深度学习进阶

1. 超参数的选择

建立一个复杂的深度神经网络前，有很多超参数需要考虑：神经网络的层数、每一层中包含多少隐藏神经元、学习速率是多少、每一层使用什么样的激励函数等等。而神经网络最终是否有效，不仅仅和这些超参数有关系，CPU和GPU的性能，输入的训练样本数量，特征数量等都会有很大的影响。

所以一次想要猜中最优的参数几乎是不可能的，整个神经网络的开发过程，应当是一个高度迭代的过程，在提出想法，参数调优，测试结果这三者之间循环。所以说，进行一次这样的循环，所需要的时间，其实是决定整个项目需要的时间的关键因素。

2. 模型性能的评估

首先，整个数据分成两部分，70%的训练数据和30%的测试数据，然后从70%的训练数据中，再拿出一部分作交叉验证的开发集（通常是60%，20%，20%）。每次得到一个新的模型后，首先进行交叉验证，然后选出交叉验证中，性能最好的模型，进行测试数据的验证。

测试集与开发集与交叉验证的作用是快速地对多个模型进行评估，选出其中最好的。所以在训练数据很多（百万）的情况下，也不需要多达20%的数据来进行评估了，拿出10000左右的数据量就足够了。

3. 保证训练集与开发集的数据分布是相同的

举个例子：我们想要开发一个让用户上传猫的图片，并进行识别是否是猫的系统，我们选取的训练集与开发集是网上的各种猫的图片，测试集则是用户通过手机上传的猫的图片。但是这里可能会遇到问题，网上的猫的图片可能是高清的，非常精美的，而用户上传的图片会很模糊，这样的话，即使我们的模型在开发集上取得了不错的成效，它对于测试集的性能也是未知的。

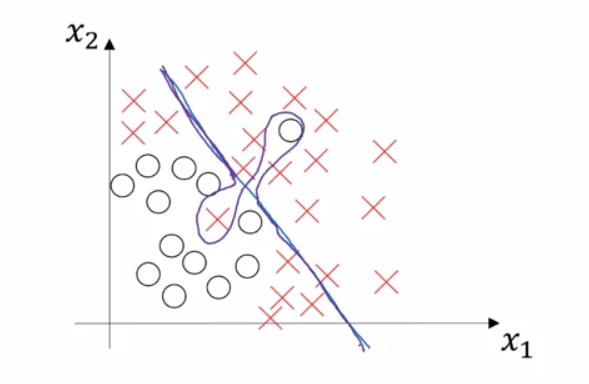
因此，最好能够保证训练集与开发集的数据分布是一致的。

4. 偏差与方差

高方差：模型在训练集上的表现非常好，比如只有1%的错误率；但是在开发集上的表现没那么好，比如有11%的错误率，这种情况下显然说明存在过拟合问题，模型不具备泛化性。

高偏差：比如模型对于猫的识别，在训练集上错误率是15%，而在开发集上错误率是16%。训练集和开发集上性能差不多，说明模型有较好的泛化性，但是15%的错误率显然说明模型存在欠拟合的问题，没有对训练集进行良好的分类。

最让人满意的模型，是低方差低偏差的，比如在训练集上0.5%的错误率，在开发集上1%的错误率。（当然这是在理想的人的识别率为100%的情况下，如果人的识别率都只有90%，那10%错误率的模型已经是令人满意的了）

这就是一个高偏差+高方差的分类器。

5. 机器学习基本原则

对一个训练好的神经网络进行优化与评估的一般原则：

a. 判断是否存在高偏差，查看模型在训练集上的表现。尝试使用新的神经网络结构，比如更多的隐藏层和隐藏神经元；或者可以去寻找一个已有的，能解决当前问题的神经网络结构。在解决了偏差的问题之前，不要进行后续的步骤。

b. 判断是否存在高方差，查看模型在开发集上的表现，看看分类是否具有一般性。解决高方差的办法，最好是获取更多的训练数据；接着是正则化，能够减少过拟合的问题；最后是尝试更好的神经网络结构，有时可以解决过拟合的问题。

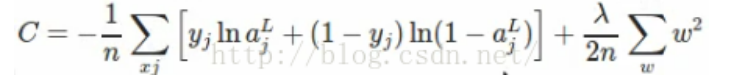
不同于早起的机器学习，在深度学习中，方差与偏差之间没有那么强大的权衡关系，使用更大的网络可以单独减小偏差而不会对方差有大的影响，而提供更多的数据可以单独减小方差而不会影响偏差。所以在优化时，使用这两种方法总是没错的，代价只是需要更多的训练时间。

6. 神经网络中的L2正则化

L2正则化可以用来改进过拟合的模型，减小方差。

在逻辑回归中，L2正则化的做法是在代价函数的末尾，增加各个w参数的平方项，同时以一个lameda来控制这些平方项产生的影响。

在神经网络中，这样的做法也是适用的，在正向传播的代价函数中，添加所有w权重的平方项。



经过计算，最后得到的结果是，在每次后向传播，更新各个w之前，都会把w乘以一个0~1之前的数，使得w的值尽可能地小，而影响程度通过lameda来控制。

7. dropout正则化