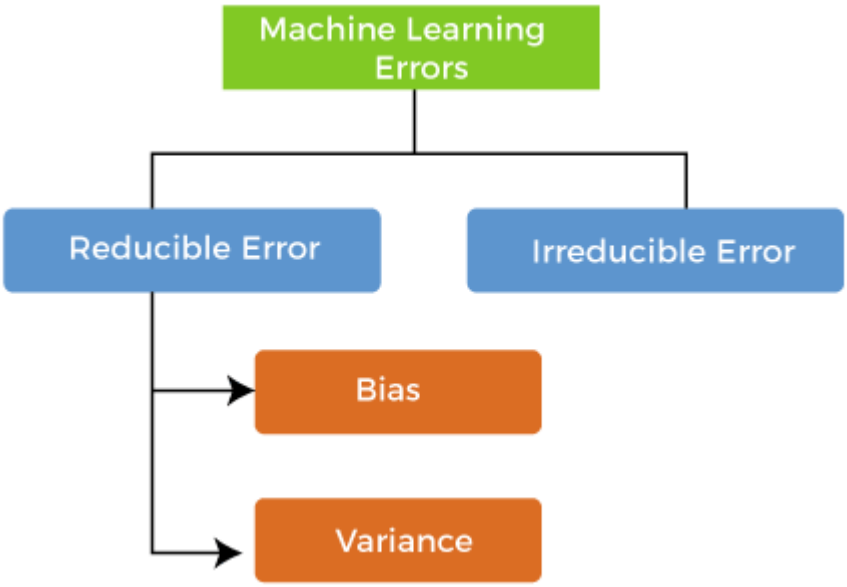
Bias-Variance

Errors in Machine Learning?

In machine learning, an error is a measure of how accurately an algorithm can make predictions for the previously unknown dataset. On the basis of these errors, the machine learning model is selected that can perform best on the particular dataset. There are mainly two types of errors in machine learning, which are:

- **Reducible errors:** These errors can be reduced to improve the model accuracy. Such errors can further be classified into bias and Variance.



- **Irreducible errors:** These errors will always be present in the model regardless of which algorithm has been used. The cause of these errors is unknown variables whose value can't be reduced.

What is Bias?

While making predictions, a difference occurs between prediction values made by the model and actual values/expected values, and this difference is known as bias errors or Errors due to bias.

Bias represents the distance of our predicted values with the original values.

- **Low Bias:** A low bias model will make fewer assumptions about the form of the target function.
- **High Bias:** A model with a high bias makes more assumptions, and the model becomes unable to capture the important features of our dataset. **A high bias model also cannot perform well on new data.**
- Generally, a linear algorithm has a high bias, as it makes them learn fast. The simpler the algorithm, the higher the bias it has likely to be introduced. Whereas a nonlinear algorithm often has low bias.
- Some examples of machine learning algorithms with low bias **are Decision Trees, k-Nearest Neighbours and Support Vector Machines**. At the same time, an algorithm with high bias is **Linear Regression, Linear Discriminant Analysis and Logistic Regression.**

Ways to reduce High Bias:

High bias mainly occurs due to a much simple model. Below are some ways to reduce the high bias:

- Increase the input features as the model is underfitted.
- Decrease the regularization term.
- Use more complex models, such as including some polynomial features.

What is a Variance Error?

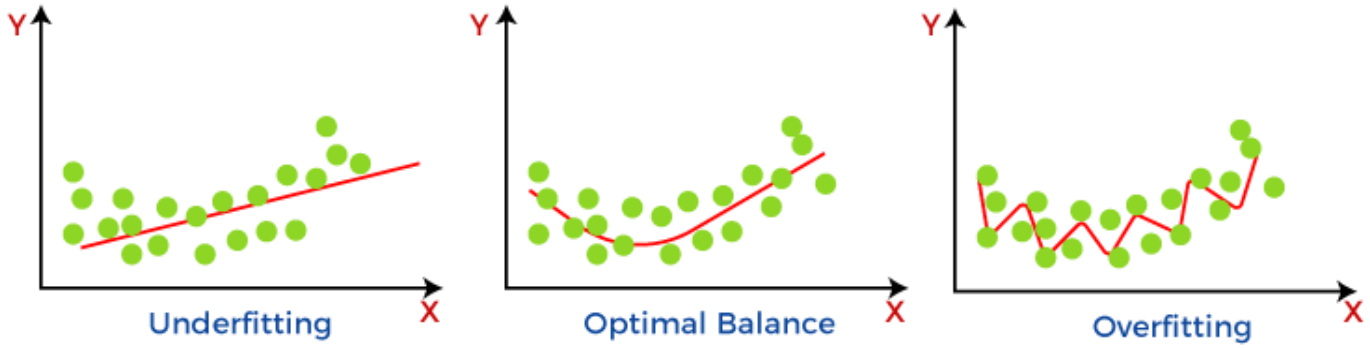
variance tells that how much a random variable is different from its expected value.

Variance represents the distance of the predicted values, which means how much the predicted values are scattered from each other.

A model with high variance has the below problems:

- A high variance model leads to overfitting.
- Increase model complexities.

Usually, nonlinear algorithms have a lot of flexibility to fit the model, have high variance.



Some examples of machine learning algorithms with low variance are, **Linear Regression, Logistic Regression, and Linear discriminant analysis**. At the same time, algorithms with high variance are **decision tree, Support Vector Machine, and K-nearest neighbours.**

Ways to Reduce High Variance:

- Reduce the input features or number of parameters as a model is overfitted.
- Do not use a much complex model.
- Increase the training data.
- Increase the Regularization term.

Different Combinations of Bias-Variance

There are four possible combinations of bias and variances, which are represented by the below diagram:

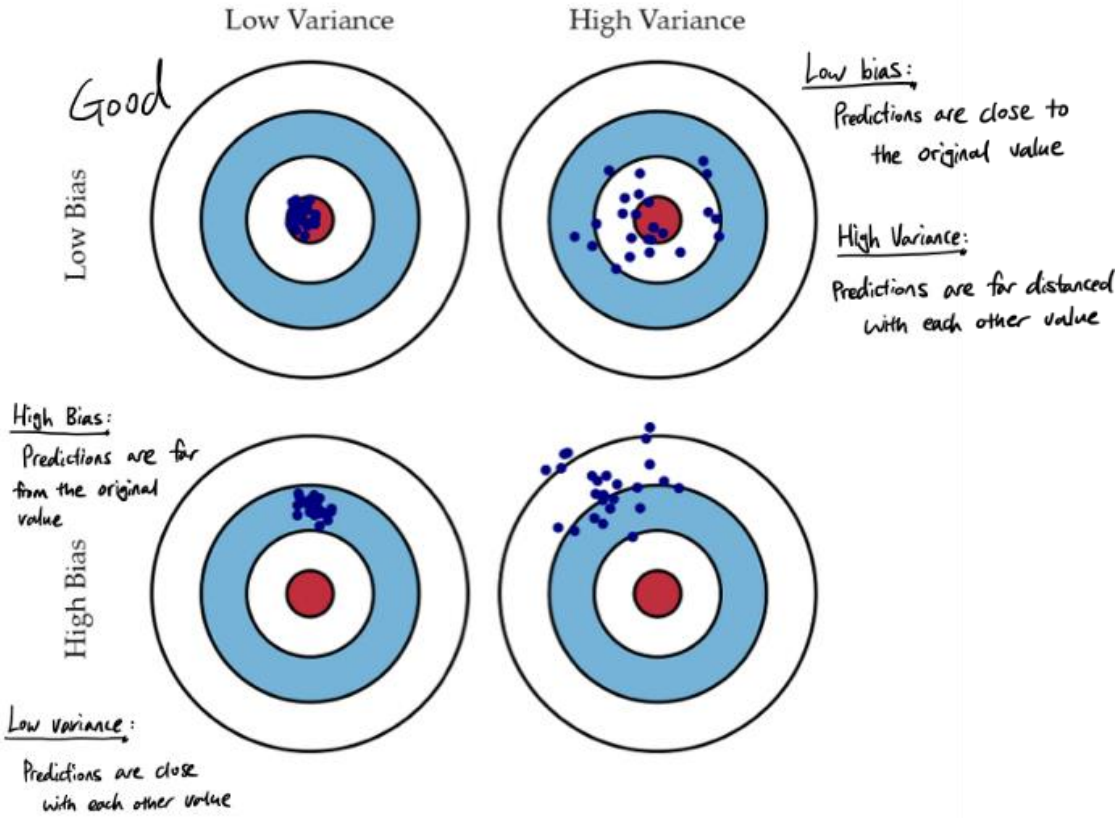
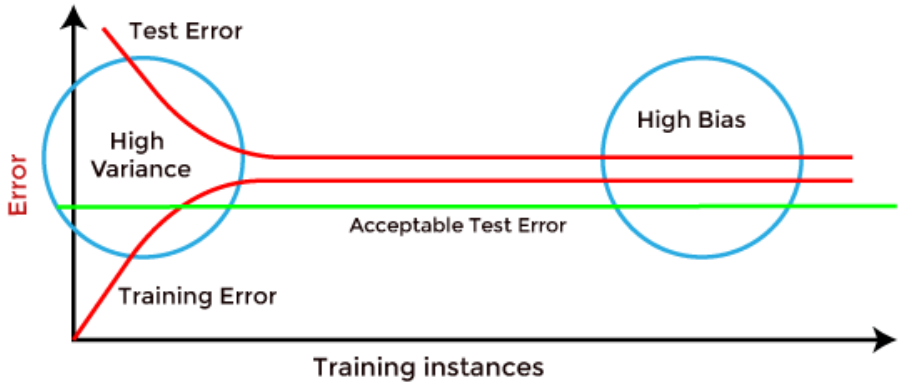


Fig. 1 Graphical illustration of bias and variance.

1. **Low-Bias, Low-Variance:** The combination of low bias and low variance shows an ideal machine learning model. However, it is not possible practically.
2. **Low-Bias, High-Variance:** With low bias and high variance, model predictions are inconsistent and accurate on average. This case occurs when the model learns with a large number of parameters and hence leads to an **overfitting**
3. **High-Bias, Low-Variance:** With High bias and low variance, predictions are consistent but inaccurate on average. This case occurs when a model does not learn well with the training dataset or uses few numbers of the parameter. It leads to **underfitting** problems in the model.
4. **High-Bias,High-Variance:** With high bias and high variance, predictions are inconsistent and also inaccurate on average.

How to identify High variance or High Bias?

High variance can be identified if the model has:



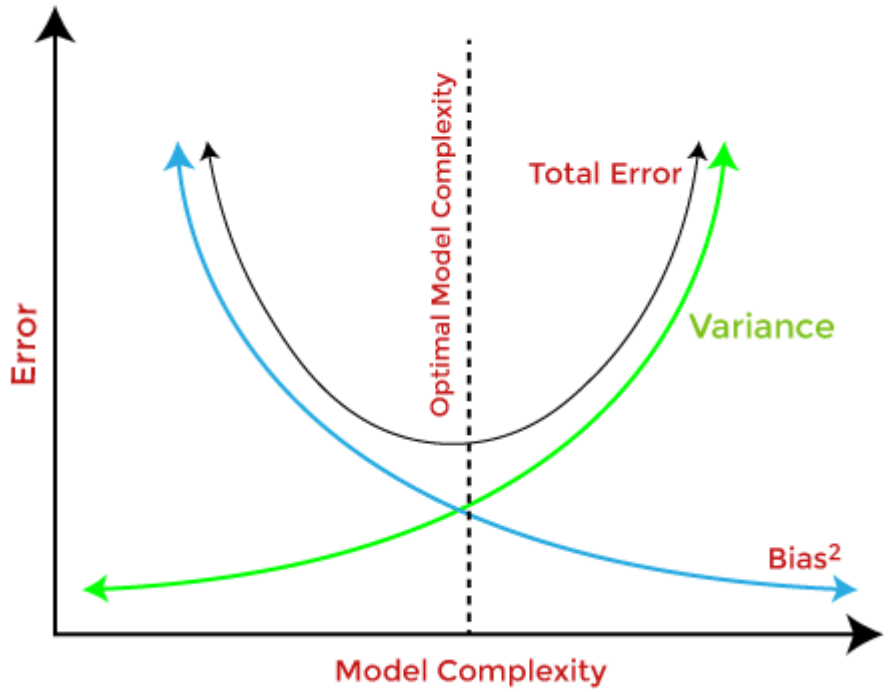
- Low training error and high test error.

High Bias can be identified if the model has:

- High training error and the test error is almost similar to training error.

Bias-Variance Trade-Off

While building the machine learning model, it is really important to take care of bias and variance in order to avoid overfitting and underfitting in the model. If the model is very simple with fewer parameters, it may have low variance and high bias. Whereas, if the model has a large number of parameters, it will have high variance and low bias. So, it is required to make a balance between bias and variance errors, and this balance between the bias error and variance error is known as **the Bias-Variance trade-off.**



For an accurate prediction of the model, algorithms need a low variance and low bias. But this is not possible because bias and variance are related to each other:

- If we decrease the variance, it will increase the bias.
- If we decrease the bias, it will increase the variance.

Bias-Variance trade-off is a central issue in supervised learning. Ideally, we need a model that accurately captures the regularities in training data and simultaneously generalizes well with the unseen dataset. Unfortunately, doing this is not possible simultaneously. Because a high variance algorithm may perform well with training data, but it may lead to overfitting to noisy data. Whereas, high bias algorithm generates a much simple model that may not even capture important regularities in the data. So, we need to find a sweet spot between bias and variance to make an optimal model.

Hence, the ***Bias-Variance trade-off is about finding the sweet spot to make a balance between bias and variance errors.***

Cost Function

Machine Learning models require a high level of accuracy to work in the actual world. But how do you calculate how wrong or right your model is? A machine learning parameter that is used for correctly judging the model, cost functions are important to understand to know how well the model has estimated the relationship between your input and output parameters.

What is Cost Function ?

After training your model, you need to see how well your model is performing. While accuracy functions tell you how well the model is performing, they do not provide you with an insight on how to better improve them. Hence, you need a correctional function that can help you compute when the model is the most accurate, as you need to hit that small spot between an undertrained model and an overtrained model.

A Cost Function is used to measure just how wrong the model is in finding a relation between the input and output. It tells you how badly your model is behaving/predicting

What Is Gradient Descent?

Gradient Descent is an algorithm that is used to optimize the cost function or the error of the model. It is used to find the minimum value of error possible in your model.

Gradient Descent can be thought of as the direction you have to take to reach the least possible error. The error in your model can be different at different points, and you have to find the quickest way to minimize it, to prevent resource wastage.

Gradient Descent can be visualized as a ball rolling down a hill. Here, the ball will roll to the lowest point on the hill. It can take this point as the point where the error is least as for any model, the error will be minimum at one point and will increase again after that.

In gradient descent, you find the error in your model for different values of input variables. This is repeated, and soon you see that the error values keep getting smaller and smaller. Soon you'll arrive at the values for variables when the error is the least, and the cost function is optimized.

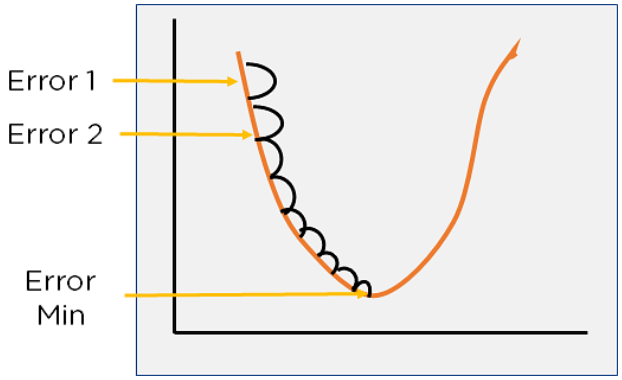


Figure 2: Gradient Descent

What Is the Cost Function For Linear Regression?

A Linear Regression model uses a straight line to fit the model. This is done using the equation for a straight line as shown :

$$\text{Output} = a*\text{Input} + b$$

Figure 3: Linear regression function

In the equation, you can see that two entities can have changeable values (variable) a, which is the point at which the line intercepts the x-axis, and b, which is how steep the line will be, or slope.

At first, if the variables are not properly optimized, you get a line that might not properly fit the model. As you optimize the values of the model, for some variables, you will get the perfect fit. The perfect fit will be a straight line running through most of the data points while ignoring the noise and outliers. A properly fit Linear Regression model looks as shown below :

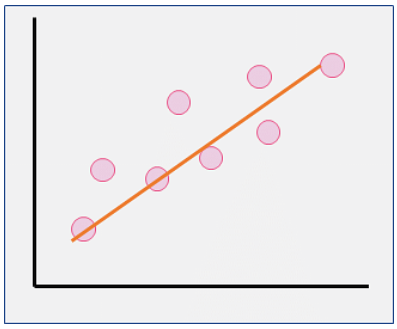


Figure 4: Linear regression graph

For the Linear regression model, the **cost function** will be the minimum of the **Root Mean Squared Error** of the model, obtained by subtracting the predicted values from actual values. The cost function will be the minimum of these error values.

$$\text{Cost Function (J)} = \frac{1}{n}\sum_{i=0}^{n}(h_\theta(x^i) - y^i)^2$$

Figure 5: Linear regression cost function

By the definition of gradient descent, you have to find the direction in which the error decreases constantly. This can be done by finding the difference between errors.

The small difference between errors can be obtained by differentiating the cost function and subtracting it from the previous gradient descent to move down the slope.

$$\text{Gradient Descent } \theta_j = \theta_j - \alpha \frac{\partial J}{\partial \theta}$$

After substituting the value of the cost function (J) in the above equation, you get :

Figure 6: Linear regression gradient descent function simplified

In the above equations, a is known as the learning rate. It decides how fast you move down the slope. If alpha is large, you take big steps, and if it is small; you take small steps. If alpha is too large, you can entirely miss the least error point and our results will not be accurate. If it is too small it will take too long to optimize the model and you will also waste computational power. Hence you need to choose an optimal value of alpha.

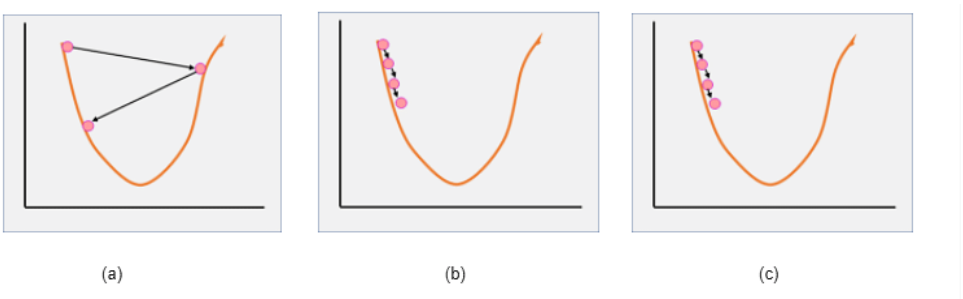


Figure 8: (a) Large learning rate, (b) Small learning rate, (c) Optimum learning rate

What is Cost Function for Neural Networks?

A neural network is a machine learning algorithm that takes in multiple inputs, runs them through an algorithm, and essentially sums the output of the different algorithms to get the final output.

The cost function of a neural network will be the sum of errors in each layer. This is done by finding the error at each layer first and then summing the individual error to get the total error. In the end, it can represent a neural network with cost function optimization as :

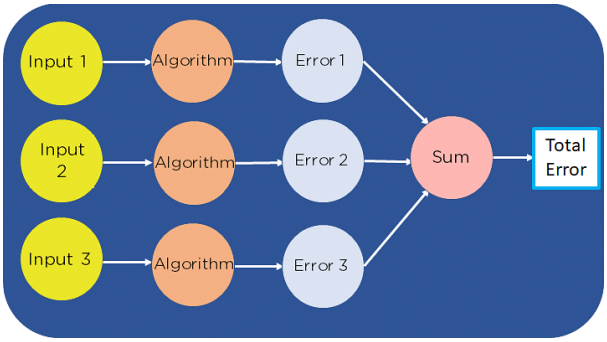


Figure 9: Neural network with the error function

For neural networks, each layer will have a cost function, and each cost function will have its own least minimum error value. Depending on where you start, you can arrive at a unique value for the minimum error. You need to find the minimum value out of all local minima. This value is called the global minima.

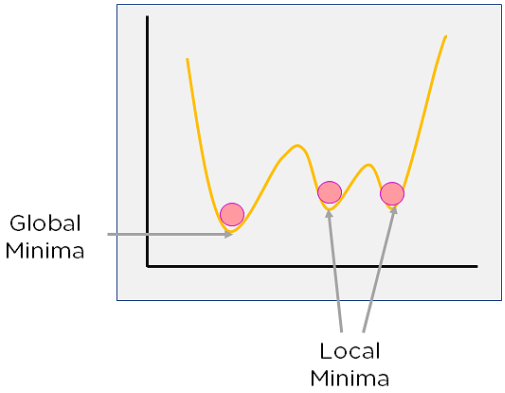


Figure 10: Cost function graph for Neural Networks

The cost function for neural networks is given as :

$$\text{Cost Function (J)} = \frac{1}{n}\sum_{i=0}^{n}(y^i - (mx^i + b))^2$$

Figure 11: Cost function for Neural Networks

Gradient descent is just the differentiation of the cost function. It is given as :

$$\text{Gradient Descent } (\frac{\partial J}{\partial \theta}) = \begin{bmatrix} \frac{1}{N}\sum_{i=0}^{n}(-2\,x_i(y_i - (mx_i + b))) \\ \frac{1}{N}\sum_{i=0}^{n}(-2\,(y_i - (mx_i + b))) \end{bmatrix}$$

Figure 12: Gradient descent for Neural Networks