

Machine Learning Yearning is a



deeplearning.ai project.

© 2018 Andrew Ng. All Rights Reserved.

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

23 Addressing Bias and Variance

Here is the simplest formula for addressing bias and variance issues:

• If you have high avoidable bias, increase the size of your model (for example, increase the

size of your neural network by adding layers/neurons).

• If you have high variance, add data to your training set.

If you are able to increase the neural network size and increase training data without limit, it

is possible to do very well on many learning problems.

In practice, increasing the size of your model will eventually cause you to run into

computational problems because training very large models is slow. You might also exhaust

your ability to acquire more training data. (Even on the internet, there is only a finite

number of cat pictures!)

Different model architectures—for example, different neural network architectures—will

have different amounts of bias/variance for your problem. A lot of recent deep learning

research has developed many innovative model architectures. So if you are using neural

networks, the academic literature can be a great source of inspiration. There are also many

great open-source implementations on github. But the results of trying new architectures are

less predictable than the simple formula of increasing the model size and adding data.

Increasing the model size generally reduces bias, but it might also increase variance and the

risk of overfitting. However, this overfitting problem usually arises only when you are not

using regularization. If you include a well-designed regularization method, then you can

usually safely increase the size of the model without increasing overfitting.

Suppose you are applying deep learning, with L2 regularization or dropout, with the

regularization parameter that performs best on the dev set. If you increase the model size,

usually your performance will stay the same or improve; it is unlikely to worsen significantly.

The only reason to avoid using a bigger model is the increased computational cost.

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

24 Bias vs. Variance tradeoff

You might have heard of the “Bias vs. Variance tradeoff.” Of the changes you could make to

most learning algorithms, there are some that reduce bias errors but at the cost of increasing

variance, and vice versa. This creates a “trade off” between bias and variance.

For example, increasing the size of your model—adding neurons/layers in a neural network,

or adding input features—generally reduces bias but could increase variance. Alternatively,

adding regularization generally increases bias but reduces variance.

In the modern era, we often have access to plentiful data and can use very large neural

networks (deep learning). Therefore, there is less of a tradeoff, and there are now more

options for reducing bias without hurting variance, and vice versa.

For example, you can usually increase a neural network size and tune the regularization

method to reduce bias without noticeably increasing variance. By adding training data, you

can also usually reduce variance without affecting bias.

If you select a model architecture that is well suited for your task, you might also reduce bias

and variance simultaneously. Selecting such an architecture can be difficult.

In the next few chapters, we discuss additional specific techniques for addressing bias and

variance.

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

25 Techniques for reducing avoidable bias

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

techniques:

•**Increase the model size** ​(such as number of neurons/layers): This technique reduces

bias, since it should allow you to fit the training set better. If you find that this increases

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

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

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

particular category of errors. (We discuss this further in the next chapter.) These new

features could help with both bias and variance. In 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.

•**Reduce or eliminate regularization**​ (L2 regularization, L1 regularization, dropout):

This will reduce avoidable bias, but increase 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.

One method that is not helpful:

•**Add more training data**​: This technique helps with variance problems, but it usually

has no significant effect on bias.

*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*