梯度推导总结：

严格来说，矩阵之间的求导是不存在链式求导法则的。因为，对于一个一般的的映射，你并不知道它内部映射机制是什么样的，举个栗子，CNN中的池化层与BP中的隐含层的映射机制是不同的，针对不同问题时f具有不同的形式。所以，笔者认为，不能像简单的标量函数求导一样，对矩阵求导的也给出一个统一的链式法则，针对不同的问题，我们应该依据f的具体形式，给出一个求导得准则。在这里，对于一个类似于BP网络隐含层上的映射,通过归纳总结，可以给出一个链式求导法则，现叙述如下（注意到）

定义分别为第l层和第l+1层神经网络的输入和输出值，它们直接的关系满足下式：

其中，是第l层到第l+1层的权重矩阵，各向量之间的联系如下：

在反向传播过程中，第l+1层的误差为则，传递到第l层误差为

现在对反向传播中使用的链式法则做详细描述，对于m维向量对n向量的导数，是一个的矩阵，在该链式法则中，每向下传播一次，应该将更深一层的导数放在上一层导数左侧，比如将放在的左侧，通过这种链式相乘，可以满足矩阵的运算。在求解上述时，需要求三次导数，其中，在求时，因为一般常用的激活函数是，即的值仅与有关。所以的结果应该是一个的对角阵，除对角线外，其它元素均为0，这时，也可以将这个对角阵的对角线抽取出来形成一个n维的列向量，通过与后两个导数做运算。

注意这里的链式法则仅仅适用于列向量求梯度时进行传递，如需要对矩阵进行求导，如对求梯度，还需添加额为的规则，这里不再赘述。实际上，有了的梯度后，再求是一个并不复杂的过程。

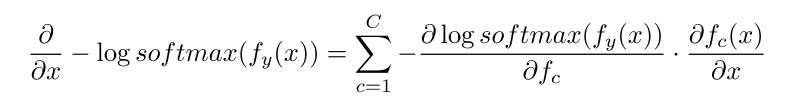
常见的类似于BP网络结构的网络有RNN及RNN相关的各种扩展。

本质上来说，不论是矩阵对矩阵之间的求导，还是向量之间的求导，都是多个因变量变量对多个自变量之间的导数，之所以将其写成矩阵的方式，是想找到一种更为方便的表示方法，否则，手工单独的对每一个自变量进行求导，将是一个非常庞大的工程。

额外补充一个交叉熵对softmax函数的梯度：

1. 梯度推导

应用链式求导法则有：



注意，当你试着对向量f的一个元素求偏导数时，尝试一下你能否求出包含f每一维偏导数的梯度向量

我们知道，标量对一个向量求导，就是用这个标量对向量的每一维度求导，所以，对f求导时，先对f的第y个维度(x)求导得

再对f的其他维度求导得

综上所述

所以对x的导数值