What is machine learning?

As the name suggests, machine learning can be loosely interpreted to mean empowering computer systems with the ability to “learn”.

The intention of ML is to enable machines to learn by themselves using the provided data and make accurate predictions.

ML is a subset of artificial intelligence; in fact, it’s simply a technique for realizing AI.

It is a method of training algorithms such that they can learn how to make decisions.

Training in machine learning entails giving a lot of data to the algorithm and allowing it to learn more about the processed information.

For example, here is a table that identifies the type of fruit based on its characteristics:

Weight (grams)	Texture	Type of Fruit
155	Rough	Orange
180	Rough	Orange
135	Smooth	Apple
110	Smooth	Apple
120	Smooth	?

As you can see on the table above, the fruits are differentiated based on their weight and texture.

However, the last row gives only the weight and texture, without the type of fruit.

And, a machine learning algorithm can be developed to try to identify whether the fruit is an orange or an apple.

After the algorithm is fed with the training data, it will learn the differing characteristics between an orange and an apple.

Therefore, if provided with data of weight and texture, it can predict accurately the type of fruit with those characteristics.

What is deep learning?

As earlier mentioned, deep learning is a subset of ML; in fact, it’s simply a technique for realizing machine learning. In other words, DL is the next evolution of machine learning.

DL algorithms are roughly inspired by the information processing patterns found in the human brain.

Just like we use our brains to identify patterns and classify various types of information, deep learning algorithms can be taught to accomplish the same tasks for machines.

The brain usually tries to decipher the information it receives. It achieves this through labelling and assigning the items into various categories.

Whenever we receive a new information, the brain tries to compare it to a known item before making sense of it—which is the same concept deep learning algorithms employ.

For example, artificial neural networks (ANNs) are a type of algorithms that aim to imitate the way our brains make decisions.

Comparing deep learning vs machine learning can assist you to understand their subtle differences.

For example, while DL can automatically discover the features to be used for classification, ML requires these features to be provided manually.

Understanding the Bias-Variance Tradeoff

Whenever we discuss model prediction, it's important to understand prediction errors (bias and variance). There is a tradeoff between a model's ability to minimize bias and variance. Gaining a proper understanding of these errors would help us not only to build accurate models but also to avoid the mistake of overfitting and underfitting.

So let's start with the basics and see how they make difference to our machine learning Models.

What is bias?

Bias is the difference between the average prediction of our model and the correct value which we are trying to predict. Model with high bias pays very little attention to the training data and oversimplifies the model. It always leads to high error on training and test data.

What is variance?

Variance is the variability of model prediction for a given data point or a value which tells us spread of our data. Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn't seen before. As a result, such models perform very well on training data but have high error rates on test data.

Mathematically

Let the variable we are trying to predict as Y and other covariates as X. We assume there is a relationship between the two such that

Y=f(X) + e

Where e is the error term and it's normally distributed with a mean of 0.

We will make a model f^(X) of f(X) using linear regression or any other modeling technique.

So the expected squared error at a point x is

$$Err(x) = E\left[(Y - \hat{f}(x))^2\right]$$

The Err(x) can be further decomposed as

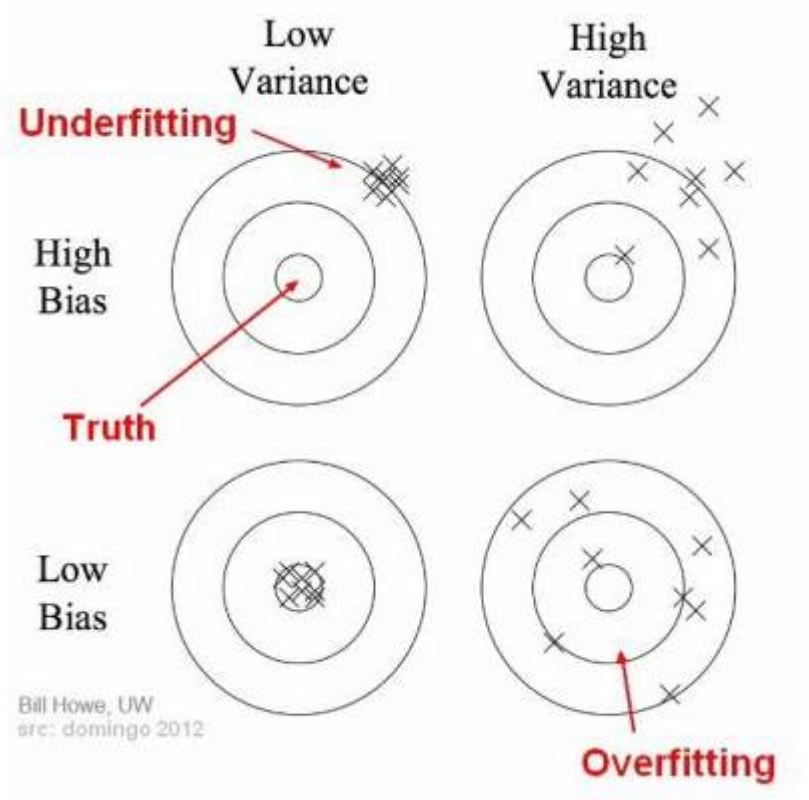
$$Err(x) = \left(E[\hat{f}\,(x)] - f(x)\right)^2 + E\left[\left(\hat{f}\,(x) - E[\hat{f}\,(x)]\right)^2\right] + \sigma_e^2$$

$$Err(x) = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

Err(x) is the sum of Bias², variance and the irreducible error.

Irreducible error is the error that can't be reduced by creating good models. It is a measure of the amount of noise in our data. Here it is important to understand that no matter how good we make our model, our data will have certain amount of noise or irreducible error that can not be removed.

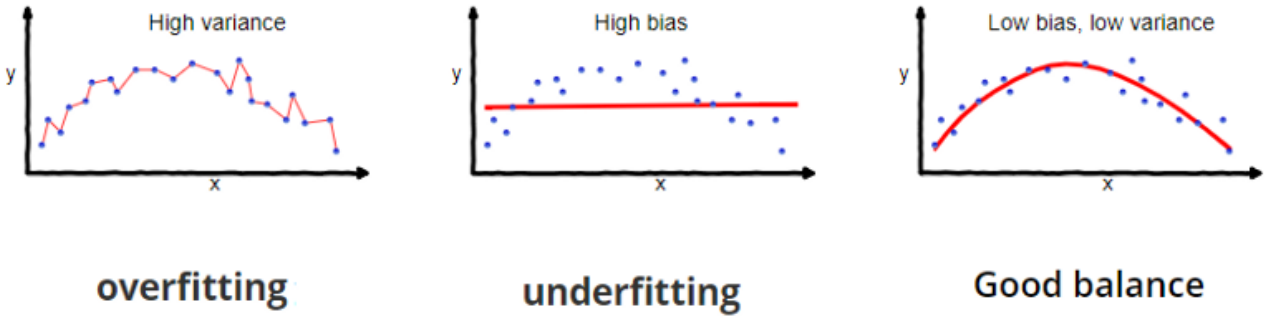
Bias and variance using bulls-eye diagram



In the above diagram, center of the target is a model that perfectly predicts correct values. As we move away from the bulls-eye our predictions become get worse and worse. We can repeat our process of model building to get separate hits on the target.

In supervised learning, **underfitting** happens when a model unable to capture the underlying pattern of the data. These models usually have high bias and low variance. It happens when we have very less amount of data to build an accurate model or when we try to build a linear model with a nonlinear data. Also, these kind of models are very simple to capture the complex patterns in data like Linear and logistic regression.

In supervised learning, **overfitting** happens when our model captures the noise along with the underlying pattern in data. It happens when we train our model a lot over noisy dataset. These models have low bias and high variance. These models are very complex like Decision trees which are prone to overfitting.



overfitting underfitting Good balance

Why is Bias Variance Tradeoff?

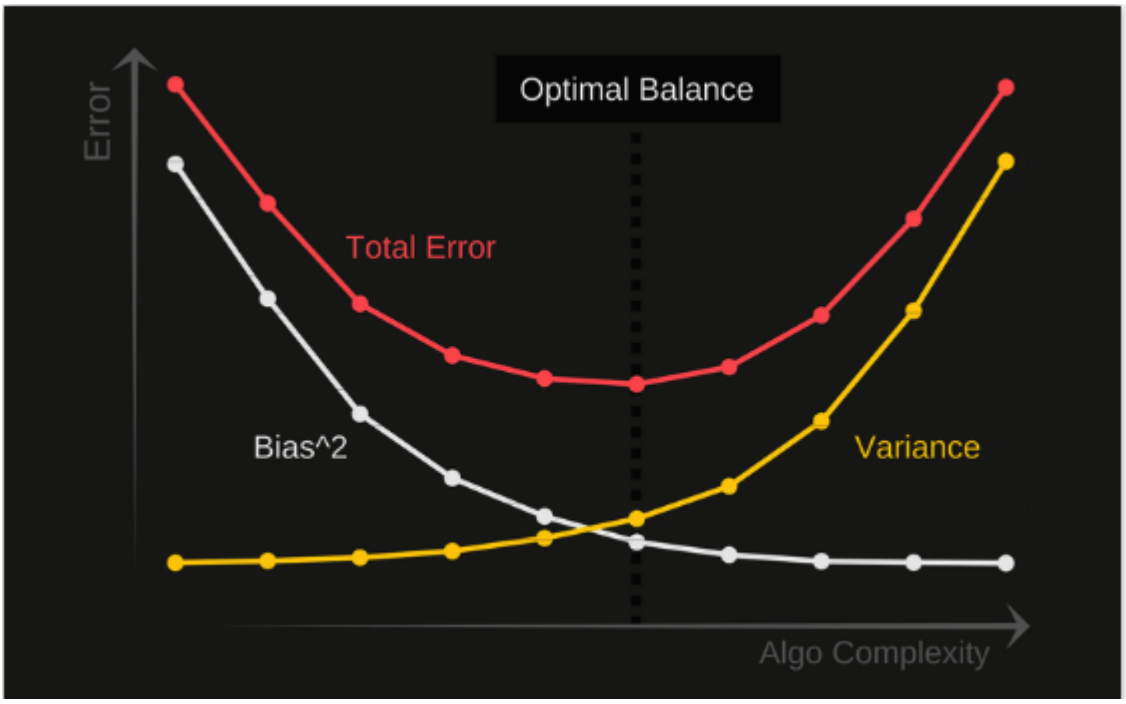
If our model is too simple and has very few parameters then it may have high bias and low variance. On the other hand if our model has large number of parameters then it's going to have high variance and low bias. So we need to find the right/good balance without overfitting and underfitting the data.

This tradeoff in complexity is why there is a tradeoff between bias and variance. An algorithm can't be more complex and less complex at the same time.

Total Error

To build a good model, we need to find a good balance between bias and variance such that it minimizes the total error.

Total Error = Bias^2 + Variance + Irreducible Error



An optimal balance of bias and variance would never overfit or underfit the model.

What is overfitting?

Overfitting is when model expressiveness is way too high. It is when your model fits training data perfectly but when you test your model against test data then it performs bad. When you are training your model on training data and it builds its rules and patterns around the training data such that it is unable to generalise on unseen data because of noise (randomness) in data then the model is unable to forecast scenarios that it has not experienced before. This model ends up accommodating stochastic behaviour in training input data and cannot generalise well. This is known as overfitting.
Overfitting is when model is bad at generalisation. Overfitting is a common issue of machine learning algorithms. This happens because training data contains noise but the model has managed to take noise into its algorithm.
To prepare forecasting model, you need to gather training and test data. If your training data contains noise then the model you will produce will provide inaccurate generalisation as it will carry the noise.
On the other hand, underfitting is opposite of overfitting. If a model is underfitting then it doesn't understand data well enough and cannot forecast values.

Avoiding Overfitting

There are several methods to avoid overfitting:
1. Increase size of your training and test data.
2. Reduce number of variables, degrees of freedom and parameters of your model. This will ensure your model is simple and will end up reducing noise (stochastic behaviour) in the training data.
3. Use cross validation technique. It compares average of the generalization error of the model with the previous average. Cross validation technique includes k-folds.

4. Penalize model parameters if they’re likely to cause overfitting. This process is known as regularization.

What does regularization mean?

One of the ways to reduce overfitting is by regularization. Extra terms can be introduced in the model to penalise overfitting. LASSO (L1) and Ridge (L2) are well-known regularization techniques. L1 and L2 are two loss function that penalize by the size / square of the size of coefficients.

- L1 minimises sum of the **absolute** differences between estimated and actual values.
- L2 minimises sum of the **squared** differences between estimated and actual values.

L1 is robust but L2 is considered stable.

7. What is gradient descend?

Gradient descend is an optimization algorithm. It aims to find points of a function that minimise its errors. Gradient descend is used in nearly all of the machine learning algorithms. When a machine learning algorithm forecasts data, we can find its cost function to estimate how good the algorithm is. Cost function monitors prediction errors in a machine learning algorithm. Predictive power of a machine learning algorithm can be improved by altering its parameters. We can iteratively enhance the parameters until the cost function is at its lowest point implying that the accuracy of the model is at its maximum. This process is known as gradient descend.

There are several variations of the algorithm including stochastic gradient descend. Stochastic Gradient Descent (SGD) is used to train neural networks.

Generalization in Machine Learning

In machine learning we describe the learning of the target function from training data as inductive learning.

Induction refers to learning general concepts from specific examples which is exactly the problem that supervised machine learning problems aim to solve. This is different from deduction that is the other way around and seeks to learn specific concepts from general rules.

Generalization refers to how well the concepts learned by a machine learning model apply to specific examples not seen by the model when it was learning.

The goal of a good machine learning model is to generalize well from the training data to any data from the problem domain. This allows us to make predictions in the future on data the model has never seen.

There is a terminology used in machine learning when we talk about how well a machine learning model learns and generalizes to new data, namely overfitting and underfitting.

Overfitting and underfitting are the two biggest causes for poor performance of machine learning algorithms.

Statistical Fit

In statistics, a fit refers to how well you approximate a target function.

This is good terminology to use in machine learning, because supervised machine learning algorithms seek to approximate the unknown underlying mapping function for the output variables given the input variables.

Statistics often describe the goodness of fit which refers to measures used to estimate how well the approximation of the function matches the target function.

Some of these methods are useful in machine learning (e.g. calculating the residual errors), but some of these techniques assume we know the form of the target function we are approximating, which is not the case in machine learning.

If we knew the form of the target function, we would use it directly to make predictions, rather than trying to learn an approximation from samples of noisy training data.

Overfitting in Machine Learning

Overfitting refers to a model that models the training data too well.

Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the

model. The problem is that these concepts do not apply to new data and negatively impact the models ability to generalize.

Overfitting is more likely with nonparametric and nonlinear models that have more flexibility when learning a target function. As such, many nonparametric machine learning algorithms also include parameters or techniques to limit and constrain how much detail the model learns.

For example, decision trees are a nonparametric machine learning algorithm that is very flexible and is subject to overfitting training data. This problem can be addressed by pruning a tree after it has learned in order to remove some of the detail it has picked up.

Underfitting in Machine Learning

Underfitting refers to a model that can neither model the training data nor generalize to new data.

An underfit machine learning model is not a suitable model and will be obvious as it will have poor performance on the training data.

Underfitting is often not discussed as it is easy to detect given a good performance metric. The remedy is to move on and try alternate machine learning algorithms. Nevertheless, it does provide a good contrast to the problem of overfitting.

A Good Fit in Machine Learning

Ideally, you want to select a model at the sweet spot between underfitting and overfitting.

This is the goal, but is very difficult to do in practice.

To understand this goal, we can look at the performance of a machine learning algorithm over time as it is learning a training data. We can plot both the skill on the training data and the skill on a test dataset we have held back from the training process.

Over time, as the algorithm learns, the error for the model on the training data goes down and so does the error on the test dataset. If we train for too long, the performance on the training dataset may continue to decrease because the model is overfitting and learning the irrelevant detail and noise in the training dataset. At the same time the error for the test set starts to rise again as the model's ability to generalize decreases.

The sweet spot is the point just before the error on the test dataset starts to increase where the model has good skill on both the training dataset and the unseen test dataset.

You can perform this experiment with your favorite machine learning algorithms. This is often not useful technique in practice, because by choosing the stopping point for training using the skill on the test dataset it means that the testset is no longer "unseen" or a standalone objective measure. Some knowledge (a lot of useful knowledge) about that data has leaked into the training procedure.

There are two additional techniques you can use to help find the sweet spot in practice: resampling methods and a validation dataset.

How To Limit Overfitting

Both overfitting and underfitting can lead to poor model performance. But by far the most common problem in applied machine learning is overfitting.

Overfitting is such a problem because the evaluation of machine learning algorithms on training data is different from the evaluation we actually care the most about, namely how well the algorithm performs on unseen data.

There are two important techniques that you can use when evaluating machine learning algorithms to limit overfitting:

1. Use a resampling technique to estimate model accuracy.
2. Hold back a validation dataset.

The most popular resampling technique is k-fold cross validation. It allows you to train and test your model k-times on different subsets of training data and build up an estimate of the performance of a machine learning model on unseen data.

A validation dataset is simply a subset of your training data that you hold back from your machine learning algorithms until the very end of your project. After you have selected and tuned your machine learning algorithms on your training dataset you can evaluate the learned models on the validation dataset to get a final objective idea of how the models might perform on unseen data.

Using cross validation is a gold standard in applied machine learning for estimating model accuracy on unseen data. If you have the data, using a validation dataset is also an excellent practice.

Further Reading

This section lists some recommended resources if you are looking to learn more about generalization, overfitting and underfitting in machine learning.

Summary

In this post, you discovered that machine learning is solving problems by the method of induction.

You learned that generalization is a description of how well the concepts learned by a model apply to new data. Finally, you learned about the terminology of generalization in machine learning of overfitting and underfitting:

- **Overfitting**: Good performance on the training data, poor generliazation to other data.

- **Underfitting**: Poor performance on the training data and poor generalization to other data

Do you have any questions about overfitting, underfitting or this post? Leave a comment and ask your question and I will do my best to answer it.