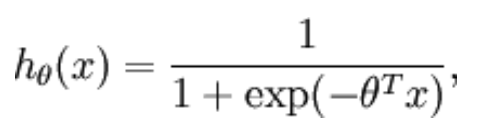
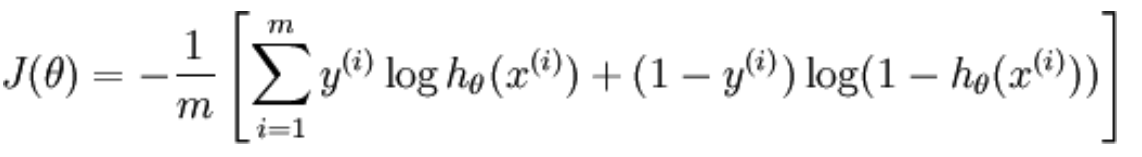
本周任务：阅读Softmax回归模型

Softmax回归：实际上是logistic回归的一般形式，属于一个k分类器

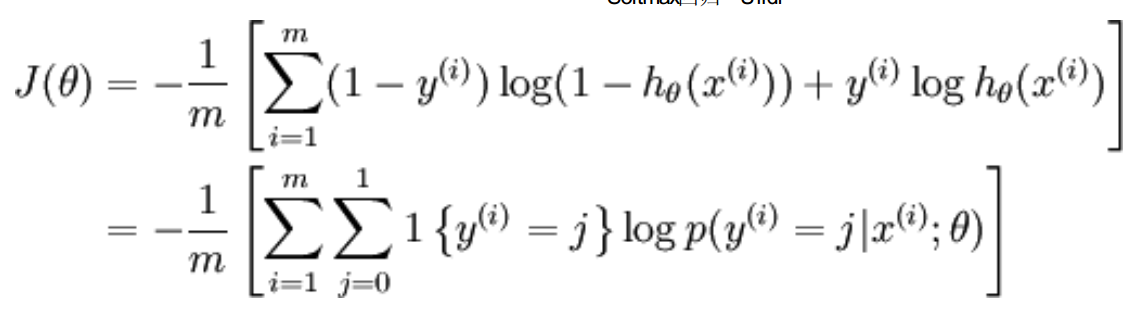
Logistic回归：二类分类器{可以简单地认为是0/1分类问题}，对于输入数据X，预测函数为：

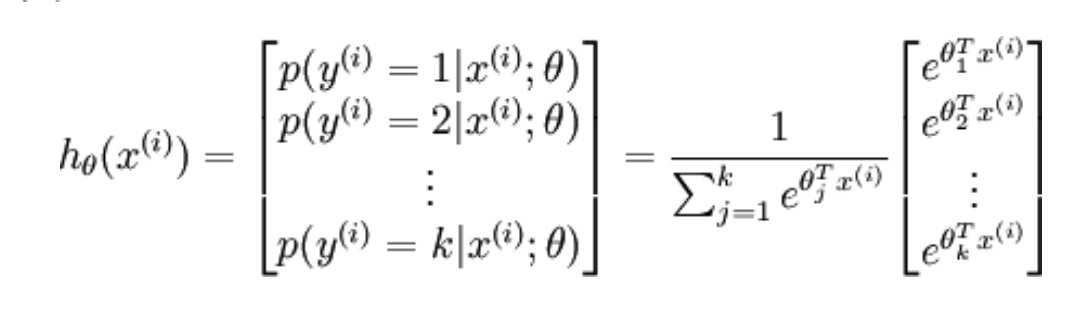
，相应地，代价函数为：



我们训练的目标是让参数θ使得代价函数最小。

Logistic回归函数的更一般形式：

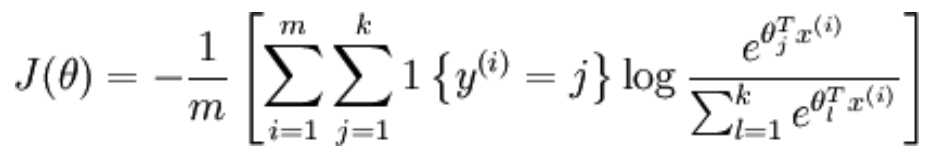
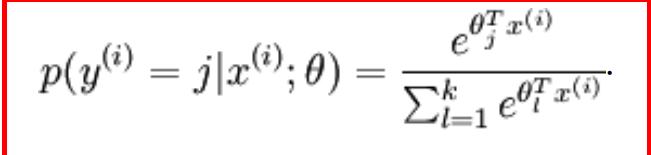


Softmax回归：要对于输入数据X用一个函数针对每一个类别j{1<=j<=k}估算出概率p，其中p为：。所以，我们输出函数的输出值是一个k为的向量，表示输入数据对于每一个类别的概率，那么函数表示为：，其中θ参数是softmax回归模型的参数（我们就是要训练参数θ使得代价最小）。表达式左边的1/Σ是让概率归一化。

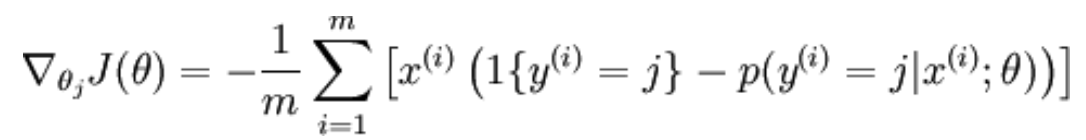
代价函数：目标是训练参数θ使得代价最小

文中引入示性函数：1{·}，当大括号里面的表达式为真时，结果为真否则为假

代价函数表达式如下：

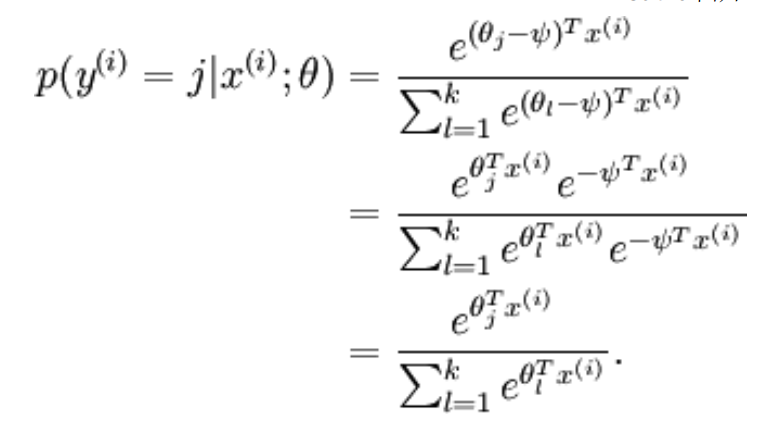
，其中每一个类别j的概率是，softmax的代价函数是对每一种类别的可能值都进行了相加。

那么代价函数对于参数θ的梯度就是：



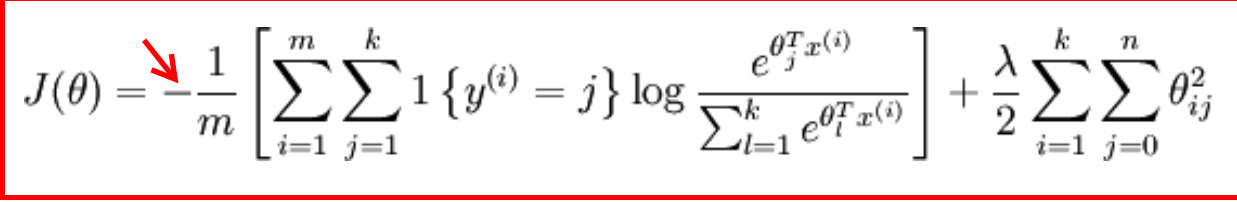
有了梯度，那么我们就可以对参数进行更新：，但是这个梯度函数会带有冗余性（这也是为什么加上了权重衰减的原因）。

参数集冗余性：让参数向量θ减去一个向量φ，那么经过下面的变换后得到的函数输出还是一样的。

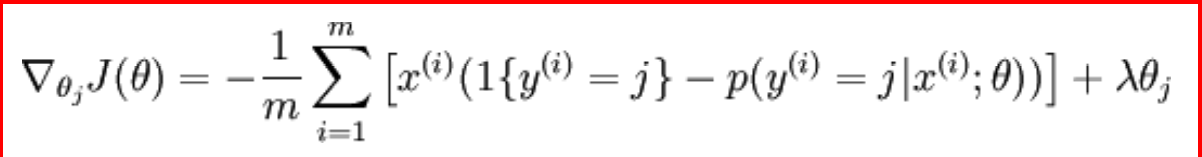
，可见，参数向量减去一个向量对于输出结果是不会有影响的，预测结果还是不会变的。则多组参数（这样的变换有无数种）都会对应相同的输出结果h。鉴于这个问题（过度参数化），加入了权重衰减。

权重衰减：

即往代价函数里添加权重因素——（λ>0）来修改代价函数。因为原来的代价函数的系数是负数，加上这个权重衰减项会使得原来代价大的值减小。那么代价函数修改为：



加入了权重衰减项后，代价函数就变成严格凸函数，这样就不会出现有多个参数满足的情况出现，只会有唯一的参数θ使得函数收敛。此时，梯度下降法就能保证收敛到全局最优解。那么新的代价函数的梯度为：



更加实际的用法：

对于一个k分类问题，如果各个类别互斥，可以用k个类别的softmax回归；如果类别之间不是互斥的，那么可以用k个二类的logistic回归

学习的网址：<http://deeplearning.stanford.edu/wiki/index.php/Softmax_Regression>

下周学习计划：自我学习