deeplearning.ai

吴恩达 DeepLearning.ai 课程提炼笔记（3-2）结构化机器学习项目 --- 机器学习策略(2)

以下为在Coursera上吴恩达老师的DeepLearning.ai课程项目中，第三部分《结构化机器学习项目》第二周课程“机器学习策略（2）”关键点的笔记。本次笔记并几乎涵盖了所有视频课程的内容。在阅读以下笔记的同时，强烈建议学习吴恩达老师的视频课程，视频请至 Coursera 或者 网易云课堂。

---

1. 误差分析

当我们在训练一个模型的时候，如一个猫和狗分类模型，最终得到了 90% 的精确度，即有 10% 的错误率。所以我们需要对模型的一些部分做相应调整，才能更好地提升分类的精度。

如果不加分析去做，可能几个月的努力对于提升精度并没有作用。所以一个好的误差分析的流程就相当重要。

收集错误样例：

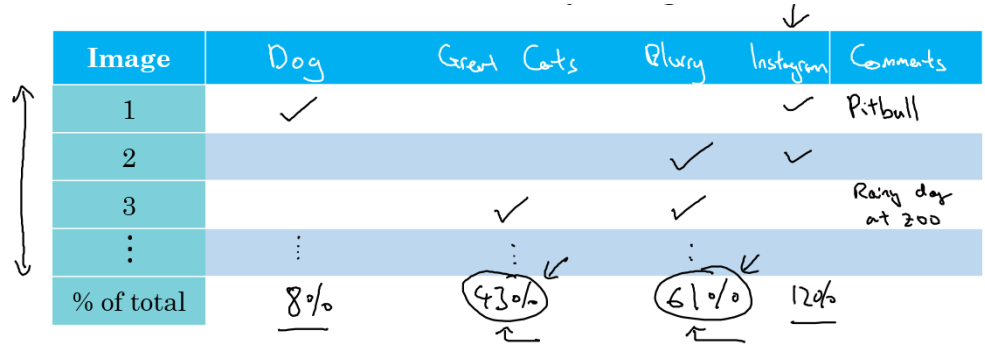
在开发集（测试集）中，获取大约100个错误标记的例子，并统计其中有多少个是狗。

- 假设一种情况是100个数据中，有5个样例是狗，那么如果我们对数据集的错误标记做努力去改进模型的精度，那么可以提升的上限就是 5% ，即仅仅可以达到 9.5% 的错误率，这有时称为**性能上限。**那么这种情况下，可能这样耗时的努力方向就不是很值得的一件事情。
- 另外一种假设是100个数据中，有50多个样例是狗，那么这种情况下，我们去改进数据集的错误标记，就是一个比较值得的改进方向，可以将模型的精确度提升至 95% 。

并行分析：

- 修改那些被分类成猫的狗狗图片标签；
- 修改那些被错误分类的大型猫科动物，如：狮子，豹子等；
- 提升模糊图片的质量。

为了并行的分析，建立表格来进行。以单个错误分类样本为对象，分析每个样本错误分类的原因。



最后，统计错误类型的百分比，这个分析步骤可以给我们一个粗略的估计，让我们大致确定是否值得去处理每个不同的错误类型。

2. 清除错误标记的样本

下面还是以猫和狗分类问题为例子，来进行分析。如下面的分类中的几个样本：

情况一：

深度学习算法对训练集中的随机误差具有相当的鲁棒性。

只要我们标记出错的例子符合随机误差，如：做标记的人不小心错误，或按错分类键。那么像这种随机误差导致的标记错误，一般来说不管这些误差可能也没有问题。

所以对于这类误差，我们可以不去用大量的时间和精力去做修正，只要数据集足够大，实际误差不会因为这些随机误差有很大的变化。

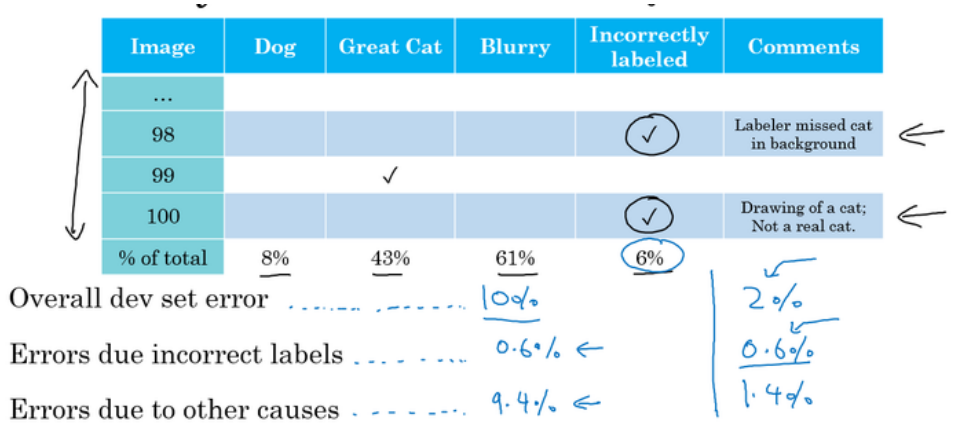
情况二：

虽然深度学习算法对随机误差具有很好的鲁棒性，但是对于系统误差就不是这样了。

如果做标记的人一直把如例子中的白色的狗标记成猫，那么最终导致我们的分类器就会出现错误。

dev、test集中错误标记的情况：

如果在开发集和测试集中出现了错误标记的问题，我们可以在误差分析的过程中，增加错误标记这一原因，再对错误的数据进行分析，得出修正这些标记错误的价值。



修正开发、测试集上错误样例：

- 对开发集和测试集上的数据进行检查，确保他们来自于相同的分布。使得我们以开发集为目标方向，更正确地将算法应用到测试集上。
- 考虑算法分类错误的样本的同时也去考虑算法分类正确的样本。（通常难度比较大，很少这么做）
- 训练集和开发/测试集来自不同的分布。

3. 搭建系统

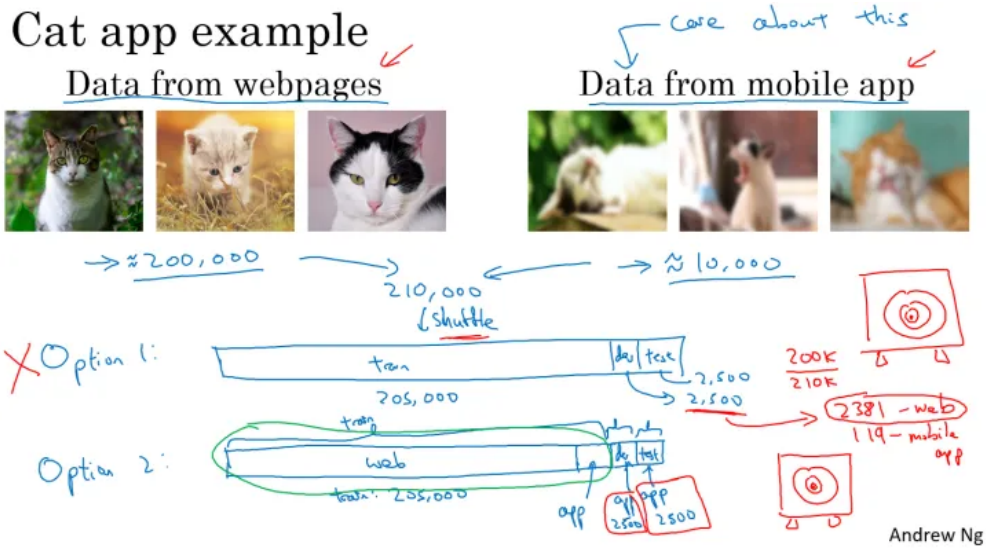
- 设置开发、测试集和优化指标（确定方向）；
- 快速地建立基本的系统；
- 使用偏差方差分析、误差分析去确定后面步骤的优先步骤。

总的来说，如果我们想建立自己的深度学习系统，我们就需要做到：快速的建立自己的基本系统，并进行迭代。而不是想的太多，在一开始就建立一个非常复杂，难以入手的系统。

4. 不同分布上的训练和测试

在深度学习的时代，因为需求的数据量非常大，现在很多的团队，使用的训练数据都是和开发集和测试集来自不同的分布。

下面是一些处理训练集和测试集存在差异的最佳的做法。以前一周中的猫的分类问题为例：



我们可以从网上获取大量的高清晰的猫的图片去做分类，如200000张，但是只能获取少量利用手机拍摄的不清晰的图片，如10000张。但是我们系统的目的是应用到手机上做分类。

也就是说，我们的训练集和开发集、测试集来自于不同的分布。

方法一：

将两组数据合并到一起，总共得到21万张图片样本。将这些样本随机分配到训练、开发、测试集中。

- 好处：三个集合中的数据均来自于同一分布；
- 坏处：我们设立开发集的目的是瞄准目标，而现在我们的目标绝大部分是为了去优化网上获取的高清晰度的照片，而不是我们真正的目标。

该方法不是一个好的方法，不推荐。

方法二：

训练集均是来自网上下载的20万张高清图片，当然也可以加上5000张手机非高清图片；对于开发和测试集都是手机非高清图片。

- 好处：开发集全部来自手机图片，瞄准目标；
- 坏处：训练集和开发、测试集来自不同的分布。

从长期来看，这样的分布能够给我们带来更好的系统性能。

5. 不同分布上的偏差和方差

通过估计学习算法的偏差和方差，可以帮助我们确定接下来应该优先努力的方向。但是当我们的训练集和开发、测试集来自不同的分布时，分析偏差和方差的方式就有一定的不同。

方差和分布原由分析

以猫分类为例，假设以人的分类误差 0% 作为贝叶斯误差。若我们模型的误差为：

- Training error： 1%
- Dev error： 10%

如果我们的训练集和开发、测试集来自相同的分布，那么我们可以说模型存在很大的方差问题。但如果数据来自不同的分布，那么我们就不能下这样的定论了。

那么我们如何去确定是由于分布不匹配的问题导致开发集的误差，还是由于算法中存在的方差问题所致？

设立“训练开发集”

训练开发集，其中的数据和训练数据来自同一分布，但是却不用于训练过程。

如果最终，我们的模型得到的误差分别为：

- Training error： 1%
- Training-dev error： 9%
- Dev error： 10%

那么，由于**训练开发集**尽管和训练集来自同一分布，但是却有很大的误差， 模型无法泛化到同分布的数据，那么说明我们的模型存在**方差问题**。

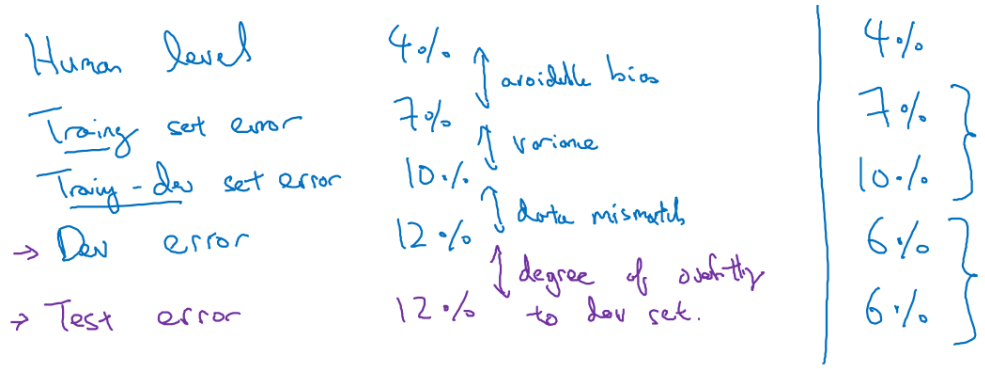
但如果我们的模型得到的误差分别为：

- Training error： 1%
- Training-dev error： 1.5%
- Dev error： 10%

那么在这样的情况下，我们可以看到，来自同分布的数据，模型的泛化能力强，而开发集的误差主要是来自于**分布不匹配**导致的。

分布不同的偏差方差分析

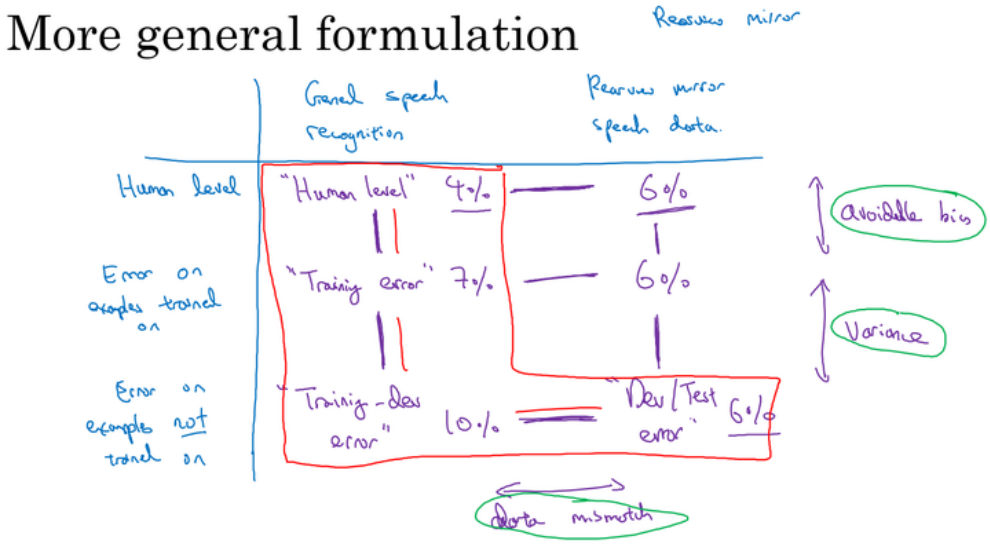
通过：Human level、Training set error、Training-dev set error、Dev error、Test error 之间误差的大小，可以分别得知我们的模型，需要依次在：可避免的偏差、方差、数据分布不匹配、开发集的或拟合程度，这些方面做改进。



误差分析

通常情况下来说，通过不同的集合上的误差分析，我们得出的结果会是中间一列误差由小变大，即误差上升的情况。但是也有一定的可能会出现右边一列误差在开发测试集上又表现的好的情况。

下面通过一个后视镜语音检测的例子来说明。我们以该例子建立更加一般的表格。



其中，横向分别是：普通语音识别数据、后视镜语音识别数据；纵向分别是：Human level、训练数据误差、未训练数据误差。表格中不同的位置分别代表不同的数据集。

通常情况下，我们分析误差会是一个递增的情况，但是对于我们的模型，在后视镜语音识别的数据数据上，可能已经可以达到人类水平误差的 6% 了，而最终的开发测试集也会 6% 的误差，要比训练误差和训练开发误差都要小。所以如果遇到这种情况，就要利用上表进行分析。

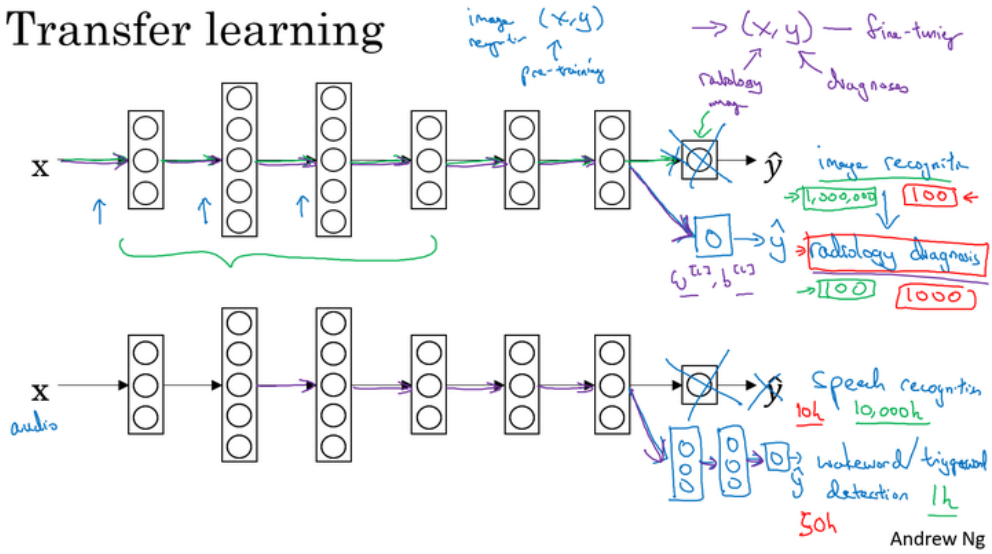
6. 解决数据分布不匹配问题

如果通过上一节的误差分析，我们可以得知，模型最终在开发和测试集上的误差最终是由于数据分布不匹配而导致。那么这样的情况下如何解决?

- 进行人工误差分析，尝试去了解训练集和开发测试集的具体差异在哪里。如：噪音等；
- 尝试把训练数据变得更像开发集，或者收集更多的类似开发集和测试集的数据，如增加噪音；

7. 迁移学习

将从一个任务中学到的知识，应用到另一个独立的任务中。



迁移学习的意义：

迁移学习适合以下场合：迁移来源问题有很多数据，但是迁移目标问题却没有那么多的数据。

假设图像识别任务中有1百万个样本，里面的数据相当多；但对与一些特定的图像识别问题，如放射科图像，也许只有一百个样本，所以对于放射学诊断问题的数据很少。所以从图像识别训练中学到的很多知识可以迁移，来帮助我们提升放射科识别任务的性能。

同样一个例子是语音识别，可能在普通的语音识别中，我们有庞大的数据量来训练模型，所以模型从中学到了很多人类声音的特征。但是对于触发字检测任务，可能我们拥有的数据量很少，所以对于这种情况下，学习人类声音特征等知识就显得相当重要。所以迁移学习可以帮助我们建立一个很好的唤醒字检测系统。

迁移学习有意义的情况：

- 任务A和任务B有着相同的输入；
- 任务A所拥有的数据要远远大于任务B（对于更有价值的任务B，任务A所拥有的数据要比B大很多）；
- 任务A的低层特征学习对任务B有一定的帮助。

8. 多任务学习

与迁移学习的串行学习方式不同，在多任务学习中，多个任务是并行进行学习的，同时希望各个任务对其他的任务均有一定的帮助。

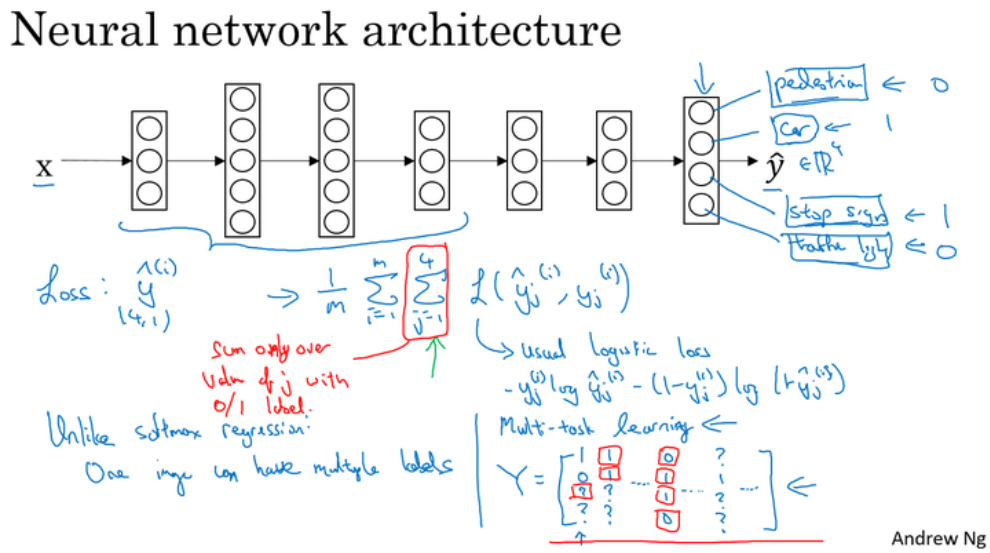
自动驾驶的例子：

假设在自动驾驶的例子中，我们需要检测的物体很多，如行人、汽车、交通灯等等。

对于现在的任务，我们的目标值变成了一个向量的形式向量中的每一个值代表检测到是否有如行人、汽车、交通灯等，一张图片有多个标签。

$$\hat{y}^{(i)} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{matrix} Pedestrians \\ Cars \\ Road\ signs - Stop \\ Traffic\ lights \end{matrix}$$

模型的神经网络结构如下图所示：



该问题的 **Loss function**：

$$loss = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{4} L(\hat{y}_j^{(i)}, y_j^{(i)}) = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{4}(y_j^{(i)}\log(\hat{y}_j^{(i)}) + (1-y_j^{(i)})\log(1-\hat{y}_j^{(i)}))$$

对于这样的问题，我们就是在做多任务学习，因为我们建立单个神经网络，来解决多个问题。

特定的对于一些问题，例如在我们的例子中，数据集中可能只标注了部分信息，如其中一张只标注了人，汽车和交通灯的标识没有标注。那么对于这样的数据集，我们依旧可以用多任务学习来训练模型。当然要注意这里Loss function求和的时候，只对带0、1标签的 **j** 进行求和。

$$Y = \begin{bmatrix} 1 & 0 & ? & ? \\ 0 & 1 & ? & 0 \\ 0 & 1 & ? & 1 \\ ? & 0 & 1 & 0 \end{bmatrix}$$

多任务学习有意义的情况：

- 如果训练的一组任务可以共用低层特征；

- 通常，对于每个任务大量的数据具有很大的相似性；（如，在迁移学习中由任务A“100万数据”迁移到任务B“1000数据”；多任务学习中，任务 A_1，...，A_n，每个任务均有1000个数据，合起来就有1000n个数据，共同帮助任务的训练）
- 可以训练一个足够大的神经网络并同时做好所有的任务。

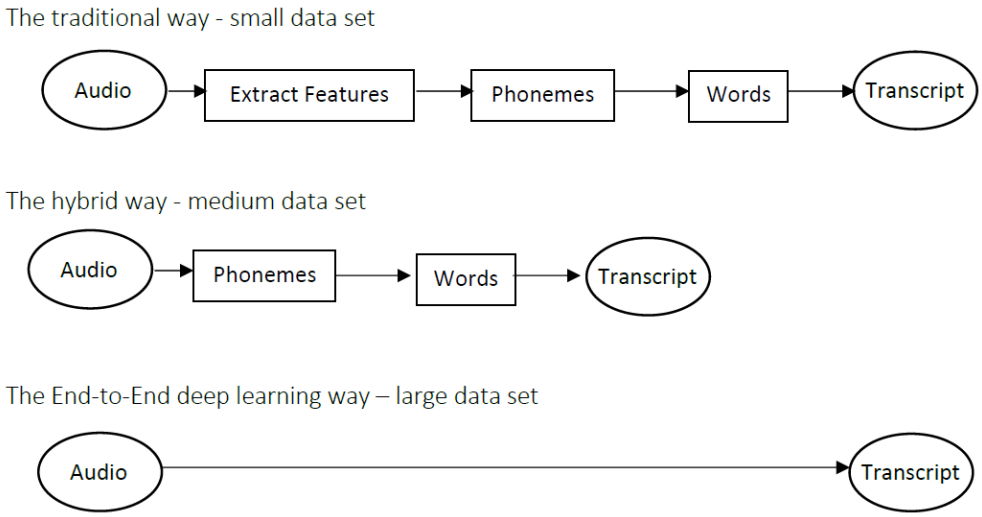
9. 端到端深度学习

端到端学习的定义：

相对于传统的一些数据处理系统或者学习系统，它们包含了多个阶段的处理过程，而端到端的深度学习则忽略了这些阶段，用单个神经网络来替代。

语音识别例子：

在少数据集的情况下传统的特征提取方式可能会取得好的效果；如果在有足够的大量数据集情况下，端到端的深度学习会发挥巨大的价值。



优缺点：

- 优点：

1. 端到端学习可以直接让数据“说话”；
2. 所需手工设计的组件更少。

- 缺点：

1. 需要大量的数据；
2. 排除了可能有用的手工设计组件。

应用端到端学习的 **Key question**：是否有足够的数据能够直接学习到从 **x** 映射到 **y** 的足够复杂的函数。

编辑于 2017-11-08 08:55

深度学习（Deep Learning）　机器学习　人工智能