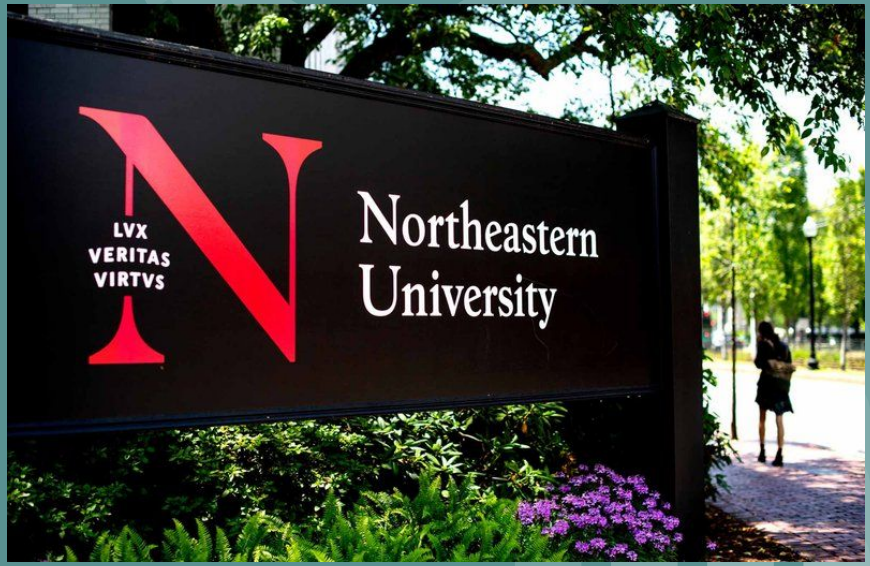
Machine Learning 2
DS4420: Lecture 2

Review of the basics notations

By Professor Wu



We are a family (at least for this semester)

Let’s act that way.

Family exercise:

- Stand up and find a person you don’t know
- Look at the person in the eyes (not anywhere else)
- Do handshake, fist bump or whatever with the person
- Say: Hello, my name is ....
  I grew up from ...
  I am on my ....th year
  I love to ....
  If I have any questions for this class,
  would you help me?

(Do this really fast, we only have 3 min to do this)

Mathematics is a beautiful Language

- Mathematics is really just another language like French, Spanish, or Chinese.
- Depending on the language, certain accents and languages sound beautiful while other ones sound like a choking cat.
- However, beauty is not the only characteristic of languages. We can also judge a language based on how fast it can transmit an idea.
- Some of the ugliest sounding language to our ears, happens to be the fastest to transmit the same idea.
- I always remember the first time I read a novel simultaneously in Spanish, Chinese, and English.
- That was when I realized that for the same chapter, the Chinese book was a lot shorter while saying exactly the same thing.
- Mathematics is one of those beautiful language.
- And it has the power to transmit the most amount of information with the least amount of writing.

For example, if we want to say that a function take 3 dimensions of real(non-imaginary) values as input and output 4 dimension.

- Notice how much writing is required to transmit this idea (basically everything in blue)
- The exact same idea can be written simply as

$$f : \mathbb{R}^3 \to \mathbb{R}^4.$$

- As we progress with the semester, I would like you to pay special attention to the meaning of symbols.
- These are the vocabulary of a new language that you will understand.

Matrix and Vectors are important

But the ability to find derivatives is **even more important**.

What is a derivative?

Function and the derivative

In machine learning, we use different functions to describe various natural phenomena. The function essentially tells us that

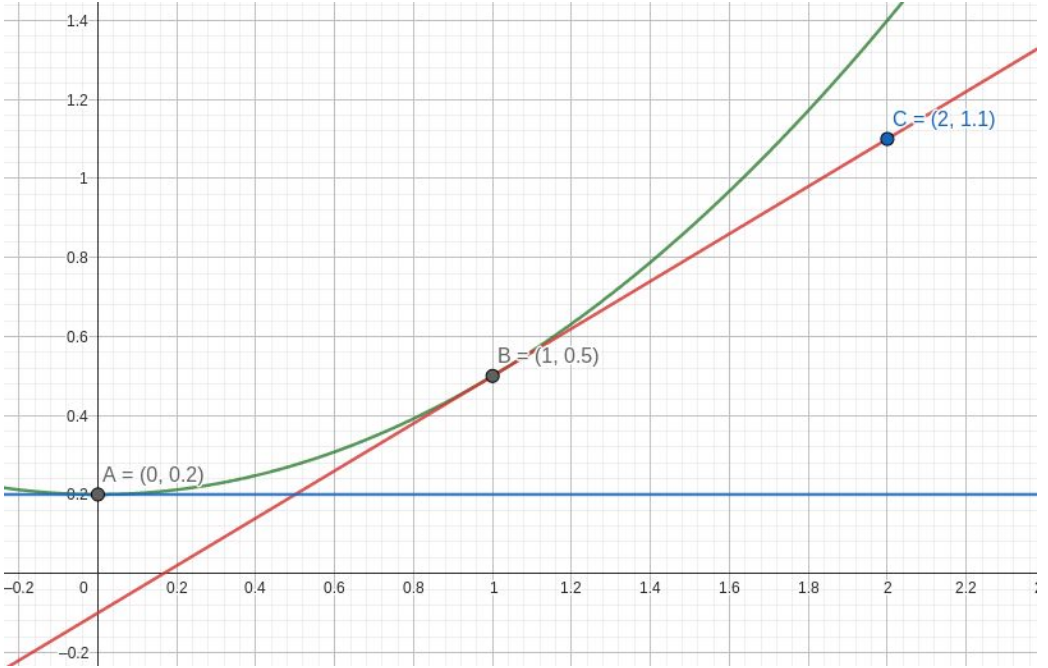
- given some input value x
- what is the output value $f(x)$

For example, given the function $f(x) = 0.3x^2 + 0.2$, shown as the green line on the right

- If we plug in 0, we would get 0.2
- If we plug in 1, we would get 0.5.

Given a function $f(x)$, we often want to know if the function going **up** or **down**.

- The derivatives or slope at a point gives us that information.
- The derivative at a point is a positive or negative number.
- if the derivative is positive, the function is going up.
- if the derivative is negative, the function is going down.
- The magnitude of the derivative tells you how quickly the function is going up or down.



Function and the derivative

- We can calculate the derivative/slope at any point by finding the derivative function.
- We call this finding the derivative of a function. We symbolically represent this process by writing

$$\frac{d}{dx}f(x) \xrightarrow{\text{taking the derivative}} \underbrace{f'(x) \quad \text{or} \quad \frac{df}{dx}(x)}_{\text{the derivative function.}}$$

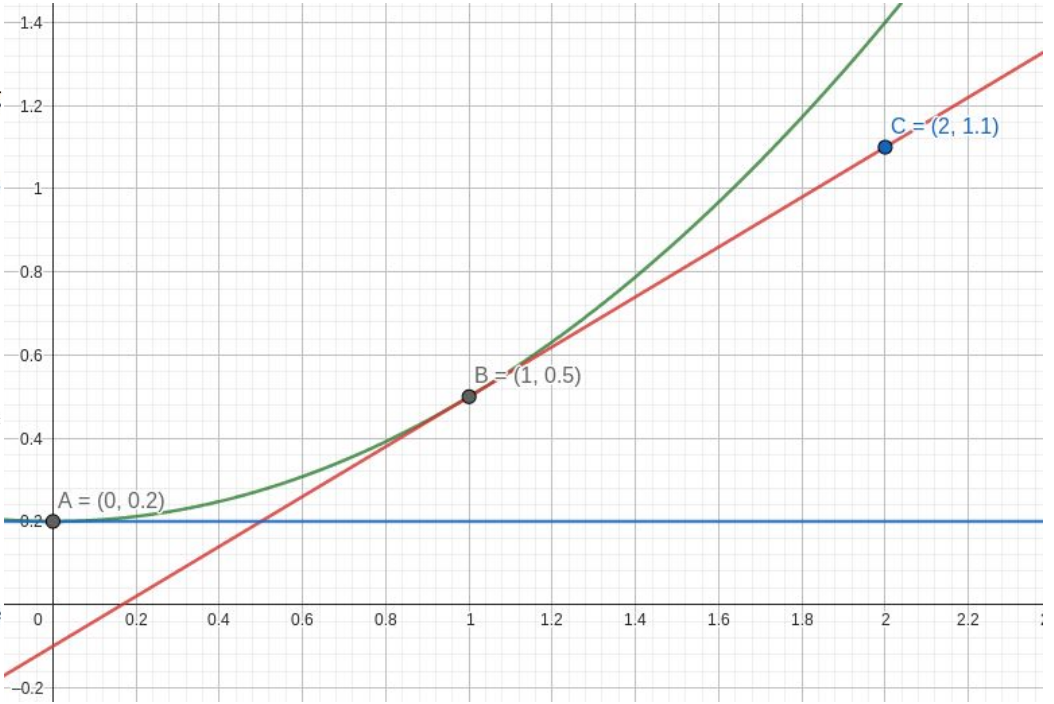
- For example, if we took the derivative for the function $f(x) = 0.3x^2 + 0.2$ (as the green line), then we would write

$$\frac{d}{dx}\, 0.3x^2 + 0.2 \quad \Longrightarrow \quad f'(x) = 0.6x.$$

- If we plug in the value of $x = 1$ into $f'(x)$, it would give us the derivative at the point $x = 1$ which is

$$f'(x = 1) = 0.6(1) = 0.6.$$

- Implying that if x is increased by 1, then $f(x)$ would increase by 0.6.
- If we look at the plot, that's exactly what happened.
- From $f'(x)$, it tells us if the function is going up or down at any point.



Functions with multiple input and output

A function can have multiple inputs and outputs. Here is an example where you have 3 inputs $\{x_1, x_2, x_3\}$ and 1 output.

$$f(x_1, x_2, x_3) = x_1 + 2x_2 - 3x_3.$$

Here is another example where we have 3 inputs and 2 outputs

$$f_1(x_1, x_2, x_3) = 2x_1 - x_3$$
$$f_2(x_1, x_2, x_3) = x_1 + x_2.$$

Since there are multiple variables, the derivative function would be symbolically represented as a vector where

$$\frac{d}{dx} f(x_1, x_2, x_3) = \begin{bmatrix} \frac{d}{dx_1} f \\ \frac{d}{dx_2} f \\ \frac{d}{dx_3} f \end{bmatrix}$$

Essentially, we would need to calculate the derivative 3 times. **With all these talks about derivatives, why do we care about them for machine learning and data science?**

- Let's look at an example.

Probability of you getting cancer.

- Let's say we can measure 3 molecules from your blood and predict the probability of you getting cancer.
- Let's call the 3 factors $x = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^\top$ and a function $f(x)$ that gives us the probability of you getting cancer given x.
- Don't worry about how we got this function. *We'll learn that later.*
- Assume we just measured your body for these 3 markers and got $x = \begin{bmatrix} 2 & 3 & 9 \end{bmatrix}^\top$ and when we plug your numbers into $f(x)$ we got 97% chance.
- Now also assume we plug your numbers into $f'(x)$ and got

$$f'\left(\begin{bmatrix} 2 \\ 3 \\ 9 \end{bmatrix}\right) = \begin{bmatrix} -0.001 \\ -6 \\ 20 \end{bmatrix} \tag{1}$$

- Take a moment and think about what the derivative information just told us.
- If you have 3 different drugs, each can increase x_1, x_2, or x_3. Which drug would you take if you want to lower the cancer rate? And which drug should you take if you are trying to die?

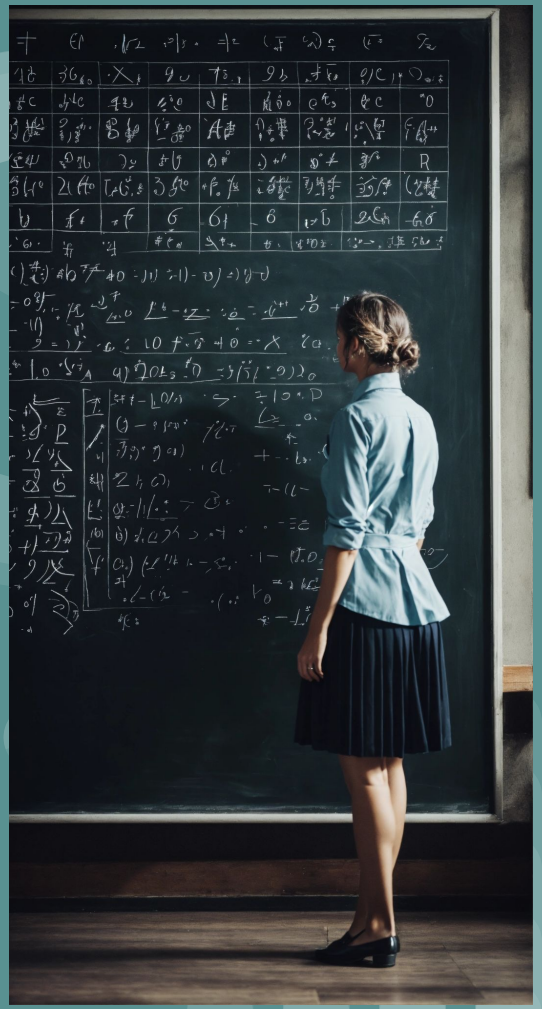
Why do we need derivative?

- From the derivative,

$$f'\left(\begin{bmatrix}2\\3\\9\end{bmatrix}\right) = \begin{bmatrix}-0.001\\-6\\20\end{bmatrix} \tag{1}$$

- We can see that if you take a drug that increases x_1, you will only very slightly decrease your probability.
- We can see that if you take a drug that increases x_2, you will significantly lower your chance of cancer.
- In contrast, if we increase x_3, it's going to increase the probability of cancer greatly. (From this, we can conclude that x_3 molecule is probably carcinogenic. You want to get rid of it.
- Machine learning is all about **decision making**, and the derivatives tell us which way we need to go.
- Again, over 99% of machine learning algorithms are about manipulating the input of some function to give us an ideal output.
- **How can you do that without knowing how to take derivatives?**

Finding Derivatives with matrices and vectors.



Combining Vectors and Matrices with Calculus

First, we need to realize that there are 2 ways to symbolically represent functions of multiple inputs

- Example of Method 1:

$$f(x_1, x_2, x_3) = x_1 + x_2^2 - 4x_3 \quad \text{or} \quad f(x_1, x_2) = e^{x_1} + \ln(x_2^2)$$

- This representation is probably the way you learned from calculus.
- This is perfectly fine if **you don't have too many inputs.**
- However, in machine learning, we could end up with thousands of inputs. If you have 5000 inputs, you wouldn't want to write this every time ...

$$f(x_1, x_2, x_3, ..., x_{5000})$$

- Instead, it makes more sense to let x be a vector $x = \begin{bmatrix} x_1 & x_2 & ... & x_{5000} \end{bmatrix}^\top$ and refer to the same function simply as

$$f(x) \quad \longleftarrow \text{we simply assume that } x \text{ is a vector.}$$

- For example:

$$f(x) = 2x_1 + 3x_2 - e^{-x}.$$

Derivative for Functions of single versus multiple inputs

- We previously learned how to take the derivative of a single input function like $f(x) = 2x - e^x$, resulting in

$$\frac{d}{dx} 2x - e^x = 2 - e^x.$$

- What does it mean when x is no longer a single variable, but an entire vector of variables?

$$\frac{d}{dx} f(x) = \frac{d}{dx} \underbrace{2x_1 - e^{x_2} - \ln(x_3)}_{f(x)} \quad \text{where} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

- The answer is very simple, for the previous example of 3 input function, we would simply have

$$\frac{d}{dx} f(x) = \begin{bmatrix} \frac{df}{dx_1}(x) \\ \frac{df}{dx_2}(x) \\ \frac{df}{dx_3}(x) \end{bmatrix} \quad \longrightarrow \quad \text{This results in a function of 3 input and 3 output.}$$

- Remember that for $\frac{df}{dx_1}$, only x_1 is a variable, everything else would be treated as a constant.
- The same logic follows with $\frac{df}{dx_2}$, and $\frac{df}{dx_3}$.
- Since x has 3 dimensions, we would put 3-dimensional data into this vector, resulting in a vector of size 3.

Pay Special Attention to how input/output dimension changed

- In our last example

$$\frac{d}{dx}f(x) = \frac{d}{dx}\underbrace{2x_1 - e^{x_2} - \ln(x_3)}_{f(x)} \quad \text{where} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{1}$$

- The answer is very simple, for the previous example of 3 input function, we would simply have

$$\frac{d}{dx}f(x) = \begin{bmatrix} \frac{df}{dx_1}(x) \\ \frac{df}{dx_2}(x) \\ \frac{df}{dx_3}(x) \end{bmatrix} \quad \longrightarrow \quad \text{This results in a function of 3 input and 3 output.}$$

- In other words, we have

$$f(x) : \mathbb{R}^3 \to \mathbb{R} \quad \text{and} \quad f'(x) : \mathbb{R}^3 \to \mathbb{R}^3$$

- The output dimension of the derivative function will always be the same dimension as the number of input data.

Quickly Consolidate what we just learned

1. What is $f'(y)$ if given

$$f(x) = x_1^2 - \ln(x_2) + e^{-x_3} \quad \text{and} \quad y = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

$$f(x) = x_1^2 - x_1 - \ln(x_2) + e^{-x_1+x_2} + 2x_2 - 3 \quad \text{and} \quad y = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Quickly Consolidate what we just learned

1. What is $f'(y)$ if given

$$f(x) = x_1^2 - \ln(x_2) + e^{-x_3} \quad \text{and} \quad y = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

$$f(x) = x_1^2 - x_1 - \ln(x_2) + e^{-x_1+x_2} + 2x_2 - 3 \quad \text{and} \quad y = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Solution 1:

$$f'(x) = \frac{d}{dx}\, x_1^2 - \ln(x_2) + e^{-x_3} = \begin{bmatrix} df/dx_1 \\ df/dx_2 \\ df/dx_3 \end{bmatrix} = \begin{bmatrix} 2x_1 \\ -\frac{1}{x_2} \\ -e^{-x_3} \end{bmatrix}$$

$$f'(y) = f'\left(\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} 2(1) \\ -\frac{1}{1} \\ -e^{-0} \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ -1 \end{bmatrix}$$

Solution 2:

$$f'\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 2x_1 - 1 - e^{-x_1+x_2} \\ -\frac{1}{x_2} + e^{-x_1+x_2} + 2 \end{bmatrix} = \begin{bmatrix} -e \\ -1 + e \end{bmatrix}$$

Matrix Vector Calculus

It is pretty easy to calculate the derivative, if all the problems look like this

$$f(x) = x_1^2 - \ln(x_2) + e^{-x_3} \quad \text{and} \quad y = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

However, most of the problem will look like this

- Given

$$x \in \mathbb{R}^d, y \in \mathbb{R}^d, w \in \mathbb{R}^d, A \in \mathbb{R}^{d \times d} \tag{1}$$

- Find the derivative of

1. $f(x) = y^\top x - w^\top x$
2. $f(x) = y^\top Ax$
3. $f(x) = y^\top Ax - y^\top x$
4. $f(x) = e^{y^\top Ax}$
5. $f(x) = ReLU(w^\top x)$

- In these cases, how do we calculate the derivative $\frac{df}{dx}$?
- This is the point of today's class.

Functions of machine learning algorithms.

Given $z \in \mathbb{R}, x, y, w \in \mathbb{R}^d, A \in \mathbb{R}^{d \times d}$

1. Linear: $f(x) = y^\top x$
2. Quadratic:
   $f(x) = x^\top A x$
3. Trace of Quadratic:
   $f(x) = \mathrm{Tr}(x^\top A x)$
4. Activation Function:
   $f(w) = \mathrm{ReLU}(w^\top x)$
5. Linear Regression:
   $f(w) = \frac{1}{2}\sum_{i=1}^{n}(w^\top x_i - y_i)^2$
6. Multivariate Gaussian Tr:
   $f(x) = e^{-\mathrm{Tr}(x^\top A x)}$
7. Multivariate Gaussian:
   $f(x) = e^{-x^\top A x}$
8. Sigmoid Function:
   $f(z) = \frac{1}{1+e^{-z}}$
9. Logistic Regression Objective:
   $f(w) = \frac{1}{1+e^{-w^\top x}}$
10. L1 Norm:
    $f(x) = ||x||_1$.
11. L2 Norm:
    $f(x) = ||x||_2$.
12. SVM objective:
    $f(w) = \sum_{i=1}^{n} \mathrm{ReLU}(1 - y_i\langle x_i, w\rangle)$
13. PCA objective:
    $f(x) = x^\top A x - \lambda(x^\top x - 1)$
14. Gaussian Maximum Likelihood:
    $\mu \in \mathbb{R}$
    $f(\mu) = \log\left(\prod_i^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}\right)$
15. Poisson Maximum Likelihood:
    $\lambda \in \mathbb{R}$
    $f(\lambda) = \log\left(\prod_i^n \lambda e^{-\lambda x_i}\right)$
16. Bernoulli Maximum Likelihood:
    α can only be 1 or 0, $\alpha \in \{0, 1\}$
    $p \in \mathbb{R}$
    $f(p) = \log\left(\prod_i^n p^{\alpha_i}(1-p)^{1-\alpha_i}\right)$
17. Uniform Maximum Likelihood:
    $a, b \in \mathbb{R} : a \leq b$
    $f(a, b) = \log\left(\prod_i^n \frac{1}{b-a}\right)$

- You don't need to know the usage of these functions for now.
- You just need to know how to take their derivative.
- The purpose of this class is to understand the purpose of these functions.
- We will go over some of them together.

Taking Derivatives of High Dimensional Data

While a lot of the equations we just saw appear intimidating, they are not hard at all

- They are simply combinations of simpler components. We will learn them today.
- For example, if we want to find $\frac{df}{dx}$ given

$$f(x) = y^\top x \quad \text{where} \quad y = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \longrightarrow \text{What is } \frac{df}{dx} \text{ when } x \text{ is a vector?} \quad \text{Answer: } \frac{df}{dx}(x) = \begin{bmatrix} \frac{df}{dx_1}(x) \\ \frac{df}{dx_2}(x) \end{bmatrix}$$

- All we have to do is to multiply the vectors out first

$$f(x) = y^\top x = \begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_1 \end{bmatrix} = \underbrace{2x_1 + x_2}_{\text{now it is easy to find } \frac{df}{dx_1}, \frac{df}{dx_2}}$$

- This rule applies for the 1st step of **any** vector/matrix derivative.
- You simply multiply everything (the matrices and vectors) out and then take the derivative 1 variable at a time.
- So as long as you know how to perform matrix/vector operations, you now also know how to take derivatives with respect to matrix and vectors.
- Once we calculate the partial derivatives, we then put the result back into vector/matrix format
- Let's see an example.

A Linear function

Here is an example we just saw, where given

$$f(x) = y^\top x \quad \text{where} \quad y = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \longrightarrow \text{What is } \frac{df}{dx} \text{ when } x \text{ is a vector?} \quad \text{Answer: } \frac{df}{dx}(x) = \begin{bmatrix} \frac{df}{dx_1}(x) \\ \frac{df}{dx_2}(x) \end{bmatrix}$$

And remember that we simply need to first multiply the vectors out first

$$f(x) = y^\top x = \begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_1 \end{bmatrix} = \underbrace{2x_1 + x_2}_{\text{now it is easy to find } \frac{df}{dx_1}, \frac{df}{dx_2}} \xrightarrow{\text{therefore}} \frac{df}{dx}(x) = \begin{bmatrix} \frac{df}{dx_1}(x) \\ \frac{df}{dx_2}(x) \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

Once we have obtained the derivative, we try to rewrite the solution back into matrix/vector format.

- Notice that the derivative is simply the y vector.
- Therefore,

$$\frac{d}{dx}\, y^\top x = \frac{df}{dx}(x) = \begin{bmatrix} \frac{df}{dx_1}(x) \\ \frac{df}{dx_2}(x) \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} = y. \quad \longrightarrow \quad \text{Allowing us to conclude that} \quad \underbrace{\frac{d}{dx}\, y^\top x = y}_{\text{This rule applies regardless of the dimension}}$$

- Don't believe me, try it yourself.

$$f(x) = y^\top x \quad \text{where} \quad y = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \longrightarrow \text{Is the solution simply } y?$$

The Quadratic equation

Given $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ and $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$, find the derivative of

$$f(x) = x^\top A x. \tag{1}$$

Step 1. Multiply all the variables out into a single equation, which results in

$$\begin{aligned} f(x) &= \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} && (2) \\ &= \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} a_{11}x_1 + a_{12}x_2 \\ a_{21}x_1 + a_{22}x_2 \end{bmatrix} && (3) \\ &= a_{11}x_1^2 + a_{12}x_1x_2 + a_{21}x_1x_2 + a_{22}x_2^2. && (4) \end{aligned}$$

Step 2. Take the partial derivative with respect to each element of the vector x.

$$\begin{aligned} f'(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} &= \begin{bmatrix} 2a_{11} & x_1 & + & (a_{12} + a_{21}) & x_2 \\ (a_{12} + a_{21}) & x_1 & + & 2a_{22} & x_2 \end{bmatrix} && (5) \\ &= \begin{bmatrix} 2a_{11} & (a_{12} + a_{21}) \\ (a_{12} + a_{21}) & 2a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} && (6) \\ &= \left(\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} + \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} \right) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} && (7) \end{aligned}$$

Step 3. Come up with a general rule of thumb next time you see a similar problem. Notice the pattern where

$$\begin{aligned} f'(x) &= \left(\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} + \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} \right) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} && (8) \\ &= (A + A^\top)x && (9) \end{aligned}$$

This function is special b/c it is the basic building block of many other functions.

The Trace of a quadratic form

Given $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, A = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix}$, the trace operation sum up the diagonal elements of a square matrix.

$$Tr(A) = Tr\left(\begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix}\right) = a_{1,1} + a_{2,2}. \tag{1}$$

Property 1 of Trace: given 1 dimension vector $x = [x_1]$

$$Tr(x) = Tr([x_1]) = x_1. \tag{2}$$

Therefore

$$Tr(x^\top Ax) = x^\top Ax \underbrace{\implies}_{\text{implying}} \frac{d}{dx} Tr(x^\top Ax) = \frac{d}{dx} x^\top Ax \tag{3}$$

Property 2 of Trace: Trace is rotation invariant

$$Tr(x^\top Ax) = Tr(Axx^\top) = Tr(xx^\top A) \tag{4}$$

therefore

$$\frac{d}{dx} Tr(x^\top Ax) = \frac{d}{dx} Tr(Axx^\top) = \frac{d}{dx} Tr(xx^\top A) = \frac{d}{dx} x^\top Ax \tag{5}$$

The regression objective

Given n samples $X \in R^{N \times d}$ and $Y \in R^N$, find the derivative of

$$f(w) = \frac{1}{2} \sum_i^N (w^\top x_i - y_i)^2.$$

Take out a piece of paper and see if you can find the derivative following the 3 steps

1. Multiply out the vector
2. Take the derivative individually, and fill them in to the vector in the same order
3. Try to represent the derivative in the original vector or matrix format, and come up with a general rule.

This function is special b/c it modern regression is based off of this format.

The Linear Regression Objective

Given n samples $X \in \mathbb{R}^{N \times d}$ and $Y \in \mathbb{R}^N$, find the derivative of

$$f(w) = \frac{1}{2} \sum_i^N (w^\top x_i - y_i)^2. \tag{1}$$

Here, we will let $q_i = w^\top x_i$, giving us

$$f(w) = \frac{1}{2} \sum_i^N (q_i - y_i)^2. \tag{2}$$

We can now find the derivative with the chain rule $f'(w) = \frac{df}{dq}\frac{dq}{dw}$

$$f'(w) = \sum_i^N \underbrace{(w^\top x_i - y_i)}_{\frac{df}{dq}} \underbrace{x_i}_{\frac{dq}{dw}}. \tag{3}$$

Note that we used the knowledge here that

$$\frac{d}{dw} w^\top x_i = x_i. \tag{4}$$

The ReLU function

46 The ReLU function

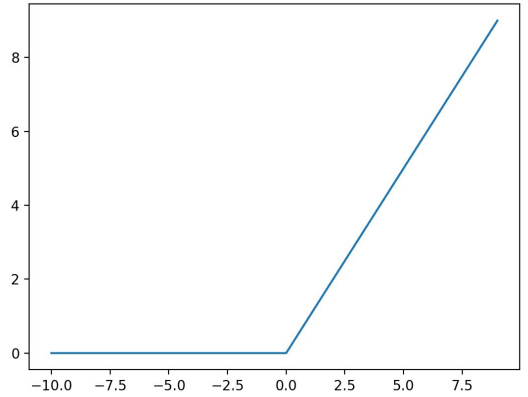
Given $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, find the derivative of

$$f(w) = ReLU(w^\top x).$$

Here, we will make use of chain rule again and let $q = w^\top x$, so we want to find $f'(w) = \frac{df}{dq}\frac{dq}{dw}$. If we look at the function of ReLU, we see that the derivative is 0 for negative values and 1 for positive values

$$f'(w) = \underbrace{1_{(w^\top x > 0)}}_{\frac{df}{dq}} \underbrace{x}_{\frac{dq}{dw}}.$$

This function is special b/c NN today use this function as activation.



The Multivariate Gaussian

Given $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, A = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix}$, find the derivative of

$$f(x) = e^{-x^\top Ax}. \tag{1}$$

We already know the derivative of $x^\top Ax$, so the result is simple

$$f'(x) = -e^{-x^\top Ax} \left[(A + A^\top)\right] x \tag{2}$$

The Sigmoid Function

Given 1 dimensional varible $x \in \mathbb{R}$, what's the derivative of

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

$$\sigma(x) = (1+e^{-x})^{-1}$$
$$\sigma'(x) = -1(1+e^{-x})^{-2}(e^{-x})(-1) = (1+e^{-x})^{-2}(e^{-x})$$

We next manipulate the result into a commonly seen form

$$(1+e^{-x})^{-2}(e^{-x}) = \frac{e^{-x}}{(1+e^{-x})(1+e^{-x})} \tag{4}$$
$$= \frac{1}{(1+e^{-x})}\frac{e^{-x}}{(1+e^{-x})} \tag{5}$$
$$= \frac{1}{(1+e^{-x})}\frac{-1+1+e^{-x}}{(1+e^{-x})} \tag{6}$$
$$= \frac{1}{(1+e^{-x})}\left(\frac{1+e^{-x}}{(1+e^{-x})} - \frac{1}{(1+e^{-x})}\right) \tag{7}$$
$$= \frac{1}{(1+e^{-x})}\left(1 - \frac{1}{(1+e^{-x})}\right) \tag{8}$$
$$= \sigma(x)(1-\sigma(x)) \tag{9}$$

This function is special b/c older NN all used this function.

The L1 norm

12 The L1 norm

Given $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ find the derivative of

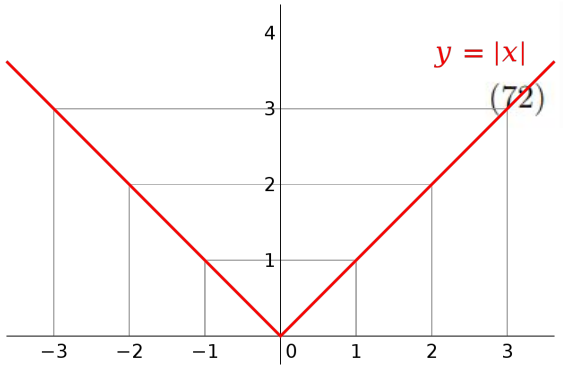
$$f(x) = ||x||_1 = |x_1| + |x_2| \tag{70}$$

If we look at the absolut value function, we know that if a value is positive, the derivative is 1 and -1 if value is negative. Therefore, the derivative is

$$f'(x) = \begin{bmatrix} 1_{(x_1>0)} - 1_{(x_1\leq 0)} \\ 1_{(x_2>0)} - 1_{(x_2\leq 0)} \end{bmatrix} = sign(x) \tag{71}$$

For example, given $v = \begin{bmatrix} 2 \\ -3 \end{bmatrix}$

$$f'(v) = sign\left(\begin{bmatrix} 2 \\ -3 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \tag{72}$$



This function is special b/c commonly used as a **regularizer** together with other functions.

The L2 norm

Given $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ find the derivative of

$$f(x) = ||x||_2 = \sqrt{x_1^2 + x_2^2} = (x_1^2 + x_2^2)^{1/2}$$

$$f'(x) = \frac{1}{2}(x_1^2 + x_2^2)^{-1/2} \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix}$$
$$= \frac{1}{(x_1^2 + x_2^2)^{1/2}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{1}{||x||_2} x$$

This function is special b/c commonly used as a regularizer together with other functions.

The PCA objective

Given $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, A = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix}$, find the derivative of

$$f(x) = x^T Ax - \lambda(x^T Ix - 1). \tag{1}$$

This is just a combination of functions we already know.

$$f'(x) = (A + A^\top)x - 2\lambda x. \tag{2}$$

Finding the derivative itself is not that special. The interesting part is that we can set the derivative to 0 and get

$$(A + A^\top)x - 2\lambda x = 0 \tag{3}$$

$$\frac{1}{2}(A + A^\top)x = \lambda x \tag{4}$$

$$Qx = \lambda x \tag{5}$$

This is the definition of the eigenvalue and vector. This is really interesting because, we know exactly where x needs to be for the derivative to be 0. Since

$$x^\top Qx = \lambda \tag{6}$$

this tells us that the eigenvector x associated with the largest eigenvalue will give you the highest value for $f(x)$, and in reverse, the smallest eigenvalue's eigenvector will give you smallest value.

This function is special b/c we do not need to search for the optimal solution, it is already known.

The Gaussian MLE objective

Given 1 dimensional varible $x \in \mathbb{R}$, what's the derivative of

$$f(\mu) = log\left(\prod_i^N \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}\right)$$

The trick is to know that $log(xy) = log(x) + log(y)$, therefore,

$$\begin{aligned} f(\mu) &= \sum_i^N log\left(\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}\right) \\ &= \sum_i^N log\left(\frac{1}{\sigma\sqrt{2\pi}}\right) + \sum_i^N log\left(e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}\right) \\ &= Nlog\left(\frac{1}{\sigma\sqrt{2\pi}}\right) - \sum_i^N \left(\frac{(x_i-\mu)^2}{2\sigma^2}\right) \end{aligned}$$

The derivative with respect to μ is now easy to find

$$f'(u) = \frac{1}{\sigma^2}\sum_i^N (x_i - u)$$

For this example, we can set the derivative to 0 and get an exact solution

$$0 = -\frac{1}{\sigma^2}\sum_i^N (x_i - \mu) \tag{6}$$

$$0 = \sum_i^N x_i - \sum_i^N \mu \tag{7}$$

$$0 = \sum_i^N x_i - N\mu \tag{8}$$

$$N\mu = \sum_i^N x_i \tag{9}$$

$$\mu = \frac{1}{N}\sum_i^N x_i \tag{10}$$

The optimal solution here is just the average of x_i.

The Uniform Distribution MLE objective

I want you to use the same trick and see if you can get the derivative for
(take 5 min to try it)

Given $v = [a, b]^\top$, what's the derivative of

$$f(a, b) = log\left(\prod_i^n \frac{1}{b-a}\right)$$

The Uniform Distribution MLE objective

Given $v = [a, b]^\top$, what's the derivative of

$$f(a,b) = log\left(\prod_i^n \frac{1}{b-a}\right) \tag{1}$$

$$= \sum_i^n log\left(\frac{1}{b-a}\right) \tag{2}$$

$$= -n\ log(b-a) \tag{3}$$

$$f'(v) = \begin{bmatrix} \frac{n}{b-a} \\ -\frac{n}{b-a} \end{bmatrix} \tag{4}$$

In Real Life

- In real life, if you need to take the derivative of a function.
- There isn't a magical solution book where you can verify your result.
- How do you know that your derivative is actually correct?
- **Solution:** you can use Python to verify it.
- On the right, I have written the Python code that automatically gives you the derivative function.
- Once you have the
  - auto-generated derivative function
  - your own derivative function
- You can plug random numbers into both and see if they are the same.

```python
#!/usr/bin/env python
# Automatically find the gradient of a function
# Download the package at : https://github.com/HIPS/autograd
import autograd.numpy as np
from autograd.numpy import log
from autograd.numpy import exp
from autograd import grad
```

Given the function

$$f(x) = \log_3(2x^2) - 2xe^{3x} + 2$$

The derivative should be

$$f'(x) = \frac{2}{x \ln 3} - 2e^{3x} - 6xe^{3x}$$

```python
def f(x):
    return log(2*x*x)/log(3) - 2*x*exp(3*x) + 2

def ∇f(x):
    return 2/(x*log(3)) - 2*exp(3*x) - 6*x*exp(3*x)

auto_grad = grad(f)  # Automatically obtain the gradient function

for i in range(10):
    x = np.random.randn()
    print('Auto ∇f : %.3f, Theoretical ∇f %.3f'%(auto_grad(x), ∇f(x)))
```

```
Auto ∇f : -4.946, Theoretical ∇f -4.946
Auto ∇f : -13.007, Theoretical ∇f -13.007
Auto ∇f : -6.007, Theoretical ∇f -6.007
Auto ∇f : -0.974, Theoretical ∇f -0.974
Auto ∇f : -2.505, Theoretical ∇f -2.505
Auto ∇f : -1.204, Theoretical ∇f -1.204
Auto ∇f : -1024.855, Theoretical ∇f -1024.855
Auto ∇f : -2.532, Theoretical ∇f -2.532
Auto ∇f : -239.588, Theoretical ∇f -239.588
Auto ∇f : -118.335, Theoretical ∇f -118.335
```

```
#!/usr/bin/env python
# Automatically find the gradient of a function
# Download the package at : https://pypi.org/project/autograd/
```

```
import autograd.numpy as np
from autograd import grad
```

```
n = 3
A = np.random.random((n,n))
w = np.random.random((n,1))
```

Given a function $(f(w) = Tr(w^\top Aw)$, we know that the is $f'(w) = (A + A^\top)w$

You cannot use A.dot(x) with autograd
It is a bug they are still trying to fix.
But should work in the future.

```
def f(w, A):
    return np.trace(np.dot(np.dot(np.transpose(w),A), w))
#
grad_foo = grad(f)        # Obtain its gradient function
#
print('Quadratic Function: Autogen Gradient : \n', grad_foo(w,A))
print('Quadratic Function: Theoretical Gradient : \n', np.dot((A+np.transpose(A)), w))
#
```

```
Quadratic Function: Autogen Gradient :
 [[0.66281681]
 [1.60622912]
 [0.9140561 ]]
Quadratic Function: Theoretical Gradient :
 [[0.66281681]
 [1.60622912]
 [0.9140561 ]]
```

This function automatically gives us the derivative function as grad_foo.

Define a function $f(w) = e^{-Tr(w^\top Aw)}$

```
def mult_gaussian(w, A):
    return np.exp(-np.trace(np.dot(np.dot(np.transpose(w),A), w)))
#
grad_foo = grad(mult_gaussian)       # Obtain its gradient function
print('Multi-var Gaussian: Autogen Gradient : \n', grad_foo(w,A))
print('Multi-var Gaussian: Theoretical Gradient : \n', -mult_gaussian(w,A)*(A+A.T).dot(w))
#
```

```
Multi-var Gaussian: Autogen Gradient :
 [[-0.34806185]
 [-0.84347148]
 [-0.47999395]]
Multi-var Gaussian: Theoretical Gradient :
 [[-0.34806185]
 [-0.84347148]
 [-0.47999395]]
```

You can programmatically check your own result

```
import autograd.numpy as np
from autograd import grad
```

```
def relu(w, x):
    v = np.dot(np.transpose(w),x)
    return np.maximum(0, v)

def grad_relu(w,x):
    if w.T.dot(x) > 0:
        return x
    else:
        return 0
for i in range(3):
    #    Initial setup
    x = np.random.randn(2,1)
    w = np.random.randn(2,1)

    grad_foo = grad(relu)        # Obtain its gradient function
    print('w.dot(x) = %.3f'%x.T.dot(w))
    print('Autogen Gradient : \n', grad_foo(w,x), '\n')
    print('Theoretical Gradient : \n', grad_relu(w,x), '\n\n')
```

$$f(w) = ReLU(w^T x),$$

```
w.dot(x) = 1.201
Autogen Gradient :
 [[-0.53292822]
 [ 0.70579309]]

Theoretical Gradient :
 [[-0.53292822]
 [ 0.70579309]]


w.dot(x) = -0.021
Autogen Gradient :
 [[0.]
 [0.]]

Theoretical Gradient :
 0


w.dot(x) = -0.139
Autogen Gradient :
 [[0.]
 [0.]]

Theoretical Gradient :
 0
```

Consolidate Today's knowledge

For each of the following function

1. Find the derivative for each function.
2. Use autograd to verify your solution

`https://pypi.org/project/autograd/`

1. Linear Function :

$$f(x) = y^\top x + x^\top \mathbb{1} \quad \text{where} \quad y = \begin{bmatrix} 2 \\ 5 \end{bmatrix}, \mathbb{1} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

2. Quadratic Function :

$$f(x) = x^\top A x \quad \text{where} \quad A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$$

3. Exponential Function :

$$f(x) = e^{x^\top A x} \quad \text{where} \quad A = \begin{bmatrix} 4 & 0 \\ 1 & 1 \end{bmatrix}$$

4. Logistic Regression Objective :

$$f(x) = \frac{1}{1 + e^{-w^\top x}} \quad \text{where} \quad w = \begin{bmatrix} 7 \\ 8 \end{bmatrix}$$