一.重点问题

1. 符号定义

：训练样本集，N是样本大小。

：模型输入，一个d维的列向量，可以是单独的一个词，或者上下文窗口，段落，文本等。

：类别标签，类别可以：sentiment, other words, named entities (loc., org. per.), buy/sell decision, later: mul,-word sequences

：查询表（lookup table）,V是词汇表的大小，查询表中的每一列是一个词向量

:softmax函数的权重矩阵，C是分类总数，矩阵的一行对应于一个类别的参数

:计算属于第y类时用到的权重权重向量，是一个d维的行向量

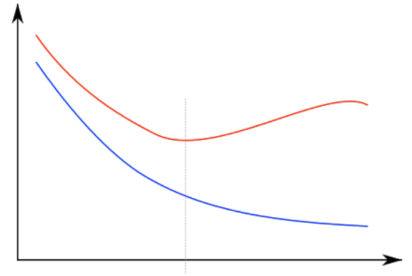
1. softmax回归函数

当给定输入x时，属于第y类的概率通过下式计算

1. 损失函数

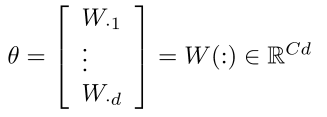
通过公式（2）所定义的损失函数容易过拟合，此时我们将所有参数都正则化，得到如下损失函数

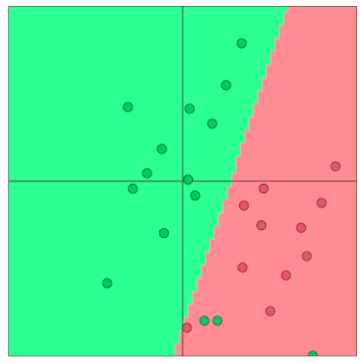
当我们的模型参数很多时或者深度很深时，正则化能够阻止过拟合，下图红线是测试误差，蓝线是训练误差，横轴表示迭代次数

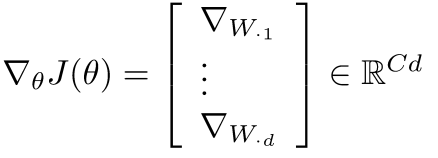


1. 模型细节

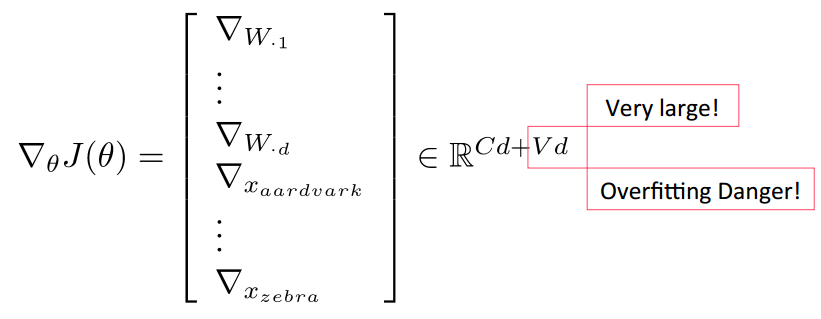
通常来说，softmax回归模型的参数一般只包括权重矩阵W，按列拼接后得一个Cd维的列向量



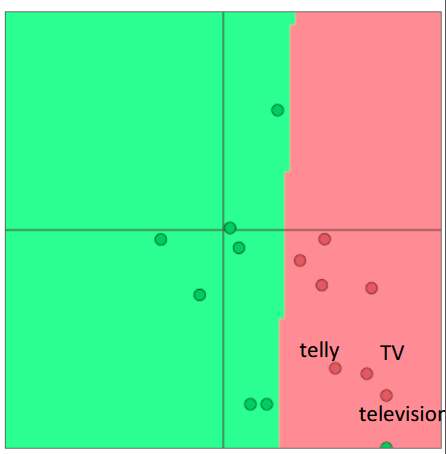
因此，每一次更新只是更新的分类边界



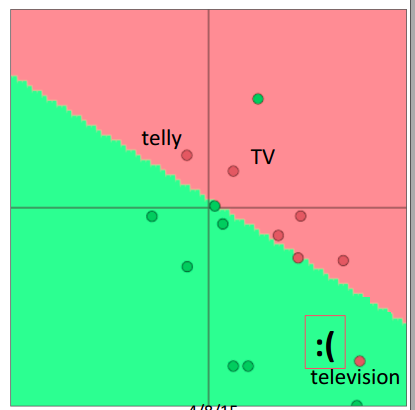
但在这里，我们还需要学习词向量x，



此时出现了一个问题，例如现在有三个词“TV”，“telly”,”television”，将“TV”，“telly”作为训练数据，“television”作为测试数据，初始时，三者关系如下



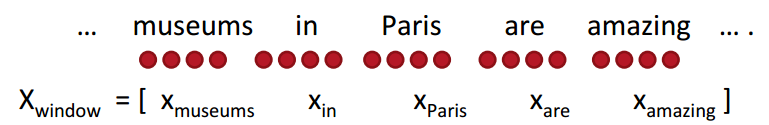
可以发现，三个向量比较近，随着模型的进一步训练，“telly”和“TV”都会有挪动，而做为测试数据的“television”却不会动，出现了下图这种情况



俩点意见：a.如果训练数据很小，则不训练词向量b．如果训练数据很大，训练词向量可能会有更好的结果

1. 基于的词窗的分类

上述讨论都是基于x是一个词向量进行的，然而实际应用中，对一个字的分类并不多见，一般都是对一个词窗进行分类的，对于词窗的分类，其实和对于一个字的分类基本是一样的，唯一的区别是输入向量x是由词窗内所有的词的词向量按列拼接后得到的一个维度更高的向量，举个例子，当窗口大小为2时：



为了方便叙述，继续定义一些符号：

：是一个列向量

：是softmax函数的输出向量，其中第c个维度的值，表示给定x的情况下，词窗分到第c类的概率

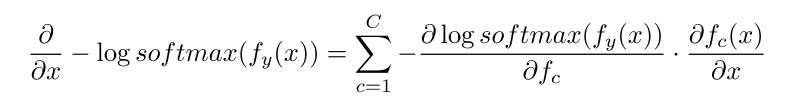
：目标概率分布（除了真正的那一类对应的维度为1，其他维度都为0）

：是向量f的第c个元素，注意到此时W是C\*5d的权重矩阵

与前面相同，依然使用交叉熵作为损失函数：

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

注意这里维度如下

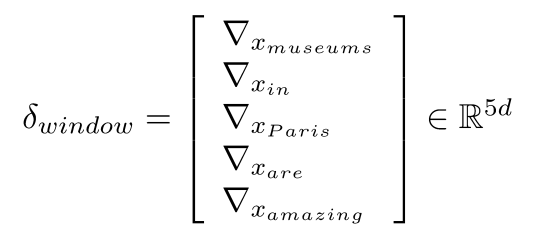


现在，我们得到了的梯度，那么如何更新每一个词的词向量呢，实际上另



当时

我们有

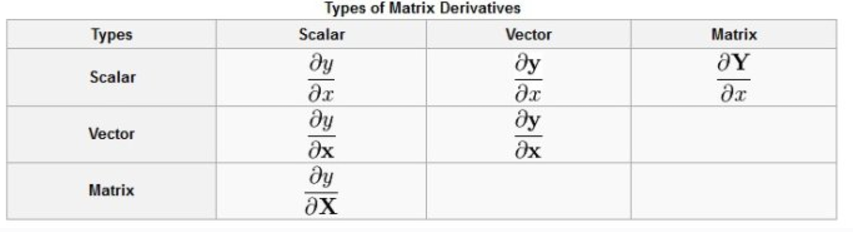


现在我们有了关于x的梯度值，还缺少对权重的梯度值

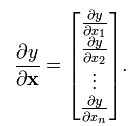
写成矩阵对矩阵求导的形式有

二．引申问题

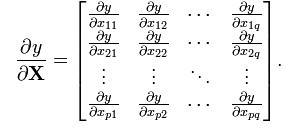
1.矩阵求导



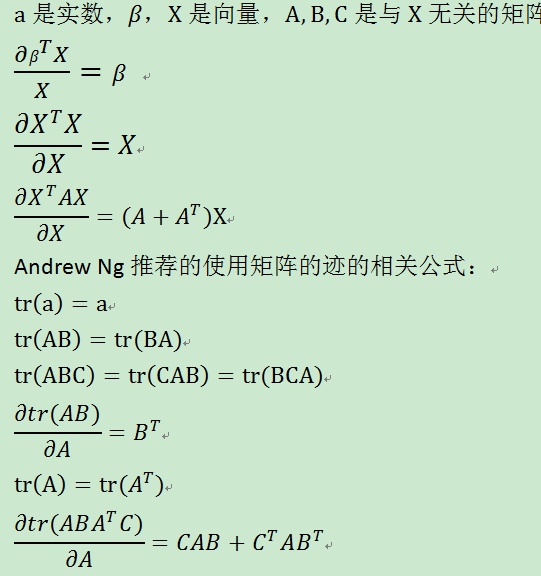
其实在实际的机器学习工作中，最常用到的就是实值函数y对向量X的求导，定义如下（其实就是y对向量X的每一个元素求导）：



实值函数对矩阵X求导也类似：



为了方便推导，下面列出一些机器学习中常用的求导公式，其中andrew ng那一套用矩阵迹的方法还是挺不错的，矩阵的迹也是实值的，而一个实数的迹等于其本身，实际工作中可以将loss函数转化成迹，然后在求导，可能会简化推导的步骤



参考：<http://blog.sina.com.cn/s/blog_8eac0b290101fsqb.html>

1. 思考的问题

1.为什么正则化后的损失函数能够熟练导全局最优