

来写一个 softmax 求导的推导过程，不仅可以给自己理清思路，还可以造福大众，岂不美哉~

softmax 经常被添加在分类任务的神经网络中的输出层，神经网络的反向传播中关键的步骤就是求导，从这个过程也可以更深刻地理解反向传播的过程，还可以对梯度传播的问题有更多的思考。

softmax 函数

softmax(柔性最大值)函数，一般在神经网络中， softmax 可以作为分类任务的输出层。其实可以认为 softmax 输出的是几个类别选择的概率，比如我有一个分类任务，要分为三个类，softmax 函数可以根据它们相对的大小，输出三个类别选取的概率，并且概率和为 1。

softmax 函数的公式是这种形式：

$$S_i = \frac{e^{z_i}}{\sum_k e^{z_k}}$$

S_i 代表的是第 i 个神经元的输出。

ok，其实就是在输出后面套一个这个函数，在推导之前，我们统一一下网络中的各个表示符号，避免后面突然出现一个什么符号懵逼推导不下去了。

神经元的输出设为：

$$z_i = \sum_j w_{ij} x_{ij} + b$$

其中 w_{ij} 是第 i 个神经元的第 j 个权重， b 是偏移值。 z_i 表示该网络的第 i 个输出。

给这个输出加上一个 softmax 函数，那就变成了这样：

$$a_i = \frac{e^{z_i}}{\sum_k e^{z_k}}$$

a_i 代表 softmax 的第 i 个输出值，右侧就是套用了 softmax 函数。

损失函数 loss function

在神经网络反向传播中，要求一个损失函数，这个损失函数其实表示的是真实值与网络的估计值的误差，知道误差了，才能知道怎样去修改网络中的权重。

损失函数可以有很多形式，这里用的是交叉熵函数，主要是由于这个求导结果比较简单，易于计算，并且交叉熵解决某些损失函数学习缓慢的问题。交叉熵的函数是这样的：

$$C = - \sum_i y_i \ln a_i$$

其中 y_i 表示真实的分类结果。

到这里可能嵌套了好几层，不过不要担心，下面会一步步推导，强烈推荐在纸上写一写，有时候光看着看着就迷糊了，自己边看边推导更有利于理解~

最后的准备

在我最开始看 softmax 推导的时候，有时候看到一半不知道是怎么推出来的，其实主要是因为一些求导法则忘记了，唉~

所以这里把基础的求导法则和公式贴出来~有些忘记的朋友可以先大概看一下：

基本初等函数求导公式

$$(1) \quad (C)' = 0$$

$$(3) \quad (\sin x)' = \cos x$$

$$(5) \quad (\tan x)' = \sec^2 x$$

$$(7) \quad (\sec x)' = \sec x \tan x$$

$$(9) \quad (a^x)' = a^x \ln a$$

$$(11) \quad (\log_a x)' = \frac{1}{x \ln a}$$

$$(13) \quad (\arcsin x)' = \frac{1}{\sqrt{1-x^2}}$$

$$(15) \quad (\arctan x)' = \frac{1}{1+x^2}$$

$$(2) \quad (x^\mu)' = \mu x^{\mu-1}$$

$$(4) \quad (\cos x)' = -\sin x$$

$$(6) \quad (\cot x)' = -\csc^2 x$$

$$(8) \quad (\csc x)' = -\csc x \cot x$$

$$(10) \quad (e^x)' = e^x$$

$$(12) \quad (\ln x)' = \frac{1}{x}$$

$$(14) \quad (\arccos x)' = -\frac{1}{\sqrt{1-x^2}}$$

$$(16) \quad (\operatorname{arccot} x)' = -\frac{1}{1+x^2}$$

函数的和、差、积、商的求导法则

设 $u = u(x)$, $v = v(x)$ 都可导, 则

$$(1) \quad (u \pm v)' = u' \pm v'$$

$$(3) \quad (uv)' = u'v + uv'$$

$$(2) \quad (Cu)' = Cu' \quad (C \text{ 是常数})$$

$$(4) \quad \left(\frac{u}{v}\right)' = \frac{u'v - uv'}{v^2}$$

推导过程

好了, 这下正式开始~

首先, 我们要明确一下我们要求什么, 我们要求的是我们的 loss 对于神经元输出 (z_i) 的梯度, 即:

$$\frac{\partial C}{\partial z_i}$$

根据复合函数求导法则:

复合函数求导法则

设 $y = f(u)$ ，而 $u = \varphi(x)$ 且 $f(u)$ 及 $\varphi(x)$ 都可导，则复合函数 $y = f[\varphi(x)]$ 的导数为

$$\frac{dy}{dx} = \frac{dy}{du} \cdot \frac{du}{dx} \text{ 或 } y' = f'(u) \cdot \varphi'(x)$$

有个人可能有疑问了，这里为什么是 a_j 而不是 a_i ，这里要看一下 softmax 的公式了，因为 softmax 公式的特性，它的分母包含了所有神经元的输出，所以，对于不等于 i 的其他输出里面，也包含着 z_i ，所有的 a 都要纳入到计算范围中，并且后面的计算可以看到需要分为 $i = j$ 和 $i \neq j$ 两种情况求导。下面我们一个一个推：

$$\frac{\partial C}{\partial a_j} = \frac{\partial(-\sum_j y_j \ln a_j)}{\partial a_j} = -\sum_j y_j \frac{1}{a_j}$$

第二个稍微复杂一点，我们先把它分为两种情况：

①如果 $i = j$ ：

$$\frac{\partial a_i}{\partial z_i} = \frac{\partial(\frac{e^{z_i}}{\sum_k e^{z_k}})}{\partial z_i} = \frac{\sum_k e^{z_k} e^{z_i} - (e^{z_i})^2}{\sum_k (e^{z_k})^2} = (\frac{e^{z_i}}{\sum_k e^{z_k}})(1 - \frac{e^{z_i}}{\sum_k e^{z_k}}) = a_i(1 - a_i)$$

②如果 $i \neq j$ ：

$$\frac{\partial a_j}{\partial z_i} = \frac{\partial(\frac{e^{z_j}}{\sum_k e^{z_k}})}{\partial z_i} = -e^{z_j}(\frac{1}{\sum_k e^{z_k}})e^{z_i} = -a_i a_j$$

ok，接下来我们只需要把上面的组合起来：

$$\frac{\partial C}{\partial z_i} = (-\sum_j y_j \frac{1}{a_j}) \frac{\partial a_j}{\partial z_i} = -\frac{y_i}{a_i} a_i(1 - a_i) + \sum_{j \neq i} \frac{y_j}{a_j} a_i a_j = -y_i + y_i a_i + \sum_{j \neq i} y_j a_i = -y_i + a_i \sum_j y_j$$

最后的结果看起来简单了很多，最后，针对分类问题，我们给定的结果 y_i 最终只会有一个类别是 1，其他类别都是 0，因此，对于分类问题，这个梯度等于：

$$\frac{\partial C}{\partial z_i} = a_i - y_i$$

作者：bakaqian

链接：<https://www.jianshu.com/p/c02a1fbffad6>

来源：简书

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。