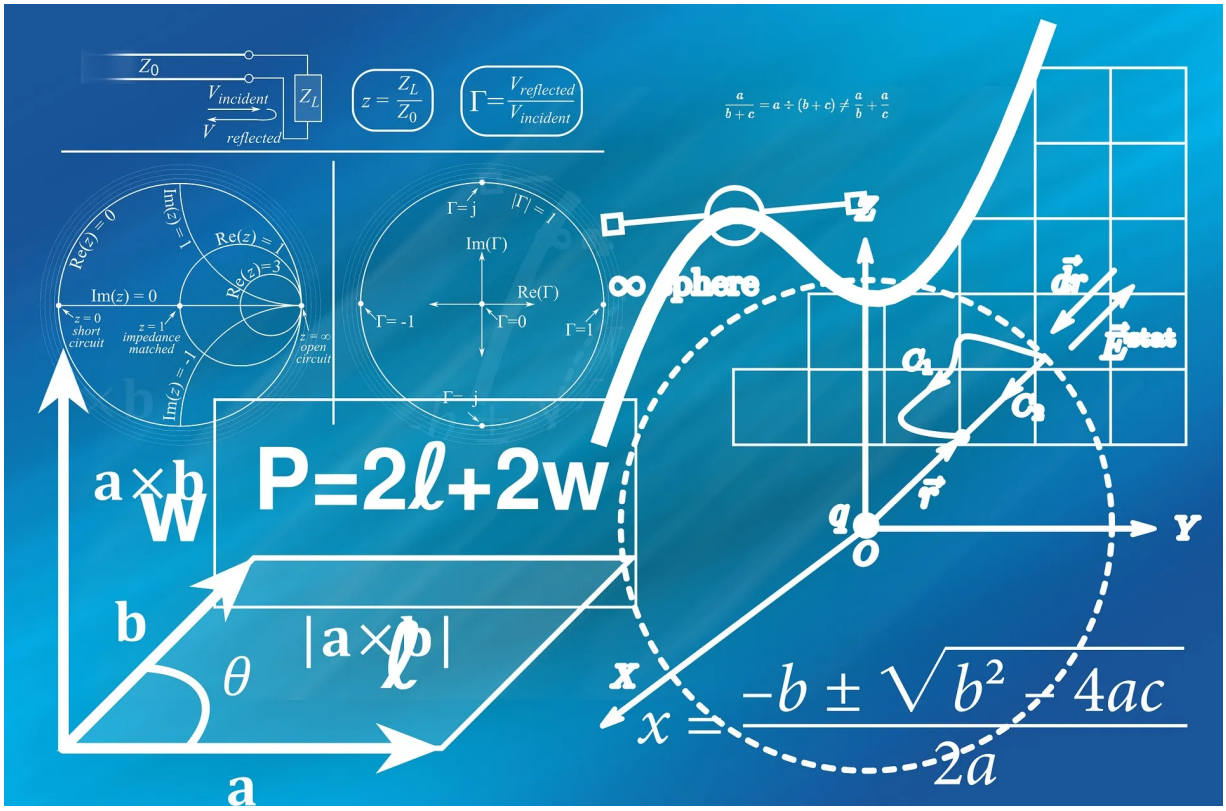
How Do You Find the Partial Derivative of a Function?

Part 2 of Step by Step: The Math Behind Neural Networks

Chi-Feng Wang · Follow

Published in Towards Data Science · 6 min read · Sep 1, 2018



Title image: Source

In Part 1, we have been given a problem: to calculate the gradient of this loss function:

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

Image 1: Loss function

Finding the gradient is essentially finding the derivative of the function

we can tweak (all the weights and biases), we have to find the derivatives with respect to each variable. This is known as the partial derivative, with the symbol ∂.

Partial Derivatives:

Computing the partial derivative of simple functions is easy: simply treat every other variable in the equation as a constant and find the usual scalar derivative. Here are some scalar derivative rules as a reminder:

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

Image 2: Scalar derivative rules // Source

Consider the partial derivative with respect to x (i.e. how y changes as x changes) in the function $f(x,y) = 3x^2y$. Treating y as a constant, we can find partial of x:

$$\frac{\partial}{\partial x} 3yx^2 = 3y \frac{\partial}{\partial x} x^2 = 3y2x = 6yx$$

Image 3: Partial with respect to x

Similarly, we can find the partial of y:

$$\frac{\partial}{\partial y} 3yx^2 = 3x^2 \frac{\partial}{\partial y} y = 3x^2 \times 1 = 3x^2$$

Image 4: Partial with respect to y

The gradient of the function $f(x,y) = 3x^2y$ is a horizontal vector, composed of the two partials:

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

Image 5: Gradient of f(x,y) // Source

This should be pretty clear: since the partial with respect to x is the gradient of the function in the x-direction, and the partial with respect to

y is the gradient of the function in the y-direction, the overall gradient is a vector composed of the two partials. This Khan Academy video offers a pretty neat graphical explanation of partial derivatives, if you want to visualize what we're doing.

Chain Rules:

For simple functions like $f(x,y) = 3x^2y$, that is all we need to know. However, if we want to compute partial derivatives of more complicated functions — such as those with nested expressions like $max(0, \mathbf{w}\cdot\mathbf{X}+b)$ — we need to be able to utilize the multivariate chain rule, known as the *single variable total-derivative chain rule* in the paper.

Single Variable Chain Rule

Let's first review the single variable chain rule. Consider the function

$y=f(g(x))=sin(x^2)$. To get the derivative of this expression, we multiply the derivative of the outer expression with the derivative of the inner expression or 'chain the pieces together'. In other words:

Image 6: Single-variable chain rule where u is the intermediate variable for nested subexpressions

For our example, $u=x^2$ and $y=sin(u)$. Hence:

Image 7: Derivatives // Source

and

Image 8: Derivative of the whole expression // Source

It's nice to think about the single-variable chain rule as a diagram of operations that x goes through, like so:

Image 9: Diagram of chain of operations for $y=\sin(x^2)$

This concept of visualizing equations as diagrams will come in extremely handy when dealing with the multivariable chain rule. Also, if you use Tensorflow (or Keras) and TensorBoard, as you build your model and write your training code, you can see a diagram of operations similar to this.

Multivariable Chain Rule

The multivariable chain rule, also known as the *single-variable total-derivative chain rule*, as called in the paper, is a variant of the scalar chain rule. Unlike what its name suggests, it can be applied to expressions with only a single variable. However, the expression should have multiple intermediate variables.

To illustrate this point, let us consider the equation $y=f(x)=x+x^2$. Using the scalar additional derivative rule, we can immediately calculate the derivative:

Image 10: Derivative of $x+x^2$

Let's try doing it with the chain rule. First, we introduce intermediate variables: $u_1(x) = x^2$ and $u_2(x, u_1) = x + u_1$. If we apply the single-variable chain rule, we get:

Image 11: Using the single-variable chain rule

Obviously, $2x \neq 1+2x$, so something is wrong here. Let's draw out the graph of our equation:

Image 12: Diagram of chain of operations for $y = x+x^2$ // // Source

The diagram in Image 12 is no longer linear, so we have to consider *all* the pathways in the diagram that lead to the final result. Since u_2 has two parameters, partial derivatives come into play. To calculate the derivative of this function, we have to calculate partial derivative with respect to x of $u_2(x, u_1)$. Here, a change in x is reflected in u_2 in two ways: as an

operand of the addition and as an operand of the square operator. In symbols, $\hat{y} = (x+\Delta x)+(x+\Delta x)^2$ and $\Delta y = \hat{y}-y$ and where $\hat{y}$ is the y-value at a tweaked x.

Hence, to computer the partial of $u_2(x, u_1)$, we need to sum up all possible contributions from changes in x to the change in y. The total derivative of $u_2(x, u_1)$ is given by:

Image 13: Derivative of $y = x+x^2$ // Source

In simpler terms, you **add up** the effect of a change in x directly to u_2 and the effect of a change in x through u_1 to u_2. I find it easier to visualize it through a graph:

Image 14: Graph of $y = x+x^2$, with partials included

And that's it! We got the correct answer: 1+2x. We can now sum that process up in a single rule, the multivariable chain rule (or the single-variable total-derivative chain rule):

Image 15: Multivariable chain rule // Source

If we introduce an alias for x as x=u(n+1), then we can rewrite that formula into its final form, which look slightly neater:

Image 16: Multivariable chain rule // Source

That's all to it! To review, let's do another example: $f(x)=sin(x+x^2)$. Our 3 intermediate variables are: $u_1(x) = x^2$, $u_2(x, u_1)=x+u_1$, and $u_3(u_2) = sin(u_2)$. Once again, we can draw our graph:

Image 17: Graph of $y = \sin(x+x^2)$, with partials included

and calculate our partials:

Image 18: Partials for the function $y = \sin(x+x^2)$

Therefore, the derivative of $f(x)=sin(x+x^2)$ is $cos(x+x^2)(1+2x)$.

How does this relate back to our problem? Remember, we need to find the partial derivative of our loss function with respect to both **w** (the vector of all our weights) and *b* (the bias). However, our loss function is not that simple — there are multiple nested subexpressions (i.e. multiple intermediate variables) which will require us to use the chain rule.

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

Image 19: Loss function

There's one more problem left. As you can see, our loss function doesn't just take in scalars as inputs, it takes in vectors as well. How can we compute the partial derivatives of vector equations, and what does a vector chain rule look like?

Check out Part 3 to find out!

If you haven't already, click here to read Part 1!

Jump ahead to other articles:

- Part 3: Vector Calculus
- Part 4: Putting It All Together

Download the original paper here.

If you like this article, don't forget to leave some claps! Do leave a comment below if you have any questions or suggestions :)

Neural Networks | Artificial Intelligence | Calculus | Derivatives | Loss Function

Written by Chi-Feng Wang

Follow

1.5K Followers · Writer for Towards Data Science

Student at UC Berkeley; Machine Learning Enthusiast

More from Chi-Feng Wang and Towards Data Science

Chi-Feng Wang in Towards Data Science

The Vanishing Gradient Problem

The Problem, Its Causes, Its Significance, and Its Solutions

3 min read · Jan 8, 2019

2.7K 10

Jacob Marks, Ph.D. in Towards Data Science

How I Turned My Company's Docs into a Searchable Database with...

And how you can do the same with your docs

15 min read · Apr 25

3.1K 40

Leonie Monigatti in Towards Data Science

Getting Started with LangChain: A Beginner's Guide to Building...

A LangChain tutorial to build anything with large language models in Python

· 12 min read · Apr 25

2.2K 18

Chi-Feng Wang in Towards Data Science

A Basic Introduction to Separable Convolutions

Explaining spatial separable convolutions, depthwise separable convolutions, and th...

8 min read · Aug 14, 2018

7.1K 44

See all from Chi-Feng Wang

See all from Towards Data Science

Recommended from Medium

Peter Kar... in Artificial Intelligence in Plain Eng...

Linear Regression in depth

The directive equation of a straight line, simple linear regression, math, cost...

· 6 min read · Jan 27

111 2

Peter Kar... in Artificial Intelligence in Plain Eng...

L1 (Lasso) and L2 (Ridge) regularizations in logistic...

Logistic regression , Lasso and Rigde regularizations, derivations, math

· 6 min read · Feb 3

38 1

Lists

What is ChatGPT?

9 stories · 64 saves

Staff Picks

330 stories · 83 saves

Alexander Nguyen in Level Up Coding

Why I Keep Failing Candidates During Google Interviews...

They don't meet the bar.

· 4 min read · Apr 13

4.2K 127

Steins

Diffusion Model Clearly Explained!

How does AI artwork work? Understanding the tech behind the rise of AI-generated...

· 7 min read · Dec 26, 2022

1.1K 3

Sayef

Logistic Regression with Gradient Descent and...

Learn how to implement logistic regression with gradient descent optimization from...

· 13 min read · Apr 10

94

Ester Hlav in Towards Data Science

Kaiming He Initialization in Neural Networks—Math Proof

Deriving optimal initial variance of weight matrices in neural network layers with ReL...

· 10 min read · Feb 15

136 3

See more recommendations