

Machine Learning Yearning is a



deeplearning.ai project.

© 2018 Andrew Ng. All Rights Reserved.

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

偏差和方差

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

20 偏差和方差

假设你的训练集，开发集和测试集都来自同一分布。那么你会觉得获取更多的训练数据就可以提高性能，对吗？

尽管更多的数据是无害的，但它并不是总会像我们所期望的那样有用。获取更多的数据需要耗费很多时间。所以，你需要什么什么时候该增加数据量，什么时候不该增加。

机器学习中有两个主要的错误来源：偏差和方差。理解它们有助于你觉得是否添加数据，以及其它提高性能的策略，这将会很好的利用你的时间。

假设你正在构建一个错误率为5%的猫咪识别器。目前，你的训练集错误率为15%，并且你的开发集错误率为16%，在这种情况下，添加数据可能不会有太大的帮助。你应该关注其它的办法。实际上，在你的训练集上添加更多的样本只会让你的算法难以在训练集上做的更好。（后面的章节我会解释原因）

如果你在训练集上的错误率为15%（85%的准确率），但是你的目标是5%的错误率（95%的准确率），那么第一个要解决的问题是提高算法在训练集上的性能。你的开发/测试集上的性能通常比训练集差。所以，如果算法在见过的样本上得到了85%的准确率，那么是不可能在没见过的样本上得到95%的准确率的。

假设如上述你的算法在开发集上有16%的错误率（84%的准确率）。我们将这16%的错误分为两部分：

* 首先，算法在训练集上的错误率。在本例中，它是15%。我们非正式的认为这是算法的**偏差(bias)。**
* 其次，算法在开发（或测试）集上比训练集差多少。在本例中，开发集比训练集差1%。我们非正式的认为这是算法的**方差(Variance)[1]。**

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

学习算法的一些改变能解决错误的第一个组成部分——偏差，并且提高算法在训练集上的性能；一些改变能解决第二个组成部分——方差，并帮助算法从训练集到开发/测试集上得到更好的泛化[2] 。

为了选择最有希望的改变，了解这两组错误中哪个更值得去解决是非常有用的。

培养你对于偏差和方差的感觉可以帮你在优化算法上有非常大的帮助。

1 统计领域有更多关于偏差和方差的正式定义，我们不必担心。粗略地说，当你有一个非常大的训练集时，偏差就是你算法在训练集上的错误率。方差是与此设置中的训练集相比，你在测试集上差多少。当你的误差衡量是均方差（mean squared error）时，你可以写下指定这两个量的公式，然后证明：Total Error = Bais + Variance。但是为了决定如何在ML问题上取得进展，这里了解偏差与方差即可。

2 这里还有一些通过对系统架构做出大的改变的方法，能够同时减少偏差和方差。但是这些方法做起来比较难。

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

21 Examples of Bias and Variance

Consider our cat classification task. An “ideal” classifier (such as a human) might achieve

nearly perfect performance in this task.

Suppose your algorithm performs as follows:

• Training error = 1%

• Dev error = 11%

What problem does it have? Applying the definitions from the previous chapter, we estimate

the bias as 1%, and the variance as 10% (=11%-1%). Thus, it has ​**high variance**​. The

classifier has very low training error, but it is failing to generalize to the dev set. This is also

called ​**overfitting**​.

Now consider this:

• Training error = 15%

• Dev error = 16%

We estimate the bias as 15%, and variance as 1%. This classifier is fitting the training set

poorly with 15% error, but its error on the dev set is barely higher than the training error.

This classifier therefore has ​**high bias**​, but low variance. We say that this algorithm is

**underfitting**​.

Now, consider this:

• Training error = 15%

• Dev error = 30%

We estimate the bias as 15%, and variance as 15%. This classifier has ​**high bias and high**

**variance**​: It is doing poorly on the training set, and therefore has high bias, and its

performance on the dev set is even worse, so it also has high variance. The

overfitting/underfitting terminology is hard to apply here since the classifier is

simultaneously overfitting and underfitting.

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

Finally, consider this:

• Training error = 0.5%

• Dev error = 1%

This classifier is doing well, as it has low bias and low variance. Congratulations on achieving

this great performance!

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

22 Comparing to the optimal error rate

In our cat recognition example, the “ideal” error rate—that is, one achievable by an “optimal”

classifier—is nearly 0%. A human looking at a picture would be able to recognize if it

contains a cat almost all the time; thus, we can hope for a machine that would do just as well.

Other problems are harder. For example, suppose that you are building a speech recognition

system, and find that 14% of the audio clips have so much background noise or are so

unintelligible that even a human cannot recognize what was said. In this case, even the most

“optimal” speech recognition system might have error around 14%.

Suppose that on this speech recognition problem, your algorithm achieves:

• Training error = 15%

• Dev error = 30%

The training set performance is already close to the optimal error rate of 14%. Thus, there is

not much room for improvement in terms of bias or in terms of training set performance.

However, this algorithm is not generalizing well to the dev set; thus there is ample room for

improvement in the errors due to variance.

This example is similar to the third example from the previous chapter, which also had a

training error of 15% and dev error of 30%. If the optimal error rate is ~0%, then a training

error of 15% leaves much room for improvement. This suggests bias-reducing changes might

be fruitful. But if the optimal error rate is 14%, then the same training set performance tells

us that there’s little room for improvement in the classifier’s bias.

For problems where the optimal error rate is far from zero, here​’​s a more detailed

breakdown of an algorithm​’​s error. Continuing with our speech recognition example above,

the total dev set error of 30% can be broken down as follows (a similar analysis can be

applied to the test set error):

•**Optimal error rate (“unavoidable bias”)**​: 14%. Suppose we decide that, even with the

best possible speech system in the world, we would still suffer 14% error. We can think of

this as the “unavoidable” part of a learning algorithm​’​s bias.

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

•**Avoidable bias**​: 1%. This is calculated as the difference between the training error and

the optimal error rate.

3

•**Variance**​: 15%. The difference between the dev error and the training error.

To relate this to our earlier definitions, Bias and Avoidable Bias are related as follows:

4

Bias = Optimal error rate (“unavoidable bias”) + Avoidable bias

The “avoidable bias” reflects how much worse your algorithm performs on the training set

than the “optimal classifier.”

The concept of variance remains the same as before. In theory, we can always reduce

variance to nearly zero by training on a massive training set. Thus, all variance is “avoidable”

with a sufficiently large dataset, so there is no such thing as “unavoidable variance.”

Consider one more example, where the optimal error rate is 14%, and we have:

• Training error = 15%

• Dev error = 16%

Whereas in the previous chapter we called this a high bias classifier, now we would say that

error from avoidable bias is 1%, and the error from variance is about 1%. Thus, the algorithm

is already doing well, with little room for improvement. It is only 2% worse than the optimal

error rate.

We see from these examples that knowing the optimal error rate is helpful for guiding our

next steps. In statistics, the optimal error rate is also called ​**Bayes error rate**​, or Bayes

rate.

How do we know what the optimal error rate is? For tasks that humans are reasonably good

at, such as recognizing pictures or transcribing audio clips, you can ask a human to provide

labels then measure the accuracy of the human labels relative to your training set. This

would give an estimate of the optimal error rate. If you are working on a problem that even

3 If this number is negative, you are doing better on the training set than the optimal error rate. This

means you are overfitting on the training set, and the algorithm has over-memorized the training set.

You should focus on variance reduction methods rather than on further bias reduction methods.

4 These definitions are chosen to convey insight on how to improve your learning algorithm. These

definitions are different than how statisticians define Bias and Variance. Technically, what I define

here as “Bias” should be called “Error we attribute to bias”; and “Avoidable bias” should be “error we

attribute to the learning algorithm’s bias that is over the optimal error rate.”

*Page 9* *Machine Learning Yearning-Draft* *Andrew Ng*

humans have a hard time solving (e.g., predicting what movie to recommend, or what ad to

show to a user) it can be hard to estimate the optimal error rate.

In the section “Comparing to Human-Level Performance (Chapters 33 to 35), I will discuss

in more detail the process of comparing a learning algorithm’s performance to human-level

performance.

In the last few chapters, you learned how to estimate avoidable/unavoidable bias and

variance by looking at training and dev set error rates. The next chapter will discuss how you

can use insights from such an analysis to prioritize techniques that reduce bias vs.

techniques that reduce variance. There are very different techniques that you should apply

depending on whether your project’s current problem is high (avoidable) bias or high

variance. Read on!

*Page 10* *Machine Learning Yearning-Draft* *Andrew Ng*