

1 神经网络

1.1 softmax, 交叉熵

$$\text{Softmax回归} : \text{softmax}(y)_i = e^{y_i} / \sum_{j=1}^n e^{y_j}$$

Softmax 层将神经网络的输出变成一个概率分布，从而可以通过交叉熵来计算预测的概率分布和真实答案的概率分布之间的距离了。

$$H(p, q) = \sum_x p(x) \log(q(x))$$

交叉熵刻画的是两个概率分布之间的距离，他是不对称的。

$$H(p, q) \neq H(q, p)$$

他刻画的是通过概率分布 q 来表达概率分布 p 的困难程度。因为正确答案是希望得到的结果，所以当交叉熵作为神经网络的损失函数的时候， p 代表的是正确答案， q 代表的师预测值。交叉熵刻画的是两个概率分布的距

离，也就是说交叉熵越小，两个概率分布越接近。

1.2 L1,L2 正则化

L1:Lasso , L2:Ridge

$$L1 : R(w) = ||w||_1 = \sum_i |w_i|$$

$$L2 : R(w) = ||w||_2 = \sum_i |w_i|^2$$

L1: 让参数更稀疏，L2 则不会。因为当参数很小时，比如 0.001，这个参数的平方基本上就可以忽略了，于是模型不会进一步减小这个参数调整为 0。

L1 不可导而 L2 可导。因为在优化的时候需要计算损失函数的偏导数，所以对 L2 正则化的损失函数的优化更加简洁，优化 L1 正则化的损失函数更加复杂。

实践中也可将 L1 和 L2 同时使用：

$$R(w) = \sum_i \alpha |w_i| + (1 - \alpha) w_i^2$$

1.3 神经网络优化过程中可能遇到的问题

1. 通过指数衰减的方式来设置学习率。通过这种方法,既可以加快训练初期的训练速度,同时在训练后期又不会出现损失函数在极小值附近徘徊往返的情况。
2. 通过正则化解决过拟合的问题。
3. 使用滑动平均模型让最后得到的模型在未知数据上更健壮了。

2 卷积神经网络

经典图片分类问题的卷积神经网络结构:

输入层 -> (卷积层-> 池化层)? -> 全连接层

卷积层之间往往会加上一个池化层 (pooling layer), 池化层可以非常有效的所系哦啊矩阵尺寸, 从而减少最后全连接层中的参数, 加快计算速度并且防止过拟合。

常用 max pooling , 其次 average pooling 。

一般卷积层的过滤器边长不会超过 5，但有些卷积神经网络结构中，处理输入的卷积层中使用了边长为 7 甚至 11 的过滤器。

在过滤器的深度上，大部分卷积神经网络都采用了逐层递增的方式。卷积层的步长一般为 1，但是有些模型中会选择 2 或者 3 作为步长，池化层最多用的是最大池化层，池化层过滤器的边长一般为 2 或者 3，步长也一般为 2 或者 3。

3 循环神经网络

3.1 LSTM：长短时记忆网络