

实践方法论

要成功的使用深度学习技术，仅仅知道存在那些算法和解释他们为何有效的原理是不够的，一个优秀的机器学习实践者还需要知道如何针对具体应用挑选一个合适的算法以及如何监控，并根据实验反馈改进机器学习系统。机器学习系统开发中，实践者需要决定

是否收集更多的数据

是否增加训练集的多样性

是否训练更长时间

是否使用更好的优化方法

是否增加和减小模型容量

是否添加后删除正则化项

是否使用dropout

是否early stopping

是否尝试新的网络架构

是否尝试新的激活函数

...

实践中正确使用一个普通的方法通常比草率的使用一个不清楚的算法效果更好。正确的使用方法需要掌握一些简单的方法论：

实践设计流程：

确定目标：使用什么样的误差度量，并为此误差度量指定目标值，这些目标和误差度量取决于该应用旨在解决的问题。

尽快建立 一个端到端的工作流程，包括估计合适的性能度量

快速迭代 使用偏差方差分析和误差分析技术，确定性能瓶颈，检查那个部分的性能低于预期，反复进行增量式的改进

正交化

在《从VC维度理解正则化和偏差方差分析》一文中指出，机器学习的两个核心问题是：1、**使训练误差足够小**，2、**使泛化误差与训练误差足够接近**，我们将数据分为训练集(train set)，开发集(dev set),测试集(test set)，但是我们开发机器学习系统的目标是在我们未看过的真实世界的的数据中表现良好，因此，如果，如果出现问题，我们需要定位误差的来源，主要包括以下几个方面：

1、**训练集上是否表现良好**

2、**开发集上是否表现良好**

3、**测试集上是否表现良好**

4、**真实世界数据是否表现良好**

本文开头介绍了一些机器学习的实践者可以调整的因素，正确定位问题和使用正确的调整策略可以帮助我们更快速的开发机器学习系统。正交化是指，某一可调整的因素只影响机器学习系统的某一方面，或者对其他方面影响都很小，正交化的因素只影响机器学习系统的某一方面，相对独立于其他问题，可以显著的减少开发和测试的时间。

1、如果训练集上表现不好，欠拟合，可以尝试：

提高模型的容量（训练一个更大的网络）

训练更长时间

采用更好的训练算法

尝试不同的网络架构

2、训练集上表现良好，但开发集表现不好，过拟合，可以尝试：

增加训练集的大小，收集更多的数据

减小模型的容量

训练更小的网络

使用更大的规则化

使用dropout

early stopping 可以减小过拟合，但是它也会影响模型的训练集上的表现，不是正交化因素，谨慎使用

3、训练集，开发集表现良好，测试集表现不好，对开发集过拟合，可以尝试：

增大开发集

4、训练集，开发集，开发集表现良好，真实应用表现不好，说明可能开发集分布设置不正确，或者成本函数测量的指标不对：

改变开发集或cost function

确定目标

确定目标，即选择合适的误差度量（指标），是必要的第一步，误差度量将指导接下来的所有工作，也应该了解大概能达到什么样的目标。

对于大多数应用而言，由于**输入特征可能无法包含输出变量的完整信息，受限于有限的训练数据**，绝对误差不可能实现绝对零误差，贝叶斯误差定义了能达到的最小错误率。

使用单一数字评估指标

尝试从多个超参数试验中选择最佳模型时，如果评估指标有多个，将非常难以判断优劣，因此，需要选择合适的单一评估指标。

设立满足指标和优化指标

多个指标合成一个指标有时候不太容易，此时可以设立满足指标和优化指标来将多个指标结合起来，可以提供明确的方案。有N个指标时，将其中一个作为优化指标，其他作为满足指标。

比如对于图像分类来说，准确率作为优化指标，满足指标为运行时间。

什么时候改变开发、测试集和指标

1、当评估指标不能正确衡量算法之间的优劣排序时，需要改变评估指标或开发，测试集：

$$err = \frac{1}{\sum_{i=1} w^i} \sum_{i=1}^{m_{dev}} w^i 1\{y_{pred}^i \neq y^i\}$$

2、当我们定义的指标在开发集测试上上都有很好的表现，但不能很好的在真实应用中表现良好时，需要改变指标和开发、测试集。

不匹配数据的划分

深度学习算法需要大量的数据，因此机器学习团队尽可能的收集数据，有些数据，甚至是大部分的数据，都来自和开发集、测试集不同的分布，因此，许多开发团队都使用与开发集和测试集不同分布的数据来训练模型。为了保证很好的泛化效果，**要尽量保证开发集，测试集数据与真实目标数据的分布一致**

通过偏差方差学习确定机器学习努力的方向，但当开发集、测试集与训练集不同分布时，如果训练集上的误差与开发集上的误差差别比较大，很难判断是由于方差引起的，还是由于分布不同引起的。因此，为了确定是哪一组因素影响的，需要重新设立一组**训练开发集**：其分布于训练集一致，但是不用于训练，然后分析在训练集，训练开发集，开发集上的误差，如果训练开发集与训练集的误差相差比较大，而与开发集误差相差比较小，那说明误差的来源是方差，因此需要采取减小方差的策略。相反，如果训练开发集误差与训练集误差非常接近，但与开发集有较大差别，那么误差来源于不同分布，称为**数据不匹配**问题，此时需要采取解决数据不匹配问题的方法。

接近数据不匹配问题策略

数据不匹配问题没有完全系统的解决方案

1、做误差分析，尝试了解训练集与开发集的具体差异，找到具体差异，然后使得训练集与开发集更相似，或者收集更多的与开发集（比如人工合成数据集，但有一个潜在的问题，合成的数据只是潜在分布的一个很小的子集，因此可能对合成数据过拟合），测试集相似的数据集。

2、迁移学习（transfer learning）：把一个通用的模型迁移到一个小数据集上，使其个性化，使能够在一个新的领域也能产生效果。

human performance

人类在许多任务上都非常擅长，当机器学习系统没有人做的好时，机器学习系统可以通过人类获得以下帮助

1、获得标记数据

2、人类可以帮助进行误差分析，找清楚错误原因

3、更好的进行偏差方差分析

当机器学习系统超过人的表现时，将得不到上述帮助，因此，机器学习将会变得更加困难，比如当机器学习系统的表现超过人类表现时，将很难分析到底是偏差还是方差影响系统的表现。

可避免偏差：人类水平误差和训练集误差之间的差

误差分析

如果希望机器学习系统能够胜任人类的工作，但还没达到人类的表现时，人工检查算法所范的错误，即在开发集或测试集上进行误差分析，并行评价多个影响因素，找出影响机器学习系统评估指标最主要的因素，可以帮助找到下一步努力的方向，然后努力在这些方面进行改进。

是否修正标签错误

深度学习算法对随机误差非常鲁棒，但对系统误差容忍度比较低，因此当训练集上标记错误为随机错误，且不是很大时，可以不用校正。

但开发集、测试集上存在标记错误时，可以在误差分析表格中，添加Incorrectly labeled 项，当标记错误对评估指标影响比较大时，需要校正标签，但是影响很小时，可以忽略这些标签错误。

Error analysis

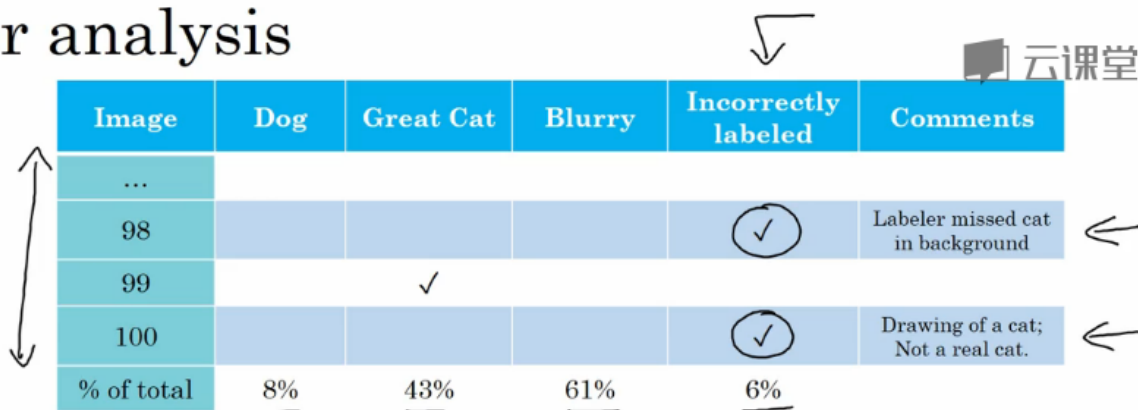


Image	Dog	Great Cat	Blurry	Incorrectly labeled	Comments
...					
98				✓	Labeler missed cat in background
99		✓			
100				✓	Drawing of a cat; Not a real cat.
% of total	<u>8%</u>	<u>43%</u>	<u>61%</u>	<u>6%</u>	

如果要修正标签时，一定在开发集测试集上进行相同的操作，以保证开发集测试集同一分布

超参数的选择

超参数有两种基本方法：手动选择和自动选择，手动选择需要对超参数有比较深入的了解，以及机器学习模型如何取得良好的性能。自动选择往往需要更高的计算成本。

手动搜索超参数 的目标是最小化受限于运行时间和内存预算的泛化误差。

自动搜索超参数：

网格搜索：对每个超参数，使用者选择一个较小的有限制取探索，这些超参数的笛卡尔乘积得到一组组超参数，网格搜索使用每组超参数训练模型，挑选使开发集误差最小的超参数。

网格搜索集合的范围：

随机搜索：网格搜索大量超参数需要大量的运行时间，随机搜索可以很方便，更快的收敛到超参数的良好取值。事实上，我们并不一定要找到所有超参数中最好的，找到前k个好的，可能就足够了，假设总共有N个超参数对，搜索m次找到前k个好的概率为 $1 - (1 - \frac{k}{N})^m$ ，例如，假设N=100，k=10，m=25,那么我们有93%的概率搜索到了前10个最佳的超参数。

随机搜索时，一般不在区间上进行均匀搜索，基于超参数的敏感性考虑，可以在对数尺度上随机搜索（0.1和0.2影响不是很大，但0.1，0.01,0.001相差可能很大）