超参数调试和batch正则化

http://blog.csdn.net/red\_stone1/article/details/78403416

超参数调节

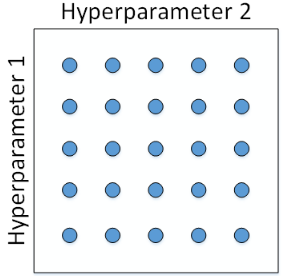
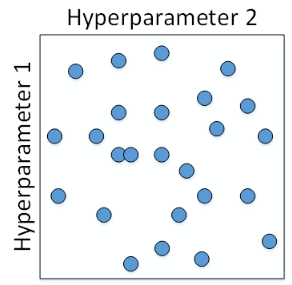
学习速率（含递减参数），算法参数（指数加权平均），神经网络层数和神经元个数，batch大小

重要性，学习速率> 神经元个数和batch size > 神经网络层数 > 算法参数

Adam算法的三个参数beta1，beta2和ε一般常设置为0.9，0.999和10−810−8，不需要反复调试

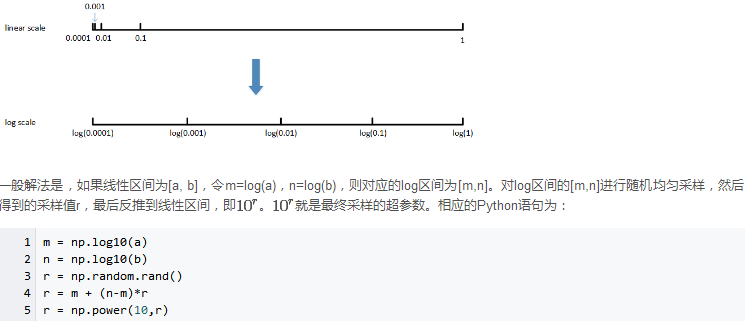
超参数重要性的排名并不是绝对的，具体情况，具体分析

深度学习中，参数调试一般不采用均匀间隔取点，而是使用随机选择 + 由粗到细

使用随机选择，相同取点总量下每个参数都获得更多情况组合。尤其适合参数重要性不相同不确定的场景

采样坐标变换，有些均匀采样需要从线性空间转换为对数空间，才是有意义的均匀



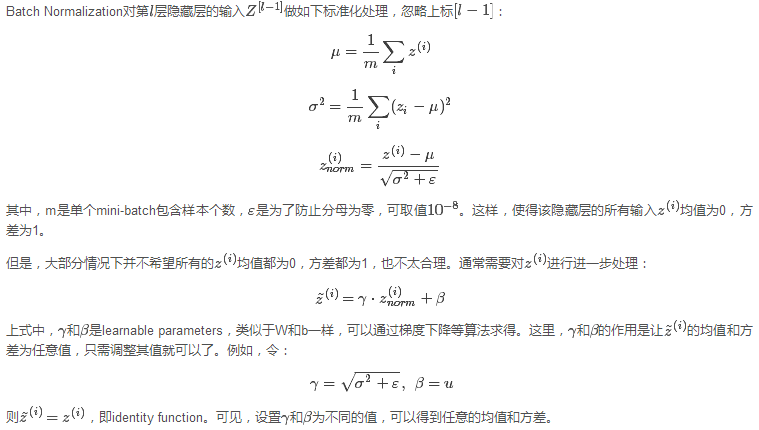
超参数调整不是一锤定音，经过一段时间后，需要根据数据和实际情况再调整

Batch正则化，对于神经网络隐藏层输入进行正则化处理

输入正则化可以提高训练速度，因为“扁碗”归一为“圆碗”后可以使用更大学习速率且不会引起梯度振荡

实际应用中，一般是对Z[l−1]进行标准化处理而不是A[l−1]，差别不是很大。

隐藏层输入通常正则化为非标准正态分布（既均值不为0，方差不为1）。因为对于隐藏层常用的tan激活函数，标准正态分布输入意味着大多数输入都位于线性区域，不利于训练非线性神经网络。如果使用relu激活函数，则不存在此问题。



Batch normal再反向传播中如何计算？？？

Batch normal的作用，不仅在于提高训练速度，**更在于能够减弱covariate shift（协便宜），是模型更具鲁棒性**

训练样本全是黑猫，而测试样本包含各种颜色的猫，导致分类效果较差，这是发生covariate shift的典型

Batch normal因为解耦了各个隐藏输入层之间的耦合关系，使得各层更加独立，所以减弱covariate shift

测试过程单个样本的batch normal。利用训练集获得各个隐藏层batch normal的均值和方差，然后以此做指数加权平均来预估单个测试样本的均值和方差。

Softmax，名称对应hardmax而来。Hardmax中成立元素为1，其余为0。Softmax中则是各个元素成立的概率，比较柔和。

