Draft Version

MACHINE LEARNING YEARNING

Technical Strategy for AI Engineers, In the Era of Deep Learning

ANDREW NG

Machine Learning Yearning is a
deeplearning.ai project.

15 在错误分析中并行多个想法

你的团队有以下几个想法，来改进你的猫咪分类器:

- 解决狗被错误分为猫咪的问题。
- 解决“大型猫科动物(greast cats)”（狮子或豹子等）被错认家猫（宠物）的问题
- 提高系统在模糊（Blurry）图像上的表现
- ...

你可以并行并且有效的评估这些想法。我通常会创建一个表格，查看 100 个分类错误的开发集样本并记录在表格上，同时进行注释。用有小开发集里的 4 个错误分类样本来说明这个过程，你的表格大概将会是下面的样子:

Image	Dog	Great cat	Blurry	Comments
1	✔			Unusual pitbull color
2			✔	
3		✔	✔	Lion; picture taken at zoo on rainy day
4		✔		Panther behind tree
% of total	25%	50%	50%	

表格中 Image3 的 Great cat 和 Blurry 列都被勾选了：可以将一个样本与多个类别相关联， 这就是为什么最后的百分比加起来不足 100%的原因。

虽然我已经将这个过程首先描述为类别分类（Dog, Great cat, Blurry），然后查看样例对他们进行分类。实际中，当你查看样例时，可能会受到启发，然后提出一些新的错误类别。例如，当你查看过十几张图像后，你发现许多错误都经过 Instagram（一款美图软件）的滤镜处理。你可以在表格中添加一列 Instagram，看看图像是否被滤镜处理过。手动查看算法出错的样例，并思考正常人是如何将这些样例正确分类的。这通常会启发你提出新的类别和解决办法。

你的想法对于改进错误类别是非常有用的。例如：如果你没办法将经过 Instagram 处理的图像还原的话，那么添加 Instagram 类别是最好的办法。但是你不必局限于你已经有想法去解决这个问题；这个过程主要目的是帮助你找到你认为最值得关注的问题。

错误分析是一个迭代的过程。开始的时候在你脑海中可以没有任何分类。通过查看图片，你可能会提出一些关于错误类别的想法。然后手动分类一些错误图片以后，可能会启发你想出一些新的错误类别，根据新的类别在返回重新检查这些图片，以此类推。

假设你完成了 100 个错误分类的开发集样本，得到如下表格:

Image	Dog	Great cat	Blurry	Comments
1	✔			Usual pitbull color
2			✔	
3		✔	✔	Lion; picture taken at zoo on rainy day
4		✔		Panther behind tree
...	...	...	...	...
% of total	8%	43%	61%	

你现在知道解决狗分类错误的问题，最多可以消除 8%的误差。而致力于 Great cat 和 Blurry 对你的帮助更大。因此，你可能会挑选后者之一来进行处理。如果你的团队有足够多的人可以同时展开多个方向，你让一部分人解决 Great cat 问题，另一部分人解决 Blurry 问题。

错误分析并不会得出一个明确的数学公式来告诉你最应该先处理哪个问题。你还必须考虑在不同错误类别上取得的进展，以及每个错误类别所需的工作量。

16 清理贴错标签的开发集和测试集样本

在错误分析期间，你可能会注意到开发集中的一些样本被错误标记(mislabeled)。当我说”dislabeled”时，我的意思是在模型训练之前，这个样本被错误的打了标签。即(x, y)中的类别 y 值不正确。例如，也许一些不是猫咪的图片被错标记为猫咪,反之亦然。如果你觉得一小部分的被错误标记的样本很重要，你可以再添加一个错误标记的类别：

Image	Dog	Great cat	Blurry	Mislabeled	Comments
...					
98				✔	Labeler missed cat in background
99		✔			
100				✔	Drawing of a cat; not a real cat.
% of total	8%	43%	61%	6%	

你应该纠正被错误标记的样本吗？记住，开发集的目的是为了帮你快速评估算法，以便你可以判断算法 A 或 B 哪个更好。

例如，假设你的分类器表现如下：

- 开发集的整体准确率.....90%(10%整体错误率)
- 贴错标签样本导致的错误.....0.6%(开发集错误的 6%)
- 其它原因导致错误.....9.4%(开发集错误的 94%)

这里，相对于你可能正在改进 9.4%的错误，由于错误标注导致 0.6%的不准确率可能没那么重要。手动修正开发集中的错误是可以的，但这不是关键。不知道系统是否有 10%或 9.4%的整体错误可能没什么问题。

假设你不断改进猫咪分类器并达到以下性能：

- 开发集整体准确率..... 98.0% (2.0% 整体误差.)
- 错误被标记的样本导致的误差..... 0.6%. (开发集错误的 30%.)
- 其它原因导致的误差..... 1.4% (开发集错误的 70%)

30%的错误是由于错误标注的开发图像造成的。这时候你需要改进你的开发集中的标注质量。处理错误的样本将帮助你算出分类器的错误是 1.4%还是 2%——这是一个相对比较明显的差异。

容忍开发集中的一些错误标注样本是很常见的，随着系统的改进，使得错误标注的原因占总误差的比例更高。

最后一章解释了如何通过算法的提升来改进错误标注的类别，例如：狗。猫科动物和模糊图像。本章你将会学到，你也可以在错误标记的类别上对标签进行改进。

无论你采用什么方法来修正开发集标签，记得也将其用于测试集标签，以便开发集和测试集任处于同一分布。开发集和测试集处于同一分布可以解决我们在第六章遇到的问题。（你的团队优化了开发集的性能，只是到后来他们才发现在根据不同的测试集进行不同的评估）。

如果你决定提升标签的质量，那么请考虑仔细检查系统错误分类的样本的标签。以及正确分类的样本标签。在一个样本中，原始标签和学习算法可能都是错误的。如果只是修正系统已经错误分类的样本的标签，最后可能会在你的评估中引入误差。如果有 1000 个开发集样本，并且分类器的准确率为 98%，那么检查错误分类的 20 个样本比检查正确分类的所有 980 个样本要容易的多。因为在实际中只检查错误分类的样本比较容易，所以偏差会蔓延到一些开发集中。如果你只对开发产品和应用程序感兴趣，那这种偏差是可以接受的。但是如果你计划在学术论文中使用这个结果，或者需要一个完全无偏差的测量测试集的准确率，就不是个很好的选择。

17 如果你有一个很大的开发集，把它分为两个子集，只着眼于其中一个

假设你有一个含有 5000 个样本的大型开发集，其中有 20%的错误率。这样，算法对约 1000 个图片进行错误分类。手动检查 1000 张图片是非常耗费时间的，所以我们可能决定在错误分析中不使用所有的图片。

在这种情况下， 我会明确的将开发集分为两个子集，只看其中一个子集，另一个不看。你可能会在你查看的那部分数据中过拟合，此时你可以使用那部分未使用的数据来进行调参。

继续上面的例子，在该例子中算法错误分类 5000 个开发集样本中的 1000 个。假设我们想手动检查约 100 个错误样本（错误分类的 10%）进行分析。你应该随机选择 10%的开发集，并将其放入我们称之为 Eyeball 开发集（Eyeball dev set）中，以提醒我们正在使用它。（对于语音识别项目，你的数据集为语音，你需要一个一个听它们，你可以将它们称为 Ear dev set）。因此，Eyeball 开发集有 500 个样本，其中我们预计算法会错误分类约 100 个。

开发集的第二个子集叫做 Balckbox 开发集（Blackbox dev set），它将拥有剩下的 4500 个样本。你可以使用 Blackbox 开发集，通过测量它们的错误率来自动评估分类器。也可以使用它来选择算法或调整参数。但是，你应该避免使用它。我们使用“Blackbox”术语是因为我们只使用数据集的子集来获得分类器的“Blackbox”评估。

为什么我们将开发集明确的分为 Eyeball 开发集和 Blackbox 开发集呢？如果你使用 Eyeball 进行开发的话，你的算法可能会过拟合。如果你发现 Eyeball 开发集比 Blackbox 开发集性能提升的更快，你已经过拟合 Eyeball 开发集。在这种情况下，你可能需要一个新的 Eyeball 开发集，将更多 Blackbox 开发集中的样本移至 Eyeball 中。也可以通过获取新的标注数据来获得。

将开发集明确分为 Eyeball 和 Blackbox 然后手动误差分析可以让你知道何时在 Eyeball 上导致过拟合。

18 Eyeball 和 Blackbox 开发集应该多大？

你的 Eyeball 开发集应该足够大，大到可以让你了解到算法的主要错误类别。如果你正在从事一项人类可以表现很好的任务（如识别图像中的猫咪），下面是一些指导方针：

• 一个使你分类器犯错 10 次的 Eyeball 开发集将会被认为是非常小的。只有 10 个错误，很难准确估计不同错误类别的影响。但是如果你的数据非常少，而且不能分出更多的 Eyeball 开发集，有总比没有好，这将有助于项目的优先顺序。

• 如果分类器在 Eyeball 开发集上犯错约 20 个样本，你将会大致了解主要的错误来源。

• 如果有约 50 个错误，你将会比较好的了解错误的来源。

• 如果有约 100 个错误，你将会很清楚错误的来源。我见过有人手动分析更多的错误——有时候多达 500 个。只要你有足够的数据。

假设你的分类器有 5%的错误率，为了确保 Eyeball 开发集中有 100 个错误标记的样本，Eyeball 开发集大概有 2000 个样本（因为 0.05*2000 = 100）.分类器的错误率越低，为了获取足够多的错误来分析，Eyeball 开发集需要足够大。

如果你正在做一个连人类都不能做好的任务，那么检查 Eyeball 开发集的联系将没有什么必要。因为很难找出算法不能正确分类一个样本的原因。这种情况下，你可能会忽略 Eyeball 开发集。我们将在后面的章节继续讨论这些指导方针。

Balckbox 开发集该多大呢？我们之前说过，开发集约有 1000-10000 个样本是正常的。一个有 1000-10000 样本的 Blackbox 开发集通常会为你提供更多足够的数据去调整参数和选择模型。一个含有 100 个样本的 Blackbox 开发集虽然小，但也还是管用的。

如果你有一个小开发集，那么你可能没有足够的数据将它分为足够大的 Eyeball 和 Blackbox 开发集。你的整个数据集都不得不用于 Eyeball 开发集，即你需要对数据一一进行检查。

在 Eyeball 和 Blackbox 开发集之间，我认为 Eyeball 开发集更重要（假设你正在研究一个人类可以很好解决的问题，检查这些样本可以提高你的洞察力）。如果你只有一个 Eyeball 开发集，你可以在这个开发集上进行错误分析，模型选择和参数调整。只有一个 Eyeball 开发集的缺点是过拟合开发集的风险更大。

如果你有足够的数据，那么 Eyeball 开发集的大小将主要取决于你有时间去手动分析样本的数据量。我很少看到有人分析超过 1000 个错误数据的。

19 总结：基本错误分析

- 当你开始一个新项目时，尤其你不是这个领域的专家时，你很难选择最有前途的方向。
- 不要一开始就尝试设计和构建完美的系统，而是尽可能快的建立和训练一个基础的系统（几天之内），然后使用错误分析。帮助你找到最优的方向，并迭代改进你的算法。
- 通过手动检查约 100 个开发集的样本来进行错误分析。计算错误分类的主要原因的比例，然后使用此信息来选择你需要修复的错误类型。
- 考虑将开发集设置为一个 Eyeball 开发集和 Blackbox 开发集，并对 Eyeball 开发集进行手动检查误差分析，如果算法在 Eyeball 上表现的性能比在 Blackbox 上表现的性能要好，说明你的算法在 Eyeball 开发集上过拟合了，这是，你需要从 Blackbox 中选择数据放入 Eyeball 中。
- Eyeball 开发集应该设置的足够大，这样你就可以得到足够被算法错误分类的数据，然后进行手动分析。对于许多应用程序来说，Blackbox 开发集有 1000-10000 个样本就差不多了。
- 如果你的开发集不足以划分，就把它划分为一个 Eyeball 集，并进行手动误差分析，模型选择，和参数调整。