the **Jacobian**) where the gradients are rows:

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

- ▶ Note that there are multiple ways to represent the Jacobian. We are using the so-called *numerator layout* but many papers and software will use the *denominator layout*. This is just transpose of the numerator layout Jacobian (flip it around its diagonal):

$$\begin{bmatrix} 6yx & 3x^2 \\ 2 & 8y^7 \end{bmatrix}^T = \begin{bmatrix} 6yx & 2 \\ 3x^2 & 8y^7 \end{bmatrix}$$

- ▶ **We recommend using the *numerator layout* in your written homework and programming assignments.**



# MATRIX CALCULUS

## GENERALIZATION OF THE JACOBIAN

- ▶ To define the Jacobian matrix more generally, let's combine multiple parameters into a single vector argument:  $f(x, y, z)$  becomes  $f(\mathbf{x})$ . We assume that all vectors are vertical by default of size  $n \times 1$ :

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

- ▶ With multiple scalar-valued functions, we can combine them all into a vector. Let  $\mathbf{y} = f(\mathbf{x})$  be a vector of  $m$  scalar-valued functions that each take a vector  $\mathbf{x}$  of length  $n$  ( $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ).
- ▶ Each  $f_j$  function within  $f$  returns a scalar just as in the previous section:

$$\begin{aligned} y_1 &= f_1(\mathbf{x}) \\ y_2 &= f_2(\mathbf{x}) \\ &\vdots \\ y_m &= f_m(\mathbf{x}) \end{aligned}$$

# MATRIX CALCULUS

## GENERALIZATION OF THE JACOBIAN

### Example 2.5 (Jacobians)

For instance, we'd represent  $f(x, y) = 3x^2y$  and  $g(x, y) = 2x + y^8$  from the last section as

$$\begin{aligned}y_1 &= f_1(\mathbf{x}) = 3x_1^2x_2 && \text{(substituting } x_1 \text{ for } x, x_2 \text{ for } y) \\y_2 &= f_2(\mathbf{x}) = 2x_1 + x_2^8\end{aligned}$$

It's very often the case that  $m = n$  because we will have a scalar function result for each element of the  $\mathbf{x}$  vector. For example, consider the identity function  $y = f(\mathbf{x}) = \mathbf{x}$ :

$$\begin{aligned}y_1 &= f_1(\mathbf{x}) = x_1 \\y_2 &= f_2(\mathbf{x}) = x_2 \\&\vdots \\y_n &= f_n(\mathbf{x}) = x_n\end{aligned}$$

So we have  $m = n$  functions and parameters, in this case.

# MATRIX CALCULUS

## GENERALIZATION OF THE JACOBIAN

- ▶ Generally speaking, the Jacobian matrix is the collection of all  $m \times n$  possible partial derivatives ( $m$  rows and  $n$  columns), which is the stack of  $m$  gradients with respect to  $\mathbf{x}$ :

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

- ▶ Each  $\frac{\partial f_i(\mathbf{x})}{\partial \mathbf{x}}$  is a horizontal  $n$ -vector. The width of the Jacobian is  $n$  if we're taking the partial derivative with respect to  $\mathbf{x}$  because there are  $n$  parameters in  $\mathbf{x}$ , each potentially changing the function's value.
- ▶ Therefore, the Jacobian is always  $m$  rows for  $m$  equations.

# MATRIX CALCULUS

## GENERALIZATION OF THE JACOBIAN

### Example 2.6 (Jacobian of identity function)

The Jacobian of the identity function  $\mathbf{f}(\mathbf{x}) = \mathbf{x}$ , with  $f_i(\mathbf{x}) = x_i$ , has  $n$  functions and each function has  $n$  parameters held in a single vector  $\mathbf{x}$ . The Jacobian is, therefore, a square matrix since  $m = n$ :

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

(because  $\frac{\partial x_i}{\partial x_j} = 0$  for  $i \neq j$ ) which simplifies to

$$\begin{bmatrix} \frac{\partial x_1}{\partial x_1} & 0 & \dots & 0 \\ 0 & \frac{\partial x_2}{\partial x_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\partial x_n}{\partial x_n} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} = I,$$

where  $I \in \mathbb{R}^{n \times n}$  is the identity matrix with ones down the diagonal.

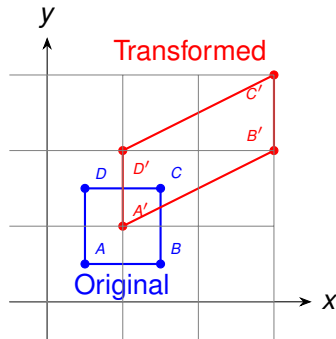
# MATRIX CALCULUS I

## GEOMETRIC UNDERSTANDING OF JACOBIANS

### Example 2.7 (Geometric meaning of Jacobians)

Consider  $\mathbf{f}(x, y) = \begin{bmatrix} u(x, y) \\ v(x, y) \end{bmatrix} = \begin{bmatrix} 2x \\ x + y \end{bmatrix}$ . Square  $ABCD$  is transformed into  $A'B'C'D'$

$$A = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, B = \begin{bmatrix} 1.5 \\ 0.5 \end{bmatrix}, C = \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}, D = \begin{bmatrix} 0.5 \\ 1.5 \end{bmatrix} \implies A' = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}, B' = \begin{bmatrix} 3.0 \\ 2.0 \end{bmatrix}, C' = \begin{bmatrix} 3.0 \\ 3.0 \end{bmatrix}, D' = \begin{bmatrix} 1.0 \\ 2.0 \end{bmatrix}. J = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix}$$



# MATRIX CALCULUS

## GEOMETRIC UNDERSTANDING OF JACOBIANS

- ▶ The Jacobian describes such *linear transformations* when the origin position remains unchanged.
- ▶ Reconsider the transformation between  $(x, y)$  and  $(u, v)$  coordinate systems above:

$$\begin{bmatrix} du \\ dv \end{bmatrix} = J \begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix}$$

- ▶ For *nonlinear transformations*, the Jacobian is a linear approximation in a small region (first order Taylor series approximation).

# MATRIX CALCULUS

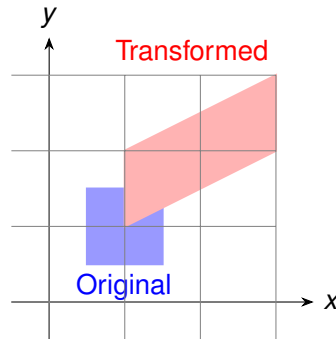
## GEOMETRIC UNDERSTANDING OF JACOBIANS

- ▶ Further, Jacobian determinant represents how much the *unit area* changes during a linear transformation.

$$du \times dv = |\det(J)|(dx \times dy)$$

$|\det(J)| > 1$ : expansion;  $|\det(J)| < 1$ : contraction.

- ▶ In the above example, area before transformation: 1, after transformation: 2.



# MATRIX CALCULUS

## VECTOR SUM REDUCTION

- ▶ Summing up the elements of a vector is an important operation in deep learning, such as the network loss function, but we can also use it as a way to simplify computing the derivative of vector dot product and other operations that reduce vectors to scalars.
- ▶ Consider a general case:  $y = \text{sum}(f(\mathbf{x})) = \sum_{i=1}^n f_i(\mathbf{x})$ . Notice we 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 gradient ( $1 \times n$  Jacobian) of vector summation is:

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



# MATRIX CALCULUS

## VECTOR SUM REDUCTION

### Example 2.8 (Vector sum reduction)

Let's look at the gradient of the simple  $y = \text{sum}(\mathbf{x})$ . The function inside the summation is just  $f_i(\mathbf{x}) = x_i$  and the gradient is then:

$$\nabla y = \left[ \sum_i \frac{\partial f_i(\mathbf{x})}{\partial x_1}, \sum_i \frac{\partial f_i(\mathbf{x})}{\partial x_2}, \dots, \sum_i \frac{\partial f_i(\mathbf{x})}{\partial x_n} \right] = \left[ \sum_i \frac{\partial x_i}{\partial x_1}, \sum_i \frac{\partial x_i}{\partial x_2}, \dots, \sum_i \frac{\partial x_i}{\partial x_n} \right]$$

Because  $\frac{\partial x_i}{\partial x_j} = 0$  for  $j \neq i$ , we can simplify to:

$$\nabla y = \left[ \frac{\partial x_1}{\partial x_1}, \frac{\partial x_2}{\partial x_2}, \dots, \frac{\partial x_n}{\partial x_n} \right] = [1, 1, \dots, 1] = \mathbf{1}^T$$

Notice that the result is a horizontal vector full of 1s, not a vertical vector, and so the gradient is  $\mathbf{1}^T$ .

It's very important **to keep the shape of all of your vectors and matrices in order**, otherwise it's impossible to compute the derivatives of complex functions.

# MATRIX CALCULUS

## THE CHAIN RULES

- ▶ We can't compute partial derivatives of very complicated functions using just the basic matrix calculus rules we've seen so far. For example, we can't take the derivative of nested expressions like  $\text{sum}(\mathbf{w} + \mathbf{x})$  directly without reducing it to its scalar equivalent.
- ▶ The chain rule is conceptually a divide and conquer strategy that breaks complicated expressions into subexpressions whose derivatives are easier to compute. We can process each simple subexpression in isolation yet still combine the intermediate results to get the correct overall result.
- ▶ For example, to compute  $\frac{d}{dx} \sin(x^2) = 2x \cos(x^2)$ , we can break it into  $\frac{d}{dx} x^2 = 2x$  and  $\frac{d}{du} \sin(u) = \cos(u)$ .

# MATRIX CALCULUS

## THE CHAIN RULES

- ▶ Chain rules are typically defined for nested functions, such as  $y = f(g(x))$  for single-variable chain rules (or using function composition  $f \circ g(x)$ ). The formulation of the single-variable chain rule is:

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

- ▶ To deploy the single-variable chain rule, follow these steps:
  1. Introduce intermediate variables for nested subexpressions. This step normalizes all equations to single operators or function applications.
  2. Compute derivatives of the intermediate variables with respect to their parameters.
  3. Combine (*chain*) all derivatives of intermediate variables by multiplying them together.
  4. Substitute intermediate variables back in if any are referenced in the derivative equation.

# MATRIX CALCULUS

## THE CHAIN RULES

- ▶ We now discuss the chain rule for vectors of functions and vector variables. We can start by computing the derivative of a sample vector function with respect to a scalar,  $y = f(x)$ .

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

- ▶ We introduce two intermediate variables,  $g_1$  and  $g_2$ , one for each  $f_i$  so that  $y$  looks like  $y = f(\mathbf{g}(x))$ :

$$\mathbf{g}(x) = \begin{bmatrix} g_1(x) \\ g_2(x) \end{bmatrix} = \begin{bmatrix} x^2 \\ 3x \end{bmatrix}$$

$$f_i(\mathbf{g}) = \begin{bmatrix} f_1(\mathbf{g}) \\ f_2(\mathbf{g}) \end{bmatrix} = \begin{bmatrix} \ln(g_1) \\ \sin(g_2) \end{bmatrix}$$

- ▶ The derivative of vector  $y$  w.r.t. scalar  $x$  is a vertical vector with elements computed using the single-variable chain rule:

$$\frac{\partial \mathbf{y}}{\partial x} = \begin{bmatrix} \frac{\partial f_1(\mathbf{g})}{\partial g_1} \frac{\partial g_1}{\partial x} & \frac{\partial f_1(\mathbf{g})}{\partial g_2} \frac{\partial g_2}{\partial x} \\ \frac{\partial f_2(\mathbf{g})}{\partial g_1} \frac{\partial g_1}{\partial x} & \frac{\partial f_2(\mathbf{g})}{\partial g_2} \frac{\partial g_2}{\partial x} \end{bmatrix} = \begin{bmatrix} \frac{1}{g_1} 2x + 0 \\ 0 + \cos(g_2) 3 \end{bmatrix} = \begin{bmatrix} \frac{2}{x} \\ 3 \cos(3x) \end{bmatrix}$$

- ▶ If we split the  $\frac{\partial f_i}{\partial g_j}$  terms, isolating the  $\frac{\partial g_j}{\partial x}$  terms into a vector, we get a matrix by vector multiplication:

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

# MATRIX CALCULUS

## THE CHAIN RULES

- ▶ That means that the Jacobian is the multiplication of two other Jacobians:

$$\begin{bmatrix} \frac{\partial f_1}{\partial g_1} & \frac{\partial f_1}{\partial g_2} \\ \frac{\partial f_2}{\partial g_1} & \frac{\partial f_2}{\partial g_2} \end{bmatrix} \begin{bmatrix} \frac{\partial g_1}{\partial x} \\ \frac{\partial g_2}{\partial x} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \cos(g_2) \end{bmatrix} \begin{bmatrix} 2x \\ 3 \end{bmatrix} = \begin{bmatrix} \frac{2}{x} \\ 3 \cos(3x) \end{bmatrix}$$

- ▶ We get the same answer as the scalar approach. This vector chain rule for vectors of functions appears to be consistent with the single-variable chain rule. Compare the vector rule:

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

with the single-variable chain rule:

$$\frac{d}{dx} f(g(x)) = \frac{df}{dg} \frac{dg}{dx}$$

- ▶ To make this formula work for multiple parameters or vector  $\mathbf{x}$ , we just have to change  $x$  to vector  $\mathbf{x}$  in the equation. The effect is that  $\frac{\partial \mathbf{g}}{\partial x}$  and the resulting Jacobian,  $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ , are now matrices instead of vertical vectors. Our complete vector chain rule is:

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

- ▶ Note: matrix multiply doesn't commute; order of  $\frac{\partial \mathbf{f}}{\partial \mathbf{g}} \frac{\partial \mathbf{g}}{\partial \mathbf{x}}$  matters.

# MATRIX CALCULUS

## THE CHAIN RULES

- ▶ In the vector chain rule, the Jacobian contains all possible combinations of  $f_i$  with respect to  $g_j$  and  $g_j$  with respect to  $x_j$ . For completeness, here are the two Jacobian components:

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

where  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^k$ ,  $\mathbf{f} : \mathbb{R}^k \rightarrow \mathbb{R}^m$ . The resulting Jacobian is  $m \times n$  (an  $m \times k$  matrix multiplied by a  $k \times n$  matrix).

- ▶ Even within this  $\frac{\partial \mathbf{f}}{\partial \mathbf{g}} \frac{\partial \mathbf{g}}{\partial \mathbf{x}}$  formula, we can simplify further because, for many applications, the Jacobians are square ( $m = n$ ) and the off-diagonal entries are zero.
- ▶ It is the nature of neural networks that the associated mathematics deals with functions of vectors not vectors of functions. For example, the ReLU activation function is  $\max(0, \mathbf{x})$ .

## COMMON RESULTS

### GRADIENTS AND JACOBIANS

- ▶ The **gradient** of a function of two variables is a horizontal 2-vector:

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

- ▶ The **Jacobian** of a vector-valued function that is a function of a vector is an  $m \times n$  ( $m = |\mathbf{f}|$  and  $n = |\mathbf{x}|$ ) matrix containing all possible scalar partial derivatives:

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

- ▶ The Jacobian of the identity function  $\mathbf{f}(\mathbf{x}) = \mathbf{x}$  is  $I$ .

## COMMON RESULTS

### SCALAR EXPANSION

- ▶ Adding scalar  $z$  to vector  $\mathbf{x}$ ,  $\mathbf{y} = \mathbf{x} + z$ , is really  $\mathbf{y} = f(\mathbf{x}) + g(z)$  where  $f(\mathbf{x}) = \mathbf{x}$  and  $g(z) = \mathbf{1}z$ .

$$\frac{\partial}{\partial \mathbf{x}}(\mathbf{x} + z) = \text{diag}(\mathbf{1}) = I$$

$$\frac{\partial}{\partial z}(\mathbf{x} + z) = \mathbf{1}^T$$

- ▶ Scalar multiplication yields:

$$\frac{\partial}{\partial \mathbf{x}}(\mathbf{x}z) = I z$$

$$\frac{\partial}{\partial z}(\mathbf{x}z) = \mathbf{x}$$



# COMMON RESULTS

## VECTOR REDUCTIONS

- ▶ The partial derivative of a vector sum with respect to one of the vectors is:

$$\nabla_{\mathbf{x}} \mathbf{y} = \left[ \sum_i \frac{\partial f_i(\mathbf{x})}{\partial x_1}, \sum_i \frac{\partial f_i(\mathbf{x})}{\partial x_2}, \dots, \sum_i \frac{\partial f_i(\mathbf{x})}{\partial x_n} \right]$$

- ▶ For  $\mathbf{y} = \text{sum}(\mathbf{x})$ :

$$\nabla_{\mathbf{x}} \mathbf{y} = \mathbf{1}^T$$

- ▶ For  $\mathbf{y} = \text{sum}(\mathbf{xz})$  and  $\mathbf{x} \in \mathbb{R}^n$ , we get:

$$\nabla_{\mathbf{x}} \mathbf{y} = [z, z, \dots, z] \in \mathbb{R}^n$$



$$\nabla_{\mathbf{z}} \mathbf{y} = \text{sum}(\mathbf{x})$$

# COMMON RESULTS

## CHAIN RULES

- ▶ Single-variable rule:  $\frac{df}{dx} = \frac{df}{du} \frac{du}{dx}$
- ▶ Single-variable (total-derivative) rule:  $\frac{\partial f(u_1, \dots, u_n)}{\partial x} = \frac{\partial f}{\partial \mathbf{u}} \frac{d\mathbf{u}}{dx}$
- ▶ Vector rule:  $\frac{\partial}{\partial \mathbf{x}} \mathbf{f}(\mathbf{g}(\mathbf{x})) = \frac{\partial \mathbf{f}}{\partial \mathbf{g}} \frac{\partial \mathbf{g}}{\partial \mathbf{x}}$

## REFERENCES

-  Murphy, K. P. (2022). ***Probabilistic machine learning: An introduction***. MIT Press.
-  Parr, T., & Howard, J. (2018). **The matrix calculus you need for deep learning**. *arXiv preprint arXiv:1802.01528*.