上两篇笔记介绍针对Training data上性能优化的方法：替换激活函数（New activation function）和Adaptive Learning Rate。本篇笔记则介绍针对Testing data的性能优化方法：Early Stopping & Regularization。

1. **Early Stopping**

我们知道在训练数据集上得到Loss曲线和在测试数据集上得到的Loss曲线并不一定保持一致，例如图1所示：



图1 Early Stopping示意图

如图1所示，在Training Set上Loss曲线趋势与Testing Set上并不是保持一致的，可能在Training Set上Loss比较低的地方，在Testing Set上Loss反而变大了，所以我们需要在Training时使参数停止在一个比较合适的地方，这个思路就是Early Stopping。那如何知道什么时间是合适的呢？我们一般参考Testing Set（Validation Set），用验证集数据作为测试，找个合适的时间点停下。

1. **Regularization**

Regularization就是在原来的Loss function上再加上一个正则项，这个正则项常用的是参数的2范数或者1范数。



其中是原代价函数，比如：最小均方误差、交叉熵等； 是正则项。



1. 当时，正则项是参数的2范数，简称为正则化。



其中 。



梯度下降法更新参数：



是learning rate，一般是个很小的值，如0.01；是正则化系数，一般也很小，比如0.001。每次做迭代时，都比无正则项时的要小一点，这样就更接近0一点，这样让更平滑点。

2）当时，正则项是参数的1范数，简称为正则化。



其中 。



梯度下降法更新参数：



可见，相比于不做正则化，正则化每次迭代的都会减去一个固定的值，当时，会多减去值，当时，会多加上值；也是靠近0。

实际上，正则化和正则化都是让参数变小（接近0），但是方式不一样，是每次迭代都减去（）或加上（）一个固定值接近0,；而是乘一个固定值。

假设：

中：



中：



明显，对于一个很大的参数，在单次更新中变化很小，是固定值；在单次更新中变化则会很大，参数更快速的接近0；所以用做正则化会使得参数最后都比较小，都更接近0；而用做正则化会使得最后参数中能保留一些较大的参数。

不管是正则化还是正则化或者其他形式，实质都是，让参数更多的接近0而避免Over-fitting。



2017.08.31.