

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 Error analysis on the training set

**noise** **User spoke**

**quickly** **Far from**

**microphone** **Comments**



Your algorithm must perform well on the training set before you can expect it to perform

well on the dev/test sets.

In addition to the techniques described earlier to address high bias, I sometimes also carry

out an error analysis on the ​*training data*​, following a protocol similar to error analysis on

the Eyeball dev set. This can be useful if your algorithm has high bias—i.e., if it is not fitting

the training set well.

For example, suppose you are building a speech recognition system for an app and have

collected a training set of audio clips from volunteers. If your system is not doing well on the

training set, you might consider listening to a set of ~100 examples that the algorithm is

doing poorly on to understand the major categories of training set errors. Similar to the dev

set error analysis, you can count the errors in different categories:

**Audio clip** **Loud background**

**1** ✔  Car noise

**2** ✔  ✔  Restaurant noise

**3** ✔  ✔  User shouting

across living room?

**4** ✔     Coffeeshop

**% of total** 75%  25%  50%

In this example, you might realize that your algorithm is having a particularly hard time with

training examples that have a lot of background noise. Thus, you might focus on techniques

that allow it to better fit training examples with background noise.

You might also double-check whether it is possible for a person to transcribe these audio

clips, given the same input audio as your learning algorithm. If there is so much background

noise that it is simply impossible for anyone to make out what was said, then it might be

unreasonable to expect any algorithm to correctly recognize such utterances. We will discuss

the benefits of comparing your algorithm to human-level performance in a later section.

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

27 Techniques for reducing variance

If your learning algorithm suffers from high variance, you might try the following

techniques:

•**Add more training data**​: This is the simplest and most reliable way to address variance,

so long as you have access to significantly more data and enough computational power to

process the data.

•**Add regularization**​ (L2 regularization, L1 regularization, dropout): This technique

reduces variance but increases bias.

•**Add early stopping**​ (i.e., stop gradient descent early, based on dev set error): This

technique reduces variance but increases bias. Early stopping behaves a lot like

regularization methods, and some authors call it a regularization technique.

•**Feature selection to decrease number/type of input features:**​ This technique

might help with variance problems, but it might also increase bias. Reducing the number

of features slightly (say going from 1,000 features to 900) is unlikely to have a huge effect

on bias. Reducing it significantly (say going from 1,000 features to 100—a 10x reduction)

is more likely to have a significant effect, so long as you are not excluding too many useful

features. In modern deep learning, when data is plentiful, there has been a shift away from

feature selection, and we are now more likely to give all the features we have to the

algorithm and let the algorithm sort out which ones to use based on the data. But when

your training set is small, feature selection can be very useful.

•**Decrease the model size** ​(such as number of neurons/layers): ​*Use with caution.*​ This

technique could decrease variance, while possibly increasing bias. However, I don’t

recommend this technique for addressing variance. Adding regularization usually gives

better classification performance. The advantage of reducing the model size is reducing

your computational cost and thus speeding up how quickly you can train models. If

speeding up model training is useful, then by all means consider decreasing the model size.

But if your goal is to reduce variance, and you are not concerned about the computational

cost, consider adding regularization instead.

Here are two additional tactics, repeated from the previous chapter on addressing bias:

•**Modify input features based on insights from error analysis**​: Say your error

analysis inspires you to create additional features that help the algorithm to eliminate a

particular category of errors. These new features could help with both bias and variance. In

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

theory, adding more features could increase the variance; but if you find this to be the case,

then use regularization, which will usually eliminate the increase in variance.

•**Modify model architecture**​ (such as neural network architecture) so that it is more

suitable for your problem: This technique can affect both bias and variance.

*Page 8* *Machine Learning Yearning-Draft* *Andrew Ng*