**反向传播算法——多类别分类Softmax**

**Softmax与多类别分类**

Softmax函数的基本思想是为多个类别给出一个概率分布，这些概率之和为1。比如，对于MNIST数据集，每个样本是一个图片，总共有60 000个样本，样本的类别有10个，即0-9的10个数字。在神经网络的最后输出层(10个神经元)之后，我们需要进行Softmax函数转换，把10个神经元的输出，转换成10个类别的概率分布。

Softmax的函数的具体形式如下：

，这里c为类别的数量，为对应各个类别的原始数值。参考图1，实际上是神经网络最后一层的激活值；可以看出。

利用神经网络和Softmax函数进行多类别分类，如图1所示。

Softmax🡪

=

… … …

=

图1. Softmax函数示意图

**计算Softmax的导数**

针对多类别分类，采用交叉熵损失函数，具体形式如下。

现在计算Softmax的导数。我们知道当的时候，。

(1)在这里，=。

(2)于是当

===)=(1-)，因为

(3)当

=

**多类别分类的损失函数和导数推导**

=

=

=

=

-=

=

=

=

(因为

于是有，到此，我们看到softmax的导数形式是很简单的，即。

**带Softmax的神经网络的反向传播过程[[1]](#footnote-1)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 输入 | 1层 | 2层 | … | L层 | Softmax | 代价函数 |
| *即* |  |  | … |  | Softmax  即  即预测值 | 为的各个分量，注意即预测值  y为实际值，其分量取值为0或者1 |

由上述推导有

|  |
| --- |
| 这是多类别分类和2值分类的MLP的反向传播的区别之处，其他步骤一致 |

根据上一个文档“2021-new-反向传播算法-MLP二值分类.docx”，有

=

for l=L-1 to 1

=

比如对于两层的MLP，有

=

因为== 。

类似地，我们得到，具体如下。

= =

和按照文档“2021-new-反向传播算法-MLP二值分类.docx”已经展示的原理，类推即可。

1. http://www.adeveloperdiary.com/data-science/deep-learning/neural-network-with-softmax-in-python/. [↑](#footnote-ref-1)