CS3105 Machine Learning - Appendix 2: Derivatives

Nguyen Dang

This appendix is useful for all the three Machine Learning lectures 2, 3 and 4. In the first part of the appendix, we will review the basics of calculus, including the basic derivative formulas and the chain rule of calculus. In the second part, a cheat-sheet of derivative formulas for all functions used the lectures is provided.

1 Basic calculus

1.1 Basic formulas

We will use the following notations in all formulas:

- k: a constant with respect to x,
- x: a variable,
- f, g, p: functions of variable x, i.e., we should have written them as $f(x)$, $g(x)$, $p(x)$, but here I omit the variable x for brevity.

Below is a list of basic formulas for calculating derivative of a function:

$\frac{\partial k}{\partial x} = 0$	$\frac{\partial x}{\partial x} = 1$	$\frac{\partial kx}{\partial x} = k$	$\frac{\partial kf}{\partial x} = k\frac{\partial f}{\partial x}$
$\frac{\partial (f+g)}{\partial x} = \frac{\partial f}{\partial x} + \frac{\partial g}{\partial x}$	$\frac{\partial (f-g)}{\partial x} = \frac{\partial f}{\partial x} - \frac{\partial g}{\partial x}$	$\frac{\partial (fg)}{\partial x} = f\frac{\partial g}{\partial x} + g\frac{\partial f}{\partial x}$	
$\frac{\partial x^k}{\partial x} = kx^{k-1}$	$\frac{\partial f^k}{\partial x} = kf^{k-1}\frac{\partial f}{\partial x}$	$\frac{e^x}{\partial x} = e^x$	$\frac{\partial e^f}{\partial x} = e^f\frac{\partial f}{\partial x}$
$\frac{\ln x}{\partial x} = \frac{1}{x}$	$\frac{\ln f}{\partial x} = \frac{1}{f}\frac{\partial f}{\partial x}$	$\frac{\log_k x}{\partial x} = \frac{1}{x \ln k}$	$\frac{\log_k f}{\partial x} = \frac{1}{f \ln k}\frac{\partial f}{\partial x}$

Exercise: Now here is a small exercise for you. Let's calculate $\frac{\partial (f/g)}{\partial x}$ (solution is at the end of this document). Hint: you can rewrite it as:

$$\frac{\partial (f/g)}{\partial x} = \frac{\partial (fg^{-1})}{\partial x} \tag{1}$$

and then apply the derivatives formulas for multiplication and power function above.

1.2 The chain rule

The chain rule of calculus is used intensively in our lectures for both Logistic Regression and Neural Networks, so it is important to know this rule.

The simple version of the rule is as follows. Given $g(.)$ as a function of x, and $f(.)$ a function of g, we have:

$$\frac{\partial f(g(x))}{\partial x} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial x} \tag{2}$$

Similarly, given $p(.)$ as a function of x, $g(.)$ as a function of p, and $f(.)$ as a function of g, then we have:

$$\frac{\partial f(g(p(x)))}{\partial x} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial p}\frac{\partial p}{\partial x} \tag{3}$$

There is a generalisation of the chain rule when we have multiple functions $g_1(.)$, $g_2(.)$, ..., $g_k(.)$, each of which is a function of x, and a **multivariate function** $f(g_1(x), g_2(x), ..., g_k(x))$. In such case, the chain rule is written as:

$$\frac{\partial f}{\partial x} = \sum_{i=1}^{k} \frac{\partial f}{\partial g_i}\frac{\partial g_i}{\partial x} \tag{4}$$

This is called the **multivariate chain rule**. This rule is useful when we calculate the partial derivatives of the loss function when Softmax activation function is used (for multiclass classification neural networks).

Exercise: Let's look at the table in section 1.1 and see how some formulas with function f were derived directly from its simpler version (without f) using the chain rule. For example, $\frac{\partial kf}{\partial x}$ can be calculated using $\frac{\partial kx}{\partial x}$ and the chain rule:

$$\frac{\partial kf}{\partial x} = \frac{\partial kf}{\partial f}\frac{\partial f}{\partial x} = k\frac{\partial f}{\partial x} \tag{5}$$

Now you can try to do the same for some other formulas in that table.

2 Derivative Cheat-sheet

Linear functions:

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} = w_n x_n + w_{n-1}x_{n-1} + ... + w_1 x_1 + w_0 \tag{6}$$

$$\frac{\partial f}{\partial w_k} = \begin{cases} x_k & \text{if } 1 \leq k \leq n \\ 1 & \text{if } k = 0 \end{cases} \tag{7}$$

Logistic/Sigmoid activation function:

$$g(z) = \frac{1}{1+e^x} \tag{8}$$

$$\frac{\partial g}{\partial z} = g(1-g) \tag{9}$$

Cross Entropy loss function for binary classification: on one training example $(\mathbf{x}, y)$

$$L(\hat{y}) = -y \log_2 \hat{y} - (1-y) \log_2 (1-\hat{y}) \tag{10}$$

where $\hat{y}$ is the prediction value for $\mathbf{x}$.

$$\frac{\partial L}{\partial \hat{y}} = \frac{\hat{y} - y}{\hat{y}(1-\hat{y})} \tag{11}$$

With the three partial derivate formulas above, you can calculate $\frac{\partial L}{\partial w_k}$ for Logistic Regression using the chain rule.

Cross Entropy loss function for multiclass classification: on one training example $(\mathbf{x}, \mathbf{y})$ with k classes

$$L(\hat{y}) = -\sum_{i=1}^{k} y_i \ln \hat{y}_i \tag{12}$$

where $\hat{\mathbf{y}}$ is the vector of prediction values for $\mathbf{x}$.

$$\frac{\partial L}{\partial \hat{y}} = -\frac{y_i}{\hat{y}_i} \tag{13}$$

Note: you may wonder why I use $\ln \hat{y}_i$ in the Cross Entropy loss function formula instead of $\log_2 \hat{y}_i$ (as in the lectures). It is actually not important which $\log$ base is used. We can use $\ln$, $\log_{10}$, $\log_2$ or whatever $\log$ based you fancy. Here I use $\ln$ so that the derivative formula is simpler. If we use $\log_2$, it will be: $\frac{y_i}{\hat{y}_i \ln 2}$.

TanH activation function:

$$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{14}$$

$$\frac{\partial g}{\partial z} = 1 - g^2 \tag{15}$$

ReLU activation function:

$$g(z) = max(0, z) \tag{16}$$

$$\frac{\partial g}{\partial z} = \begin{cases} 0 & \text{if } z \leq 0 \\ 1 & \text{if } z > 0 \end{cases} \tag{17}$$

Softmax activation function:

Recall that the Softmax activation function is used for the output layer of a neural network for multiclass classification. The number of nodes in the output layer is equal to the number of classes k. Let's call z_1, z_2, ..., z_k results of the linear function on the output nodes (before Softmax is applied). The Softmax activation function for output node i is defined as:

$$g(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{18}$$

$$\frac{\partial g(z_j)}{\partial z_i} = g(z_i)(\delta_{ij} - g(z_j)) \tag{19}$$

where

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{20}$$

Now if we want to calculate $\frac{\partial L}{\partial z_i}$ (L is a loss function for multiclass classification), we will need to apply the **multivariate** chain rule because L is a function of all variables $g(z_1)$, $g(z_2)$, ..., $g(z_k)$.

$$\frac{\partial L}{\partial z_i} = \sum_{j=1}^{k} \frac{\partial L}{\partial g(z_j)} \frac{\partial g(z_j)}{\partial z_i} \tag{21}$$

Solution for the exercise in section 1.1:

$$\frac{\partial(f/g)}{\partial x} = \frac{1}{g}\frac{\partial f}{\partial x} - \frac{f}{g^2}\frac{\partial g}{\partial x} \tag{22}$$