

Machine Learning Yearning is a



deeplearning.ai project.

© 2018 Andrew Ng. All Rights Reserved.

*Page 2* *Machine Learning Yearning-Draft* *Andrew Ng*

23 处理偏差和方差

以下是处理偏差和方差问题最简单的公式:

• 如果具有较高的可避免偏差，那么增加模型的大小（如：增加神经网络的隐藏层或者神经元）

• 如果是高方差，那么增加训练集。

如果你可以增加神经网络的大小，并且可以无限制的增加数据集，那么你可以在很多机器学习问题上都做得很好。

在实践中，增加神经网络的大小会导致你遇到计算上的问题，因为神经网络规模越大，训练的就越慢了，你也许会竭尽全力去寻找训练集，但是网络上的猫咪图片是有限的。

不同的神经网络架构对于你的问题将会有不同的偏差和方差。最近很多深度学习研究者已经开发出了很多新的神经网络架构。所以，如果你在使用神经网络的时候，学术文献是一个很不错的灵感来源。Github上也有很多很好的开源实现。但是尝试新的神经网络架构的结果要比增加神经网络架构和增加数据集更难预测。

增加模型的大小可以减少偏差，但也可能会增加偏差和方差过拟合的风险。然而，这种过拟合问题通常只会在你没有进行正则化的时候出现，如果你使用了一个非常好的正则化方法，那么你通常可以安全的增加模型大小，也不会增加过拟合的风险。

假设你正在使用深度学习，有L2正则化和dropout，有在开发集上表现最好的正则化参数。如果你增加模型的大小，通常你的表现会保持不变或提升的不太明显。避免使用更大的模型的原因就是为了避免太大的计算代价。

*Page 3* *Machine Learning Yearning-Draft* *Andrew Ng*

24 偏差和方差之间的权衡

你可能听说过“偏差和方差之间的权衡”。在你对大部分学习算法进行修改的时候，有些方法可以减少偏差，但是代价是增加了方差，反之亦然，这就在偏差和方差之间产生了“权衡”。

例如，增加模型的大小（在神经网络中添加神经元/层，或增加输入特征），通常可以减少偏差，但可能会增加方差。另外，增加正则化一般会增加偏差，但是可能会减少方差。

在现代，我们往往能够获得充足的数据，并且可以使用非常大的神经网络（深度学习）。因此，这种权衡比较少，并且现在有更多的选择可以在不增加方差的情况下减少偏差，反之亦然。

例如，你通常可以增加神经网络的大小，并调整正则化方法去减少偏差，而不会明显的增加方差，通过增加训练集，你也可以做到在不影响偏差的情况下减少方差。

如果你选择了一个比较适合你任务的模型，那么你可以同时减少偏差和方差，但是选择适合的架构是非常难的。

在接下来几章中，我们将讨论处理偏差和方差的其它方法。

*Page 4* *Machine Learning Yearning-Draft* *Andrew Ng*

25 减少可避免偏差的方法

如果你的学习算法是高可避免偏差的话，你可以尝试以下办法:

•**增加模型大小** ​(如神经元和层数): 该方法可以减少偏差，因为它可以让你更好的适应训练集。如果你发现该方法增加了方差，那么使用正则化方法，它通常可以消除方差的增加。

•**基于错误分析修改输入特征**​: 假设错误分析启发你去创建额外的特征，以帮助算法消除特定类别的错误。（我们将在下一章进一步讨论）这些新特征可能有助于减少偏差和方差。理论上来说，增加更多的特征可能会增加方差，如果你发现方差增加了，那么使用正则化的方法，它通常可以消除方差的增加。

•**减少或消除正则化**​： (L2 正则化, L1 正则化, dropout): 这将减少可避免的偏差，但会增加方差。

•**修改模型架构：**​ (如神经网络架构) 以便算法更加适用于你的问题：这种方法可能会同时影响偏差和方差。

下面的方法是没有用的:

•**添加更多的训练数据**​: 这种方法有助于减少方差问题，但是它通常对偏差没有显著的影响。

*Page 5* *Machine Learning Yearning-Draft* *Andrew Ng*

26 训练集误差分析

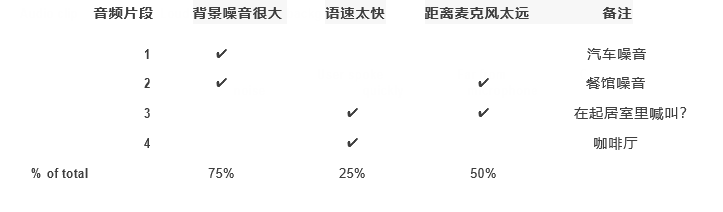
**noise** **User spoke**

**quickly** **Far from**

**microphone** **Comments**

你的算法必须在训练集上表现得很好，才能期望它在开发集和测试集上能够有着良好的表现。除了先前提到的用于处理高偏差的技术外，我通常也会在训练数据上进行误差分析，处理方式类似于在开发集上设置一个 Eyeball 开发集。当你的算法有着高偏差时（例如算法没有很好拟合训练集的时候）这将有所帮助。

举个例子，假设你正在为一个应用程序构建一个语音识别系统，并收集了一组志愿者的音频片段。如果系统在训练集上表现不佳，你可能会考虑以约 100 个算法处理得很差的样本作为一组并人为去听它们，从而知道训练集误差的主要种类。类似于开发集上的误差分析，你可以计算不同类别的错误样本数量：



在本例中，你可能会发现算法在面对具有大量背景噪音的训练样本时遇到了特别困难的情况。因此你可能会关注一些技术，使其能够更好地适应背景噪音的训练样本。

你也可以仔细检查正常人是否能转录这些音频片段，这些音频应该与你的学习算法的输入音频相同。如果背景噪音过于嘈杂，导致任何人都不能理解音频里说了什么，那么期望算法正确地识别这样的话语就不太合理。我们将在后面的章节中讨论将算法的性能与人类水平进行比较的好处。

*Page 6* *Machine Learning Yearning-Draft* *Andrew Ng*

27 减少方差的技术

如果你的学习算法存在着高方差，则可以尝试下面的技术：

•**添加更多的训练数据**：这是最简单也是最可靠的一种方式来处理方差，只要你能访问大量的数据并有足够的计算能力来处理它们。

**• 加入正则化（L2 正则化，L1 正则化，dropout）：**这项技术可以降低方差，但却增大了偏差。

加入提前终止（比如根据开发集误差提前终止梯度下降）：这项技术可以降低方差但却增大了偏差。提前终止（Early stopping）有点像正则化理论，一些学者认为它是正则化技术之一。

**• 通过特征选择减少输入特征的数量和种类：**这种技术可能有助于解决方差问题，但也可能增加偏差。稍微减少特征的数量（比如从 1000 个特征减少到 900 个）不太可能会对偏差产生很大的影响，但显著地减少它们（比如从 1000 个特征减少到 100 个，10 倍地降低）则很有可能产生很大的影响，你可能排除了太多有用的特征。在现代深度学习中，当数据充足时，特征选择的比重就有所改变，现在我们更有可能将拥有的所有特征提供给算法，并让算法根据数据来确定哪些特征可以使用。但当你的训练集很小的时候，特征选择是非常有用的。

**• 减小模型规模（比如神经元/层的数量）：**谨慎使用。这种技术可以减少方差，同时可能增加偏差。然而我不推荐这种处理方差的方法，添加正则化通常会提供更好的分类性能。 减少模型规模的好处是降低了计算成本，从而加快了你对模型进行训练的速度。如果加速模型训练是有用的，那么无论如何都要考虑减少模型的规模。但如果你的目标是减少方差，并且不关心计算成本，那么考虑添加正则化会更好。

下面是两种额外的策略，和解决偏差问题章节所提到的方法重复：

**根据误差分析结果修改输入特征：**假设误差分析的结果鼓励你创建额外的特征，从而帮助算法消除某个特定类别的误差。这些新的特征对处理偏差和方差都有所帮助。理论上，添加更多的特征将增大方差；然而当你发现这种情况时，加入正则化，这可以消除方差的增加。

修改模型架构（比如神经网络架构）使之更适用于你的问题：这项技术将同时影响偏

差和方差。

*Page 7* *Machine Learning Yearning-Draft* *Andrew Ng*