# 手打例子一步一步带你看懂softmax函数以及相关求导过程

[](https://www.jianshu.com/u/97ed7fbbd56c)

[忆臻HIT\_NLP](https://www.jianshu.com/u/97ed7fbbd56c)

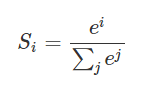
2017.03.16 00:20\* 字数 2210 阅读 14961评论 23喜欢 43

这几天学习了一下softmax激活函数，以及它的梯度求导过程，整理一下便于分享和交流！

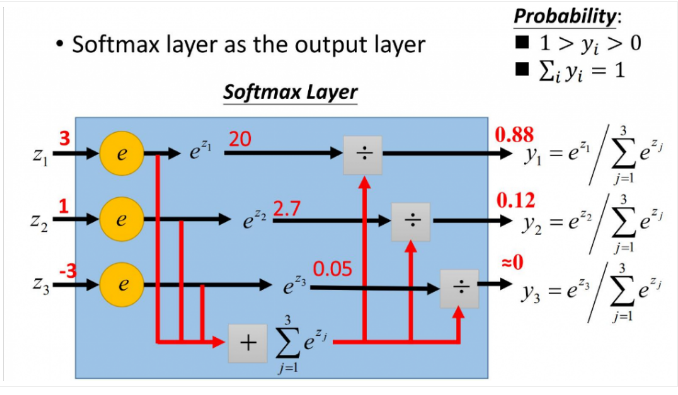
# **softmax函数**

**softmax用于多分类过程中**，它将多个神经元的输出，映射到（0,1）区间内，可以看成概率来理解，从而来进行多分类！

假设我们有一个数组，V，Vi表示V中的第i个元素，那么这个元素的softmax值就是



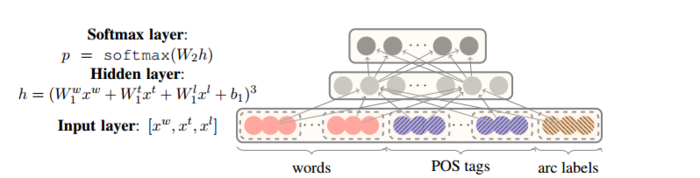
更形象的如下图表示：



s**oftmax直白来说就是将原来输出是3,1,-3通过softmax函数一作用，就映射成为(0,1)的值，而这些值的累和为1（满足概率的性质），那么我们就可以将它理解成概率，在最后选取输出结点的时候，我们就可以选取概率最大（也就是值对应最大的）结点，作为我们的预测目标！**

**举一个我最近碰到利用softmax的例子：**我现在要实现基于神经网络的句法分析器。用到是基于转移系统来做，那么神经网络的用途就是帮我预测我这一个状态将要进行的动作是什么？比如有10个输出神经元，那么就有10个动作，1动作，2动作，3动作...一直到10动作。（**这里涉及到nlp的知识，大家不用管，只要知道我现在根据每个状态（输入），来预测动作（得到概率最大的输出），最终得到的一系列动作序列就可以完成我的任务即可**）

原理图如下图所示：



那么比如在一次的输出过程中输出结点的值是如下：

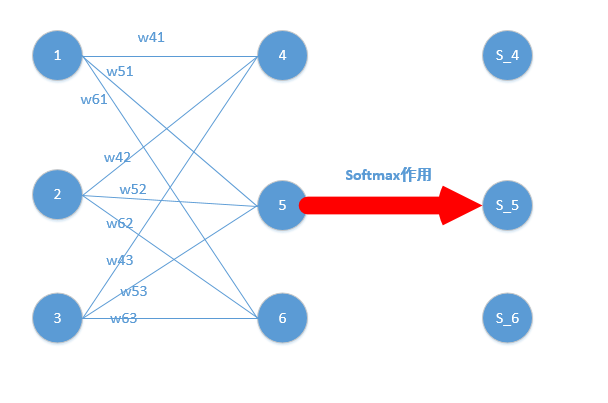
[0.2,0.1,0.05,0.1,0.2,0.02,0.08,0.01,0.01,0.23]

**那么我们就知道这次我选取的动作是动作10，因为0.23是这次概率最大的，那么怎么理解多分类呢？很容易，如果你想选取俩个动作，那么就找概率最大的俩个值即可~（这里只是简单的告诉大家softmax在实际问题中一般怎么应用）**

# **softmax相关求导**

当我们对分类的Loss进行改进的时候，我们要通过梯度下降，每次优化一个step大小的梯度，**这个时候我们就要求Loss对每个权重矩阵的偏导，然后应用链式法则**。那么这个过程的第一步，就是对softmax求导传回去，不用着急，我后面会举例子非常详细的说明。在这个过程中，你会发现用了softmax函数之后，**梯度求导过程非常非常方便！**

**下面我们举出一个简单例子，原理一样，目的是为了帮助大家容易理解！**



**我们能得到下面公式：**

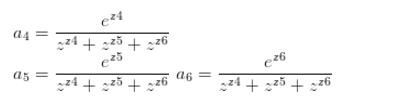
z4 = w41\*o1+w42\*o2+w43\*o3

z5 = w51\*o1+w52\*o2+w53\*o3

z6 = w61\*o1+w62\*o2+w63\*o3

z4,z5,z6分别代表结点4,5,6的输出，01,02,03代表是结点1,2,3往后传的输入.

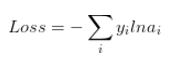
那么我们可以经过softmax函数得到



**好了，我们的重头戏来了，怎么根据求梯度，然后利用梯度下降方法更新梯度！**

要使用梯度下降，肯定需要一个损失函数，**这里我们使用交叉熵作为我们的损失函数，为什么使用交叉熵损失函数，不是这篇文章重点，**后面有时间会单独写一下为什么要用到交叉熵函数（这里我们默认选取它作为损失函数）

交叉熵函数形式如下：



其中y代表我们的真实值，a代表我们softmax求出的值。i代表的是输出结点的标号！在上面例子，i就可以取值为4,5,6三个结点（**当然我这里只是为了简单，真实应用中可能有很多结点**）

**现在看起来是不是感觉复杂了，居然还有累和，然后还要求导，每一个a都是softmax之后的形式！**

**但是实际上不是这样的，我们往往在真实中，如果只预测一个结果，那么在目标中只有一个结点的值为1，比如我认为在该状态下，我想要输出的是第四个动作（第四个结点）,那么训练数据的输出就是a4 = 1,a5=0,a6=0，哎呀，这太好了，除了一个为1，其它都是0，那么所谓的求和符合，就是一个幌子，我可以去掉啦！**

**为了形式化说明，我这里认为训练数据的真实输出为第j个为1，其它均为0！**

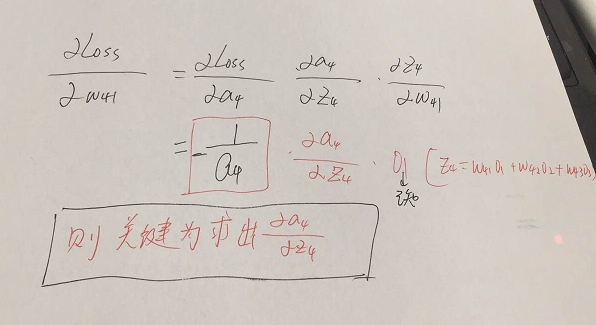
**那么Loss就变成了Loss = -yilnaj(yi,aj的i，j均为下标，公式不好打)**,累和已经去掉了，太好了。现在我们要开始求导数了！

我们在整理一下上面公式，为了更加明白的看出相关变量的关系：

其中yj=1,,那么形式变为Loss = -lnaj

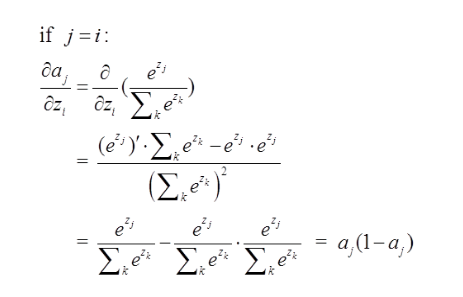
那么形式越来越简单了，求导分析如下：

**参数的形式在该例子中，总共分为w41,w42,w43,w51,w52,w53,w61,w62,w63.这些，那么比如我要求出w41,w42,w43的偏导，就需要将Loss函数求偏导传到结点4，然后再利用链式法则继续求导即可，举个例子此时求w41的偏导为:**



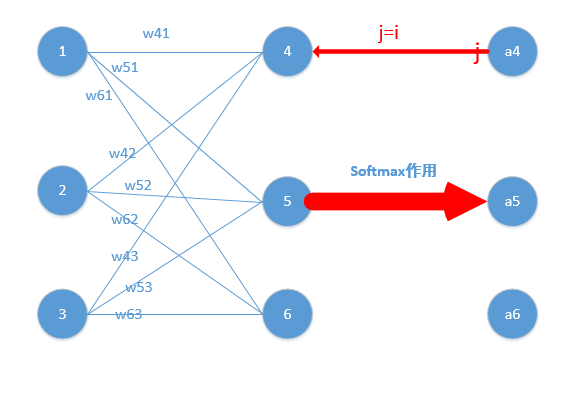
**w51.....w63等参数的偏导同理可以求出，那么我们的关键就在于Loss函数对于结点4,5,6的偏导怎么求，如下：**

**这里分为俩种情况：**



j=i对应例子里就是如下图所示：

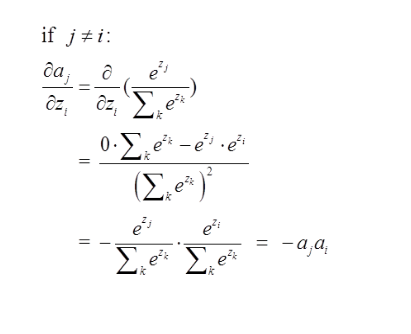
比如我选定了j为4，那么就是说我现在求导传到4结点这！



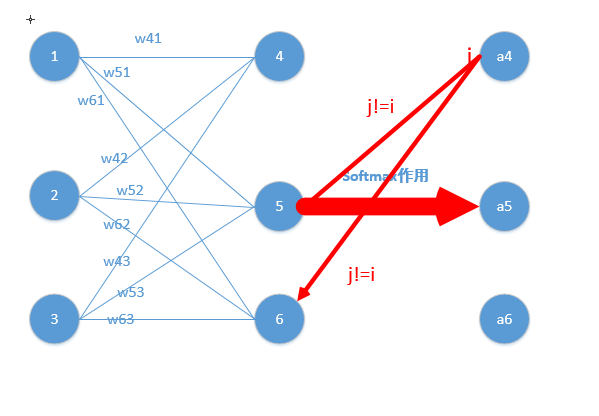
那么由上面求导结果再乘以交叉熵损失函数求导

Loss = -lnaj,它的导数为-1/aj,与上面的aj(1-aj)相乘为aj-1（**形式非常简单，这说明我只要正向求一次得出结果，然后反向传梯度的时候，只需要将它结果减1即可，后面还会举例子！**）那么我们可以得到Loss对于4结点的偏导就求出了了（**这里假定4是我们的预计输出**）

第二种情况为：



这里对应我的例子图如下，我这时对的是j不等于i，往前传：



那么由上面求导结果再乘以交叉熵损失函数求导

Loss = -lnaj,它的导数是-1/aj,与上面-ajai相乘为ai（**形式非常简单，这说明我只要正向求一次得出结果，然后反向传梯度的时候，只需要将它结果保存即可，后续例子会讲到**）**这里就求出了除4之外的其它所有结点的偏导，然后利用链式法则继续传递过去即可！我们的问题也就解决了！**

# **下面我举个例子来说明为什么计算会比较方便，给大家一个直观的理解**

**举个例子，通过若干层的计算，最后得到的某个训练样本的向量的分数是[ 2, 3, 4 ],**

那么经过softmax函数作用后概率分别就是=[e^2/(e^2+e^3+e^4),e^3/(e^2+e^3+e^4),e^4/(e^2+e^3+e^4)] = [0.0903,0.2447,0.665],如果这个样本正确的分类是第二个的话，那么计算出来的偏导就是[0.0903,0.2447-1,0.665]=[0.0903,-0.7553,0.665]，是不是非常简单！！然后再根据这个进行back propagation就可以了

**到这里，这篇文章的内容就讲完了，我希望根据自己的理解，通过列出大量例子，直白的给大家讲解softmax的相关内容，让大家少走弯路，真心希望对大家的理解有帮助！欢迎交流指错！画图整理不易，觉得有帮助的给个赞呗，哈哈！**

**参考：**

[softmax的log似然代价函数（公式求导）](https://link.jianshu.com?t=http://link.zhihu.com/?target=http://m.blog.csdn.net/article/details?id=51045303" \t "/home/nan/Documents\\x/_blank)

[交叉熵代价函数（作用及公式推导）](https://link.jianshu.com?t=http://link.zhihu.com/?target=http://m.blog.csdn.net/article/details?id=51043064" \t "/home/nan/Documents\\x/_blank)

[Softmax回归 - Ufldl](https://link.jianshu.com?t=http://link.zhihu.com/?target=http://deeplearning.stanford.edu/wiki/index.php/Softmax%E5%9B%9E%E5%BD%92" \t "/home/nan/Documents\\x/_blank)

部分图片来自于网络！