结构化机器学习项目

第一周 机器学习（ML）策略 1

1.1 为什么是 ML 策略

当你构建出一个机器学习算法并利用数据加以训练后，可能结果并不能满足实际应用的需求，这时，为了提高算法准确率，可以尝试以下一些措施：① 收集更多的数据 ②增大数据集的多样性 ③延长训练时间 ④使用更大、更小的神经网络 ⑤使用 Adam 优化算法代替梯度下降法 ⑥使用 Dropout、L2 正则化等方法

在本门课中，我们将介绍如何灵活地使用这些优化算法使得神经网络更加准确。

1.2 正交化

解释正交化的概念，让我们从两个例子入手

如上图所示，对于老式的电视机，我们用很多个旋钮，分别可以调整画面的长度、宽度、旋转、压缩等；同样的，汽车有方向盘、油门、刹车，分别控制方向和速度。这种将不同属性区分开来，分别一一进行控制的方法能够让我们更加便利地处理问题。现在，设想一下，如果一台汽车的控制是这样的：一个按钮按下去加速 5%，方向右偏 1°，一个按钮按下去减速 1%，方向左偏 2°…。按照线性代数的理论，通过这些按钮我们也可以控制车的方向和速度，但实际上恐怕很少有人做得到，下面我们将给出正则化的详细定义：正则化是一种系统设计的技巧，它保证了调试一条指令或算法的组成部分不会对系统的其他组件造成附带影响。

下面我们将给出一些常见的需求以及实现方法：①拟合训练集：更大的神经网络、Adam 优化算法等 ②拟合开发集：正则化、更大的训练集 ③拟合测试集：更大的开发集 ④实际应用表现良好：更换开发集、成本函数

1.3 单一数字评估指标

对于一个模型而言，评估指标可能具有很强的多样性。拿猫分类器作为例子，评估的分类准确性的两个重要指标是：查准率（算法找出的猫中真正是猫的）、查全率（训练集中的猫被找出来的），假设我们有两个分类器，其查准率和查全率如右图所示。对于这两个指标，两个分类器各有优劣，很难做出取舍。

Classifier	Precision	Recall
A	95%	90%
B	98%	85%

这样，我们就需要一个单一化的

指标来对算法进行评估，对于上述例子，我们可以引入一个叫做 F1 分数的量，$F_1 = \frac{1}{\frac{1}{P}+\frac{1}{R}}$，即查准率和查全率的调和平均值，这样就可以利用单一的参数反映出算法的优劣。

类似地，如果有一组数据反映出该算法在不同地区的准确率，如下图

Algorithm	US	China	India	Other	Average
A	3%	7%	5%	9%	6%
B	5%	6%	5%	10%	6.5%
C	2%	3%	4%	5%	3.5%
D	5%	8%	7%	2%	5.25%
E	4%	5%	2%	4%	3.75%
F	7%	11%	8%	12%	9.5%

那么我们可以通过求出几个地区算法准确率算术平均值的方法来对不同指标进行统一。

总结一下，在我们评估一个模型时，应该尽量将多个指标进行整合，最终使用一个指标进行评估，一般可以使用各种平均值，如算术平均值、几何平均值、调和平均值、加权平均值等，这样有助于提高评估的准确率和全面性。

1.4 满足和优化指标

在很多时候，要评估一个算法的优劣，往往存在很多个指标进行衡量，这些指标中，大致可以分为两类指标：满足指标和优化指标。下面我们将以上节的猫分类器为例，详细讲解这两种指标的区别。

Classifier	Accuracy	Running time
A	90%	80ms
B	92%	95ms
C	95%	1,500ms

对于一个猫分类器，我们首先往往关注其识别的准确率，其次，我们还会关注算法的运行时间，否则一个运行一天的算法在实际应用中是没有市场的。但假如我们仍然使用上一节的方式，设参数 $F = a \times A_{CC} + b \times T_R$，即用两个指标线性组合的方式来进行衡量，这种方法显然是有缺陷的。

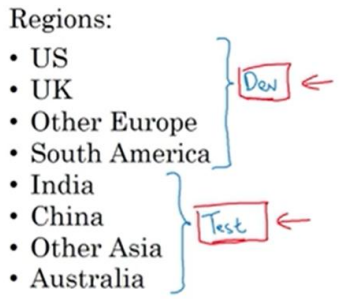
我们不妨换一种思路，对于准确率，显然是多多益善的；但对于运行时间，只要其不超过某个值，那么算法就是可以运行的。这样我们可以定义如下的评估标准：在运行时间不超过 100ms 的情况下，只看准确率的大小进行比较。

从上文的例子中可以明显看出准确率和运行时间这两个指标的区别，像准确率这种指标，往往是多多益善的，要尽量去优化，选择最优解；对于运行时间这种指标，往往只要不超过一定范围对用户体验影响不是很大。这样，我们按照两种指标的区别就可以对优化目标进行设定。对于满足指标，我们只要求其取值满足一定条件，对于优化指标，我们才会要求其越大越好或越小越好。

1.5 训练/开发/测试集的划分

在实际开发过程中，一个好的数据集划分可以很大程度地提高团队工作效率。本节将主要介绍如何进行训练/开发/测试集的划分及划分中的注意事项。

仍以上文的猫分类器为例，我们有大量的数据，这些数据分别来自美国、英国、中国等。如果我们按照如右图的方式来进行开发集、测试集的划分，到最后我们很可能会发现原本适用于开发集的算法在测试集上表现不佳。这种现象出现的原因就是开发集和测试集属于不同的分布，这可能会存在着分布的差异。



合理的做法应该是将所有的数据混合，按照一定的比例进行训练集、开发集、测试集的划分，这样可以最大程度上保证训练集、开发集、测试集服从同一分布。

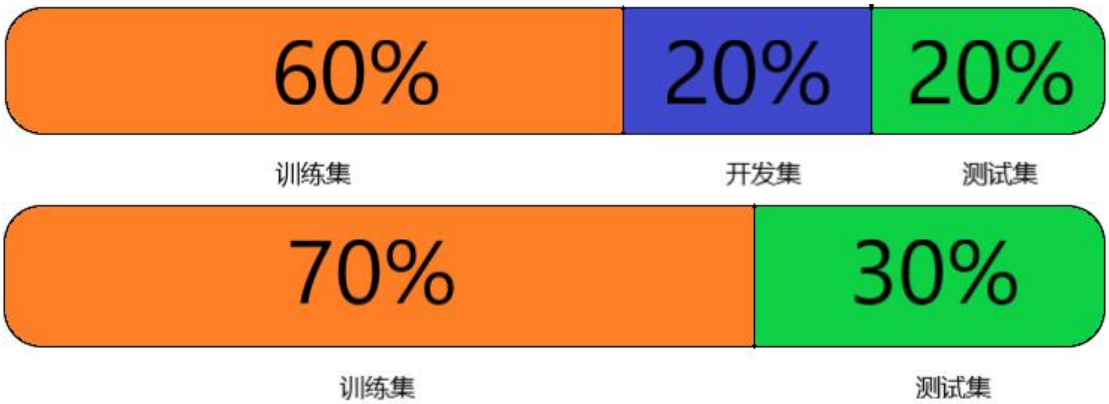
同时，在选择开发集、测试集时，一定要选择能够反应你未来可能得到数据的开发集和测试集。举个例子，一个团队在预测中收入群体的还贷能力，产品经理突然告诉你，我们用这个模型去试试低收入群体吧，如果数据选择不当，这种情况就会非常让人崩溃。

总结一下，在开发集、测试集的选择过程中，首先要确保这两个集合服从同一分布；其次，防产品经理改需求！做一个简单地类比，训练集、开发集、测试集和指标就好比是你要瞄准的目标，一定要确保目标定的准确。

1.6 开发集和测试集的大小

本节将主要介绍在实际场景中，训练集、开发集、测试集的比例选择问题。

在机器学习发展的早期，由于数据集很小，如下图的 7/3、6/2/2 划分法还是比较合理的



但随着数据集的不断增大，上述分配比例就不尽合理了，目前大多数时候百万条级别的数据采用的是 98/1/1 的比例分配训练集、开发集、测试集，更大的数据量级还要进一步压缩开发集、测试集的比例。

1.7 什么时候该改变开发/测试集和指标

本节仍以猫分类器为例，讨论什么时候去改变开发集/测试集和指标，以及具体改变的方法。

目前猫分类器有两个算法 A 和 B，评估指标是错误率，A 的错误率为 3%，B 的错误率为 5%，但 A 会把色情图片识别为猫。这种情况下，按照目前的指标，A 算法是优于 B 算法的，但无论从用户的角度还是公司的角度，A 算法是无法被接受的，这个时候，我们就需要对评估指标进行修改，可以使用如下加权的方式进

行修改。

原本评估方法：$Error = \frac{1}{n_{dev}} \sum_{i=1}^{n_{dev}} I\{\hat{y}^{(i)} \neq y^{(i)}\}$

现评估方法：$Error = \frac{1}{\sum_{i=1}^{n_{dev}} w^{(i)}} \sum_{i=1}^{n_{dev}} w^{(i)} \times I\{\hat{y}^{(i)} \neq y^{(i)}\}$，

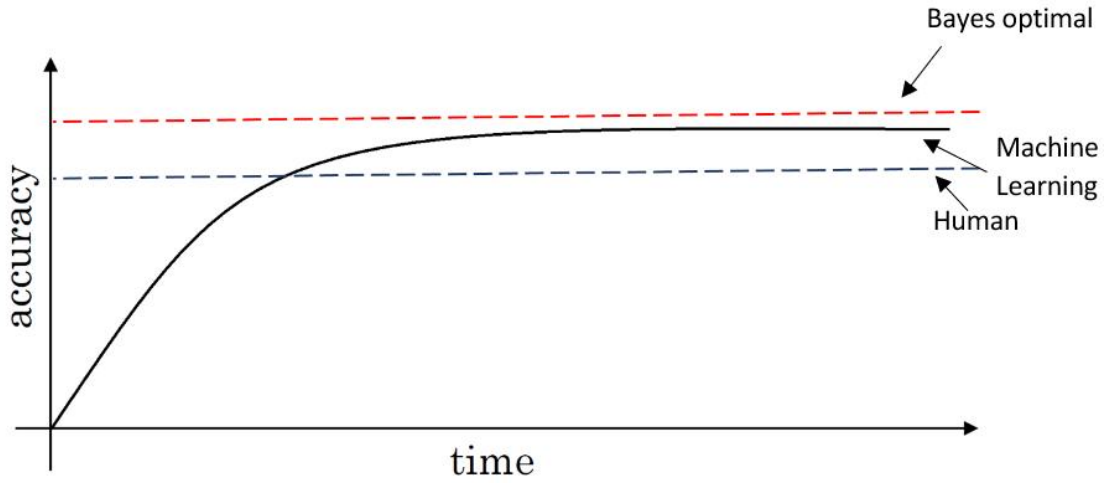
其中$w^{(i)} = \begin{cases} 1, x^{(i)}\text{是色情图片} \\ 100, x^{(i)}\text{不是色情图片} \end{cases}$

这样就可以有效地在指标中表现出来色情图片对结果的影响。

在开发过程中，我们还可能遇到下面的情况，算法 A 在开发集上运行效果由于算法 B，但实际应用上，算法 B 的效果要明显优于算法 A。这种情况出现的原因就是开发集的选取不当。还是以猫分类器为例，如果开发集中的图片来源于网上取景专业、清晰的照片，但实际上，往往拍出来的照片是模糊的、不完整的，这样就造成了上述情况的发生。遭遇这种情况时，我们就往往需要改变开发集的选取。

1.8 为什么是人的表现

目前很多机器学习团队都会比较机器学习算法与人类表现，这种现象出现的原因主要有以下两点：第一，在机器学习许多应用领域，机器学习算法已经开始超越人类的表现；第二，在让机器做的做的事时，往往机器的表现会好于人类，效率也更高。下面用一个例子说明这个问题。



对于机器学习算法，以时间为横轴，准确率为纵轴。当机器学习算法的性能低于人类时，其算法性能提升速度往往十分迅速；当机器学习算法性能高于人类时，算法性能提升速度往往就十分缓慢。这种现象出现的原因主要有以下几点：第一，很多问题上人类的准确率已经几乎接近贝叶斯偏差（如识别图像中是否有猫），机器学习算法的上升空间十分有限 ；第二，当机器学习算法准确率低于人类时，一方面算法可以得到人工标记的数据进行训练，另一方面人类也可以通过分析为什么人能够做对而机器做错的方法来提高机器学习算法的准确率 ；第三，机器学习算法准确率低于人类时，可以获得更好的偏差与方差。

1.9 可避免误差

为了解释可避免误差这一概念，我们还是以猫分类器为例，下图为两种情况下的人类偏差、训练集偏差、开发集偏差

	Classification error (%)	
	Scenario A	Scenario B
Humans	1	7.5
Training error	8	8
Development error	10	10

由于人类在图像识别、语音识别等领域具有很好的表现，可以近似地认为人类偏差即为贝叶斯偏差，即“不可避免”的偏差。我们将训练集偏差与人类偏差的差值称为可避免误差，下面的介绍将会帮助你直观理解可避免误差。

如上图所示，状态 A 的可避免误差为 7%，方差指标为 2%，这个时候，对于算法的改进应该更倾向于去减小可避免误差，这样算法性能会有较大幅度提升。相反，状态 B 可避免误差为 0.5%，方差指标为 2%，这种状态下设法降低方差就比设法降低偏差更加容易。

1.10 理解人的表现

在深度学习相关论文中，“人类水平误差”一次经常被提到，本节中我们将给出“人类水平误差”的定义。

Medical image classification example:

Suppose:

(a) Typical human ..... 3 % error

(b) Typical doctor ..... 1 % error

(c) Experienced doctor ..... 0.7 % error

(d) Team of experienced doctors .. 0.5 % error

对于上图，对于识别 X 射线影像，普通人的误差率为 3%，普通医生误差率为 1%，有经验的医生误差率为 0.7%，有经验医生团队误差率为 0.5%。上述误差从字面上都可以称之为“人类水平误差”，但应用中“人类水平误差”是要用来估计贝叶斯误差的，所以说根据定义，贝叶斯误差 $Bayers \leq 0.5\%$，所以说，人类水平误差应该定义为人类中误差的最低值。

但由于应用场景的不同，有时人类水平误差也可以有其他的定义，比如上述例子中，仅要求算法性能超过不同医生即可，或者由于某种理由，只要超过正常人就可以满足需求。只需要注意一点，人类水平误差的定义根本上是取决于需求的。

1.11 超过人的表现

如右图所示，状态 A 中，机器学习

	Classification error (%)	
	Scenario A	Scenario B
Team of humans	0.5	0.5
One human	1.0	1
Training error	0.6	0.3
Development error	0.8	0.4

算法已经超过了单人误差率，但没有超过多人误差率，这种情况下，机器学习算法偏差为 0.1%、方差为 0.2%，所以应当采取减小方差的措施。但状态 B 中，机器学习算法已经超过了人类最低的误差率，现在根据已有数据根本无法判断机器学习算法是过拟合还是确实超过了人的表现，也无法判断真正的贝叶斯误差，这种情况下之前已知调整偏差、方差的工具就都失效了，机器学习算法的改进也就难以进行。

但在很多使用结构化数据的领域，机器学习算法已经超过人类，例如：广告推荐、产品推荐、逻辑回归、是否批准贷款等。同时，在一些使用非结构化数据的领域，比如：语音识别。

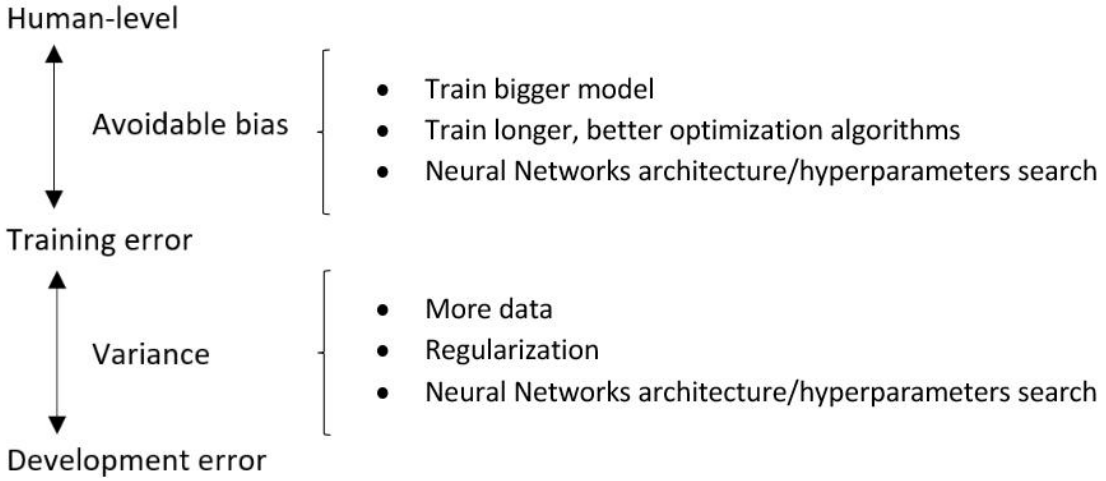
1.12 改善模型表现

为了改善模型表现，我们根据以下思路进行调试。

首先，对比训练集误差和人类水平误差（贝叶斯误差），如果该误差较大，可以尝试训练更大神经网络、延长训练时间、使用 RMSprop、Adam 等优化算法、更换神经网络结构等降低偏差的方法。

然后，当偏差处于可以接受范围时，对比训练集误差和开发集误差，如果该误差较大，可以尝试提供更多数据、正则化、数据增强等降低方差的方法。

直观流程图如下图所示



第二周 机器学习（ML）策略 2

2.1 进行误差分析

在训练神经网络过程中，我们往往会遇到偏差较大的情况，解决问题的具体

方法也是多样的，但有些方法会十分耗时，并且效果未知。本小节将主要讲述如何选择神经网络优化方法及其论证思路。

仍以猫分类器为例，假如编写的猫分类器在开发集的误差率为 10%，这个时候，有人发现很多错误来源于分类器将部分狗识别为了猫。初步的想法就是编写一套方法，专门将猫与狗加以区分，但这套方法将会耗时很长。下面我们将会介绍一套方法用以检验十分要去执行这个方法。

首先，我们随机选择 100 个错误标记的开发集例子进行分析，统计其中狗图占总量的比例。假如狗图仅占 5%，那么表明，即使使用上述方法能够完全解决猫狗区分的问题，算法的准确率最多只能提升到 9.5%，这种收益与投入显然是不成正比的。但假如狗图占了超过 50%，这个时候使用上述方法对算法进行改善的收益就十分可观了，可以进行一定的尝试。

一般地，我们对于一个问题往往有很多想法，这时我们可以利用如下表格进行误差分析。

Image	Dog	Great Cats	Blurry	Instagram	Comments
1	✓				Pitbull
2			✓		
3		✓	✓		Rainy day at zoo
⋮	⋮	⋮	⋮		
% of total	8%	43%	61%		

还是针对上述问题，我们可能会想到是狗图对算法造成了影响，还会想到其他猫科动物、图片清晰度，以及滤镜对算法造成影响。这时，对于多种可能性，我们可以在上图表格中，同步地分析这些想法的可行性。首先在表格中标明错误类型、备注、以及占总数百分比。然后取多个错误标记的开发集样本，人工区分错误类型，并在上图中进行记录。以上图为例，狗图的比重仅占 8%，这样的比例相对于其他猫科动物的 43%和图片模糊的 61%就显得不那么重要了，所以说对于上述问题，我们应先对其他猫科动物干扰和图片模糊的问题进行修正。

通过上述方法，我们可以排列出问题的优先级并得到解决各类问题所带来算法性能提升的最大值，有助于我们对神经网络的调试。

2.2 清楚标注错误的数据

在训练神经网络时，训练集、开发集、测试集中会不可避免地出现坏数据，如下图所示，第六张图将白色的狗标记为了猫。

实际上，神经网络的鲁棒性对于这种随机误差或者近似随机误差具有较强的鲁棒性，像上图中偶尔出现的错误标记敏感度不大。但是神经网络对于系统性误差的鲁棒性不是很好，假如训练集中的白狗均被标注为猫，神经网络的输出结果就会出现很大的问题。下面的内容将会讨论对于各种情况的随机误差是否进行修

正以及修正的注意事项。

对于随机误差，我们可以采用上节中的图标方法进行分析

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	8%	43%	61%	6%	

如果错误标注的数据在错误标记样本中占有较大比例，那么就需要去花时间一一校对，但如果比例不大，那么在对算法性能影响不大的情况下完全可以忽略错误标记。

注意：①上文中所说的错误标注数据占比指的是错误标注数据在所有错误输出数据中所占的比例，0.6%在 2%中的占比也是很大的！ ②在修正标签过程中，一定要同时对开发集和测试集进行修正，确保二者服从同一分布 ③在修正错误输出的标签时也应当注意正确输出的标签，因为有的时候算法可能是将错误标签“错误地”输出为正确的结果（但一般情况下不常用，原因是 2%的错误数据很容易去校对，但 98%的正确数据校对起来却很难） ④由于神经网络算法对随机误差具有较强的鲁棒性，所以可以只修正开发集和测试集中的标签，训练集标签可以不进行修正

2.3 快速搭建你的第一个系统，并进行迭代

对于机器学习系统的开发，最好的方式就是先快速搭建第一个系统，然后对此系统进行迭代，下面我们以例子说明这个过程。

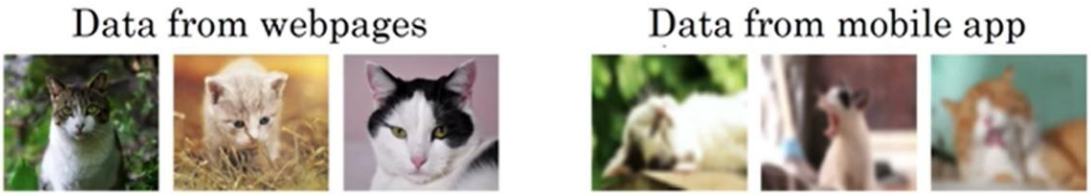
以语音识别系统为例，在建立系统时，有很多种增强技术供我们考虑，例如：背景噪音、方言演讲、远场通话、儿童讲话、口吃等。但是多种技术也会使我们不知道最开始进行优化的方向，这种情况，我们可以按照如下的步骤进行分析。首先，我们应该快速设定开发集、测试集以及指标，这样就设定了一个目标和评价准则。然后，快速搭建一个最初版本的机器学习系统，看一看最初的算法在开发集、测试集、评估指标上表现如何。利用前几门课中讲过的偏差、方差和错误分析等方法对算法进行优化。举个例子，如果偏差分析表明，大部分偏差来自于说话者远离麦克风，那么我们就可以利用相应的技术对其进行优化。

总结一下，对于开发机器学习系统，我们最初不应想的过于复杂，应当先做出一个较为初级、简单的系统，然后在此基础上根据需求进行迭代，进而开发出一个较为复杂、可靠的系统。

2.4 在不同的划分上进行训练并测试

由于机器学习算法对于数据的需求量很大，很多团队在收集数据时往往会使开发集和测试集的数据来自不同的分布，这种情况很普遍，下面我们将介绍一些方法来训练集和测试集存在差异的情况。

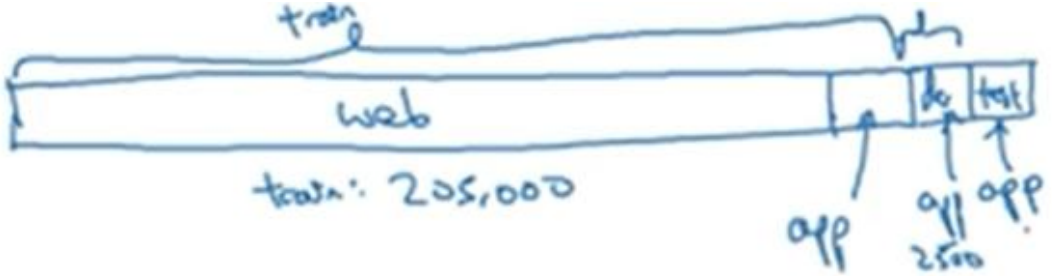
仍以猫分类器为例



在构建开发集使，我们可以使用爬虫在互联网上下载很多猫图片，假设有200000 张，这些图片取景专业、画质较高，如左图；但在实际应用中，我们得到的猫图片，假设有 10000 张，而这些图片往往是非专业、画质模糊的，如右图。

对于这种情况，我们有　　种选择

第一种选择如上图，我们将这 210000 张图片按比例随机分为训练集、开发集、测试集，但这种方法存在一个较大的问题，在开发集和测试集中，绝大部分图片是来自于网络下载，只有很少一部分是真正用户上传的图片类型。这样很大程度上，我们优化的方向就是网络下载图片，这样可能会对实际应用造成影响，不推荐使用这种划分方式。

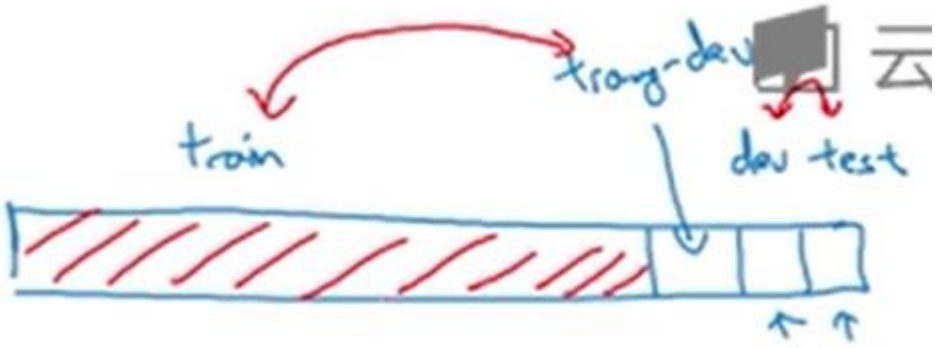


第二种选择如上图，我们先将 200000 张网络下载图片和 5000 张手机拍摄图片组成训练集，再将剩余的 5000 张手机拍摄图片组成开发集和测试集。这样做有一定的问题，因为训练集和开发集、测试集来自不同的分布，但事实证明，这样做在长期上可以为系统带来更好的性能。

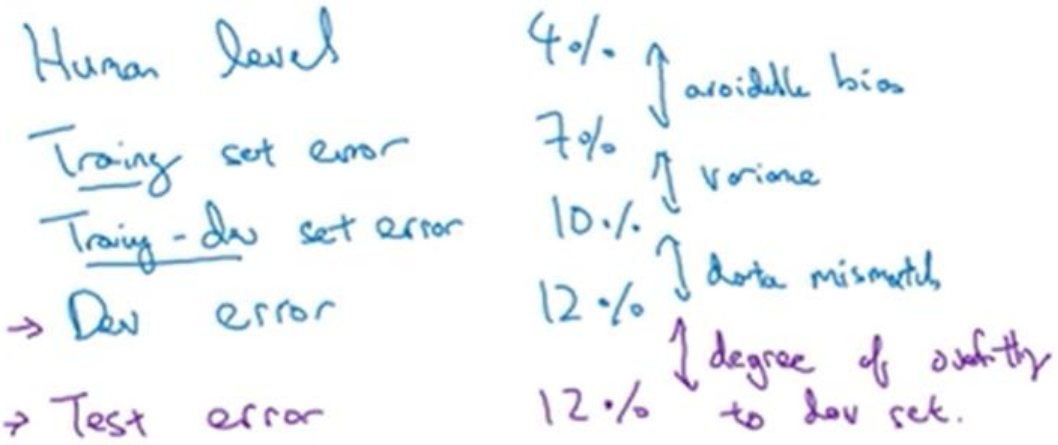
2.5 不匹配数据划分的偏差和方差

很多时候我们的开发集和测试集数据往往是来自不同的分布的，这时对偏差和方差的估计就要做一定的调整。

仍以猫分类器为例，假如训练集误差率为 1%，开发集误差率为 9%。由于训练集和开发集的分布不同，我们很难说明训练集与开发集误差率之差是由于分布不同造成的还是由于方差较大造成的，下面我们将介绍一种方法来进行区分。



如上图所示，我们可以将训练集中的一部分拿出来，作为 training-dev 集，这个集合与训练集来自于同一分布。这样，如下图所示，我们可以通过几个集合误差率的横向对比，来反应算法的各个指标。



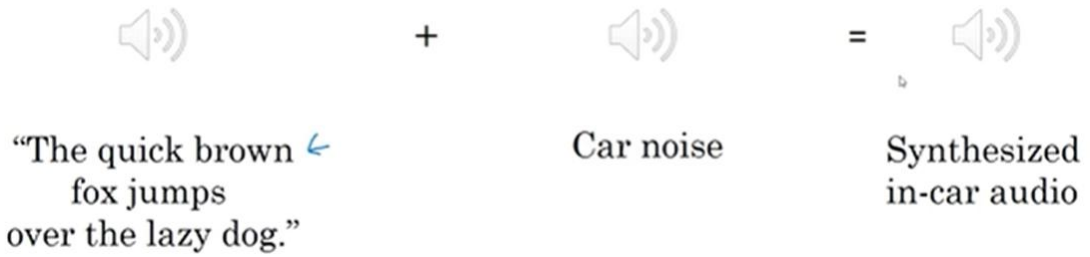
训练集与人类水平的误差率之差，代表着可避免误差；训练-开发集与训练集的误差率之差，代表着方差；开发集与训练-开发集误差率之差，代表着数据不匹配造成的误差；测试集与开发集的误差率之差，代表着过拟合程度。

2.6 定位数据不匹配

在实际编程过程中，我们常常会遇到数据不匹配问题，对于这类问题，我们往往没有系统的解决方法，本节将主要介绍几种常见可供尝试的方法。

首先，可以进行人工误差分析，尝试理解训练集与开发集的差异。然后可以设法使得训练集数据与开发集数据更相似，或者收集更多与开发集相似的数据。

以语音识别为例，很多时候实际环境下会存在很多的噪音，所以我们的开发集必定会带有一定的噪音，为了降低数据不匹配造成的误差，我们可以通过收集



更多的数据，使得训练集数据与开发集数据更相似。但事实上，我们很难去收集一些特定条件的数据，比如在汽车噪音条件下的对话。这时，我们就需要利用语

音合成技术来进行，如上图所示，我们可以通过将对话音频与汽车噪音音频合成的方法得到汽车噪音背景下的对话。

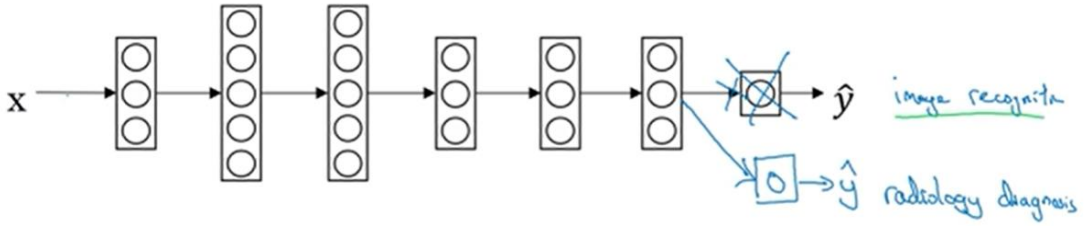
但注意到，这种方法会带来一定的问题，在实际情况下，假如对话音频时间的 10000 小时，但汽车背景噪音音频只有 1 小时，这样极有可能会造成机器学习算法对这 1 小时噪音的过拟合。

总结一下，对于数据不匹配问题，我们可以考虑人工分析一下训练集与开发集数据的不同之处，进而做出相应的修正方案。收集或者合成更多的数据是一个很好的解决方案，但在人工合成数据的过程中要注意过拟合现象的产生。

2.7 迁移学习

深度学习最神奇之处就是在已训练的神经网络可以“利用”已学到的知识来解决其他相近领域的问题，本节将介绍这种被称为迁移学习的现象。

仍以猫分类器为例，我们有一个已经训练好的猫分类器，但现在我们需要训练一个能够识别 X 射线照片的神经网络，这个时候，如果数据集较小，我们往往可以直接将猫分类器是输出层权重随机初始化，使用 X 射线照片的训练集进行训练，如下图所示。



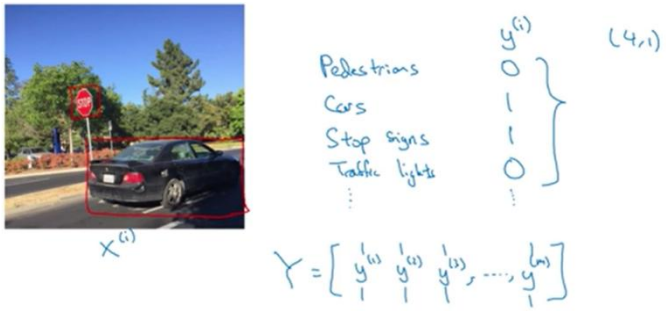
这种方法可行的原因就是对于很多低层次特征，如边缘检测、曲线检测、对象阳性检测等，通过较大的数据库习得的这些知识会对识别 X 射线照片有很大的帮助。

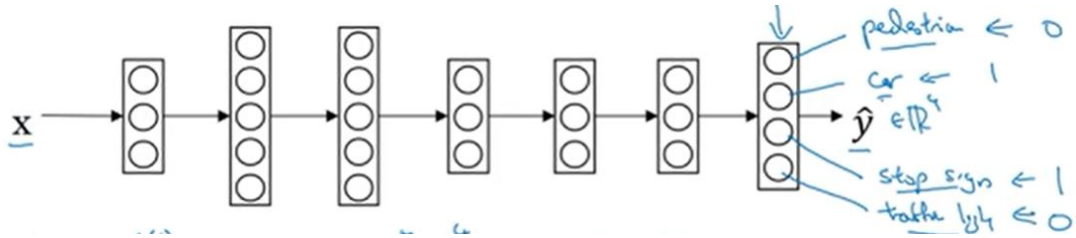
迁移学习应用的场景主要是我们拥有很多关于迁移来源问题的数据，但关于迁移目标问题却没有那么多数据的时候。但注意到，迁移来源问题与迁移目标问题处理的数据必须是同一类型的，例子中的数据就都是图片。

2.8 多任务学习

对于一个深度学习系统，我们往往需要识别很多相似但却不同的物体，比如在自动驾驶中，我们需要识别车、行人、路障、标识牌等，如果对于每个物体分别进行训练，这样神经网络的效率就会低很多。多任务学习就是将很多相似输入放在一起进行学习，这样很大程度上提供了神经网络算法的效率。下面我们将以无人驾驶为例介绍多任务学习。

如右图所示，不妨设我们只需要识别其中四种物体，行人、车、停止标识、信号灯。我们可以将输出 Y 变为一个 4×1 列向量，这样就可以只遍历一次训练集就可以训练神经网络识别所有物体的能力。





我们可以通过构建如上的神经网络来进行多任务学习，相应地，成本函数为

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{4} L\left(\hat{y}_j^{(i)}, y_j^{(i)}\right)$$

多任务学习适用条件：①训练的一组任务共用低层次特征 ②每个任务的数据量类似 ③能够训练一个足够大的神经网络来做好每个任务

总结：多任务学习具有一定的局限性，就适用条件③而言，训练一个大的完成四个任务的神经网络不会比训练四个四个相对小完成单个任务的神经网络效率高很多。大多数时候多任务学习不如迁移学习使用普遍。但在计算机视觉领域，利用多任务学习的方法，训练一个大的神经网络来识别各种物体的效率确实比较高。