The Best Guide to Regularization in Machine Learning

Lesson 24 of 33

By Mayank Ba

Last updated on Oct 26, 2022

Previous Next

Tutorial Playlist

Table of Contents

While training a machine learning model, the model can easily be overfitted or under fitted. To avoid this, we use regularization in machine learning to properly fit a model onto our test set. Regularization techniques help reduce the chance of overfitting and help us get an optimal model. In this article titled ‘The Best Guide to Regularization in Machine Learning’, you will learn all you need to know about regularization.

What Are Overfitting and Underfitting?

To train our machine learning model, we give it some data to learn from. The process of plott a series of data points and drawing the best fit line to understand the relationship between th variables is called Data Fitting. Our model is the best fit when it can find all necessary patterr our data and avoid the random data points and unnecessary patterns called Noise.

If we allow our machine learning model to look at the data too many times, it will find a lot of patterns in our data, including the ones which are unnecessary. It will learn really well on the test dataset and fit very well to it. It will learn important patterns, but it will also learn from the noise in our data and will not be able to predict on other datasets.

A scenario where the machine learning model tries to learn from the details along with the noise in the data and tries to fit each data point on the curve is called Overfitting.

In the figure depicted below, we can see that the model is fit for every point in our data. If given new data, the model curves may not correspond to the patterns in the new data, and the model cannot predict very well in it.

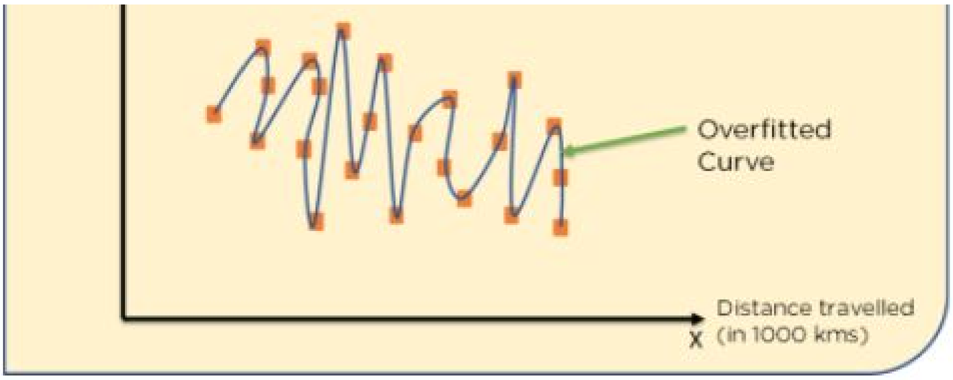


Figure 1: Overfitted Model

Conversely, in a scenario where the model has not been allowed to look at our data a sufficie number of times, the model won't be able to find patterns in our test dataset. It will not fit properly to our test dataset and fail to perform on new data too.

A scenario where a machine learning model can neither learn the relationship between variat in the testing data nor predict or classify a new data point is called Underfitting.

The below diagram shows an under-fitted model. We can see that it has not fit properly to the data given to it. It has not found patterns in the data and has ignored a large part of the dataset. It cannot perform on both known and unknown data.

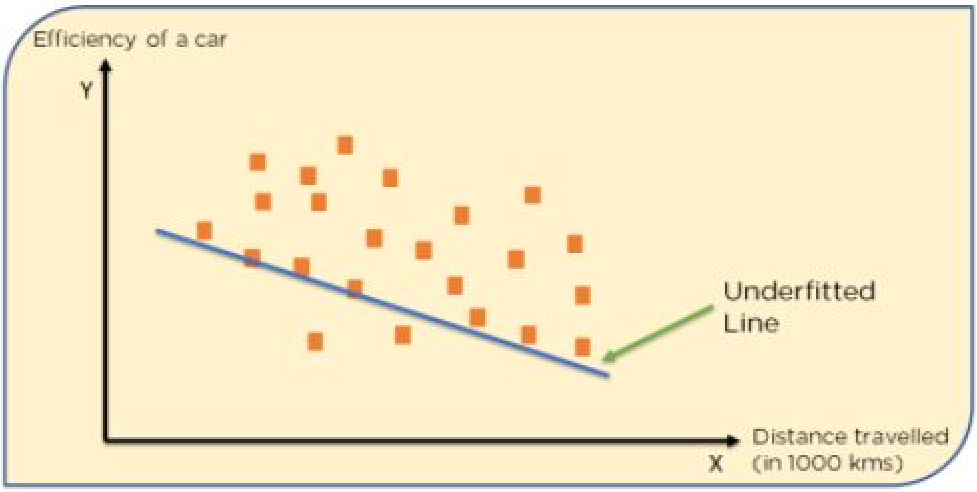


Figure 2: Underfitted Model

What are Bias and Variance?

A Bias occurs when an algorithm has limited flexibility to learn from data. Such models pay very little attention to the training data and oversimplify the model therefore the validation error or prediction error and training error follow similar trends. Such models always lead to a high error prediction error and training error follow similar trends. Such models always lead to a high error on training and test data. High Bias causes underfitting in our model.

Variance defines the algorithm's sensitivity to specific sets of data. A model with a high variance pays a lot of attention to training data and does not generalize therefore the validation error or prediction error are far apart from each other. Such models usually perform very well on training data but have high error rates on test data. High Variance causes overfitting in our model.

An optimal model is one in which the model is sensitive to the pattern in our model, but at the same time can generalize to new data. This happens when Bias and Variance are both optim We call this Bias-Variance Tradeoff and we can achieve it in over or under fitted models by us Regression.

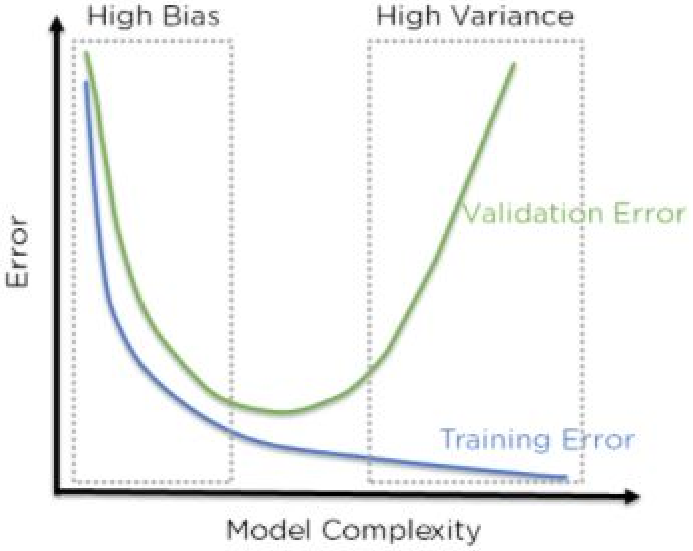
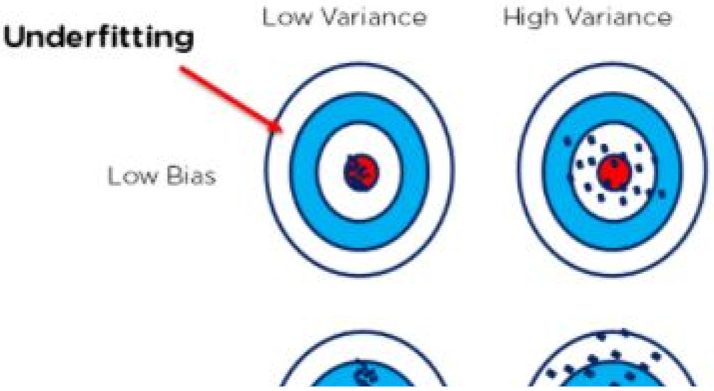


Figure 3: Error in testing and training datasets with high bias and variance

In the above figure, we can see that when bias is high, the error in both testing and training set is also high. When Variance is high, the model performs well on our training set and gives a low error, but the error in our testing set is very high. In the middle of this exists a region where the bias and variance are in perfect balance to each other, and here, but the training and testing errors are low.



High Bias
Overfitting

Figure 4: Bullseye diagram for different bias and variance levels

What is Regularization in Machine Learning?

Regularization refers to techniques that are used to calibrate machine learning models in order to minimize the adjusted loss function and prevent overfitting or underfitting.

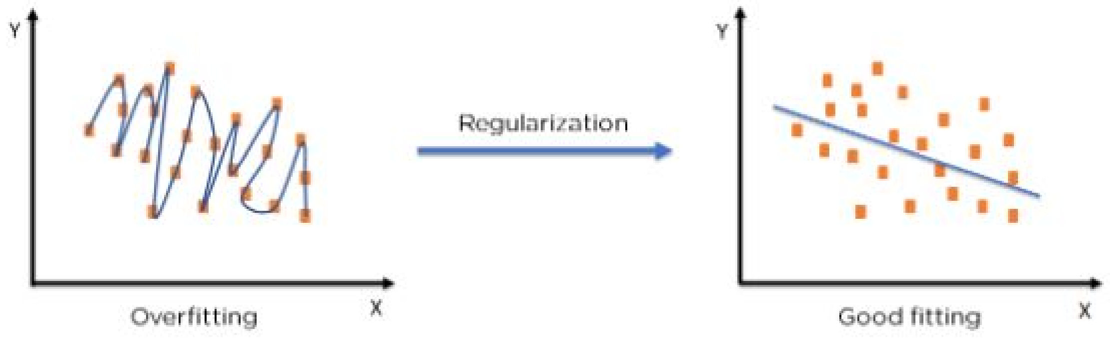


Figure 5: Regularization on an over-fitted model

Using Regularization, we can fit our machine learning model appropriately on a given test set and hence reduce the errors in it.

Regularization Techniques

Regularization Techniques

There are two main types of regularization techniques: Ridge Regularization and Lasso Regularization.

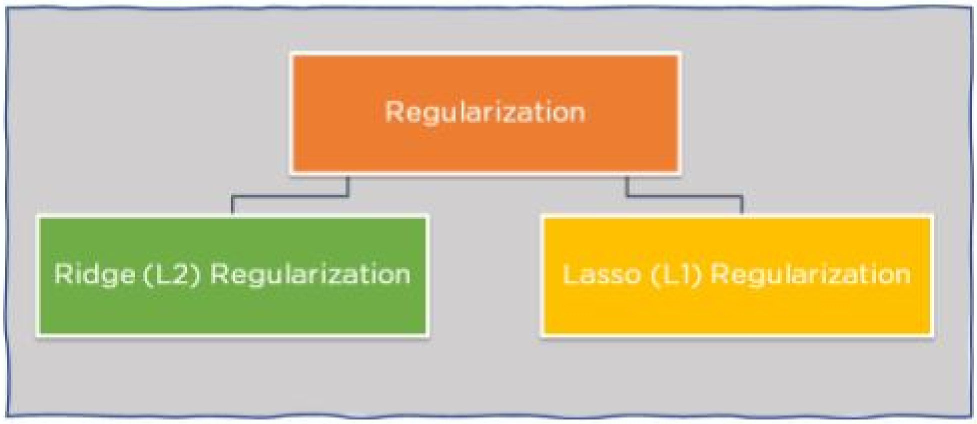
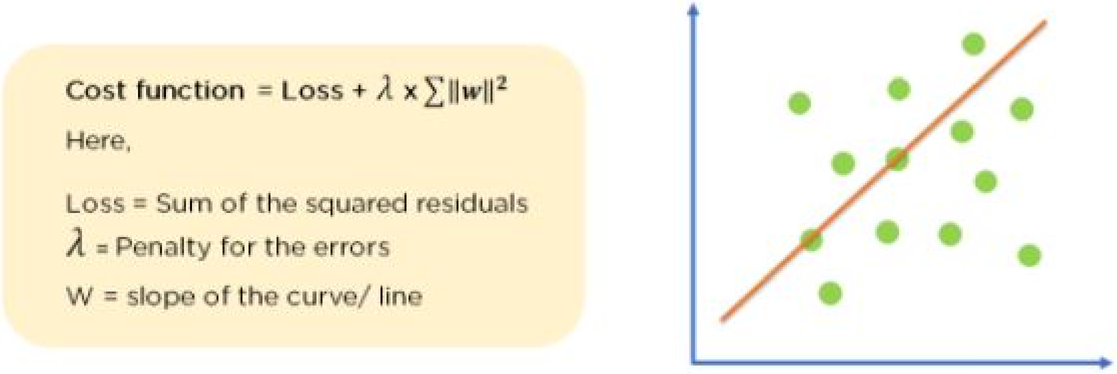


Figure 6: Regularization techniques

Ridge Regularization :

Also known as Ridge Regression, it modifies the over-fitted or under fitted models by adding the penalty equivalent to the sum of the squares of the magnitude of coefficients.

This means that the mathematical function representing our machine learning model is minimized and coefficients are calculated. The magnitude of coefficients is squared and added. Ridge Regression performs regularization by shrinking the coefficients present. The function depicted below shows the cost function of ridge regression :



Cost function = Loss + $\lambda \times \sum \|w\|^2$

Here,

Loss = Sum of the squared residuals

λ = Penalty for the errors

W = slope of the curve/ line

Figure 7: Cost Function of Ridge Regression

In the cost function, the penalty term is represented by Lambda λ. By changing the values of the penalty function, we are controlling the penalty term. The higher the penalty, it reduces the magnitude of coefficients. It shrinks the parameters. Therefore, it is used to prevent multicollinearity, and it reduces the model complexity by coefficient shrinkage.

Consider the graph illustrated below which represents Linear regression :

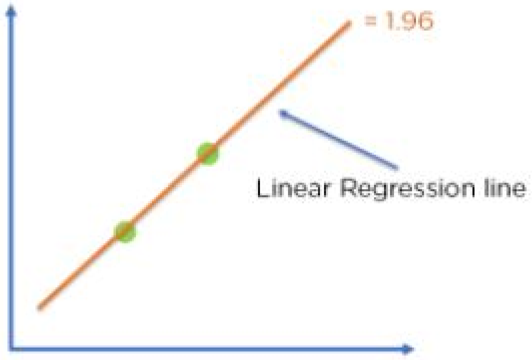


Figure 8: Linear regression model

Cost function = Loss + λ x∑‖w‖^2

For Linear Regression line, let’s consider two points that are on the line,

Loss = 0 (considering the two points on the line)

λ= 1

w = 1.4

Then, Cost function = 0 + 1 x 1.42

= 1.96

For Ridge Regression, let’s assume,

Loss = 0.32 + 0.22 = 0.13

λ = 1

w = 0.7

Then, Cost function = 0.13 + 1 x 0.72

= 0.62

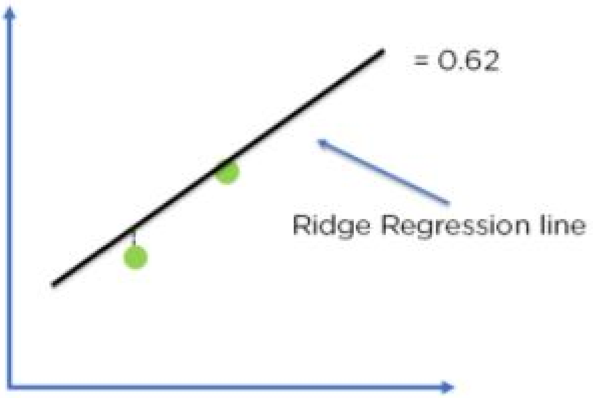


Figure 9: Ridge regression model

Comparing the two models, with all data points, we can see that the Ridge regression line fits model more accurately than the linear regression line.

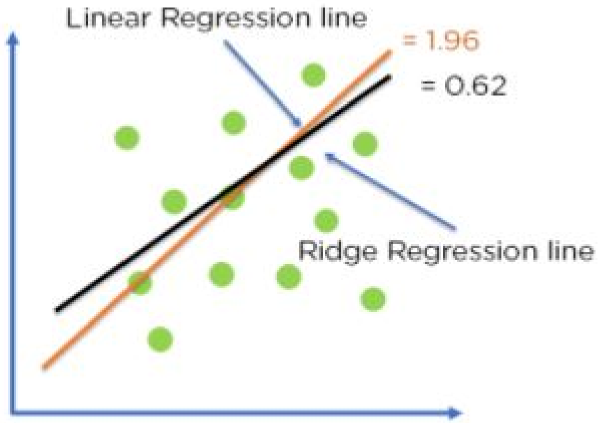
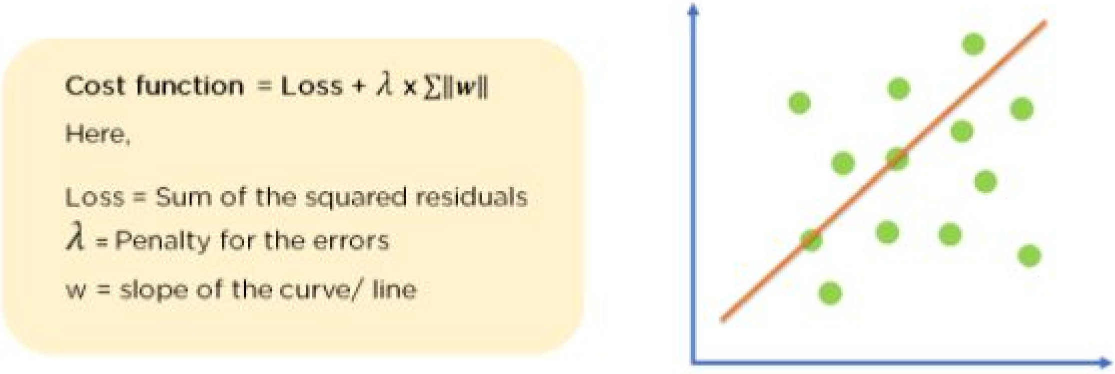


Figure 10: Optimization of model fit using Ridge Regression

Lasso Regression

It modifies the over-fitted or under-fitted models by adding the penalty equivalent to the sum of the absolute values of coefficients.

Lasso regression also performs coefficient minimization, but instead of squaring the magnitudes of the coefficients, it takes the true values of coefficients. This means that the coefficient sum can also be 0, because of the presence of negative coefficients. Consider the cost function for Lasso regression :



Cost function = Loss + λ x ∑‖*w*‖

Here,

Loss = Sum of the squared residuals

λ = Penalty for the errors

w = slope of the curve/ line

Figure 11: Cost function for Lasso Regression

We can control the coefficient values by controlling the penalty terms, just like we did in Ridge Regression. Again consider a Linear Regression model :

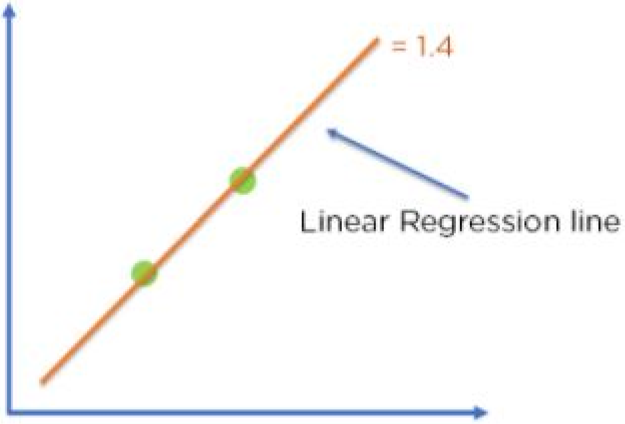


Figure 12: Linear Regression Model

Figure 12: Linear Regression Model

Cost function = Loss + λ x ∑‖w‖

For Linear Regression line, let’s assume, Loss = 0 (considering the two points on the line)

λ = 1

w = 1.4

Then, Cost function = 0 + 1 x 1.4

= 1.4

For Ridge Regression, let’s assume,

Loss = 0.32 + 0.12 = 0.1

λ = 1

w = 0.7

Then, Cost function = 0.1 + 1 x 0.7

= 0.8

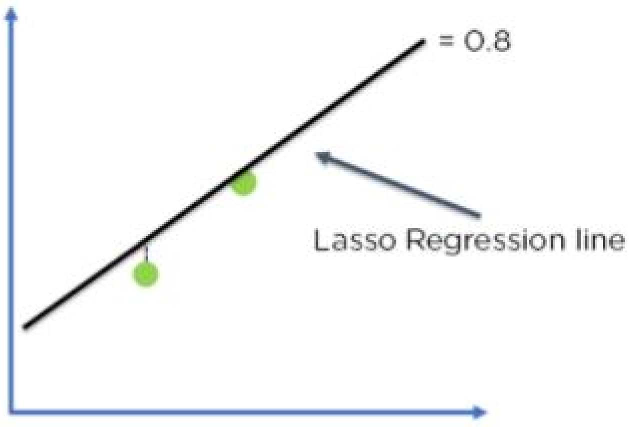


Figure 13: Lasso Regression

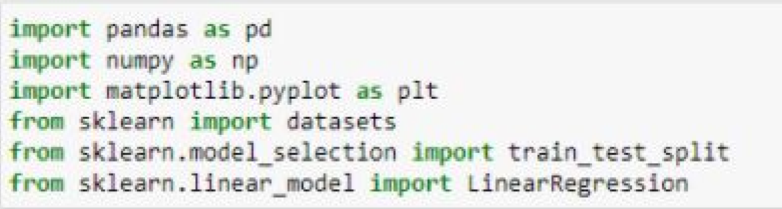
Comparing the two models, with all data points, we can see that the Lasso regression line fits the model more accurately than the linear regression line.

Artificial Intelligence Engineer

Regularization Using Python in Machine Learning

Let's look at how regularization can be implemented in Python. We have taken the Boston Housing Dataset on which we will be using Linear Regression to predict housing prices in Boston.

We start by importing all the necessary modules.



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

Figure 11: Importing modules in python

We then load the Boston Housing Dataset from sklearn’s datasets.

Figure 12: Loading Boston Housing Dataset

We then convert the dataset into a DataFrame and set the columns and the target variable.

Figure 13: Converting dataset into DataFrame

The below figure shows us the Boston housing dataset.

Figure 14: Boston Housing Dataset

Figure 14: Boston Housing Dataset

We then split our data into training and testing sets.

Figure 15: Splitting to training and testing sets

We can now use these to train our Linear Regression model. We start by creating our model and fitting the data to it. We then predict on the test set and find the error in our prediction using mean_squared_error. Finally, we print the coefficients of our Linear Regression model.

Figure 16: Linear Regression

The coefficients of our Linear Regression model are given below.

Figure 17: Coefficients of Linear Regression

Now, let’s plot the coefficient score.

Figure 17: Plotting coefficient score of Linear Regression model

Figure 18: Coefficient score of Linear Regression model

Now, let us perform Ridge regression and plot new coefficients that we get from it.

Figure 19: Ridge Regression and plotting coefficients

Figure 20: Coefficients of Ridge Regression model

Now let’s plot the coefficient score of the Ridge Regression model

Figure 21: Plotting coefficient score of Ridge Regression model

Figure 22: Coefficient score of Ridge Regression model

Let’s perform Lasso Regression and find the coefficients for it.

Figure 23: Lasso Regression and plotting coefficients

Figure 23: Coefficients of Lasso Regression model

Conclusion

In this article - The Best Guide to Regularization in Machine Learning, we were introduced to the various ways in which models can become unstable by being under-fitted or over-fitted. We also saw the role of bias and variance in model optimization. We then moved on to regularization and saw various regularization techniques to overcome over and under fitting. Finally, we saw how to implement regularization in Python with a demo.

Was this article on regularization useful to you? Do you have any doubts or questions for us? Mention them in this article's comments section, and we'll have our experts answer them for at the earliest!

Looking forward to becoming a Machine Learning Engineer? Check out Simplilearn's AI ML Course and get certified today.

Find our Professional Certificate Program in AI and Machine Learning Online Bootcamp in top cities:

Name	Date	Place
Professional Certificate Program in AI and Machine Learning	Cohort starts on 10th Nov 2022, Weekend batch	Your City
Professional Certificate Program in AI and Machine Learning	Cohort starts on 17th Nov 2022, Weekend batch	Hyderabad
Professional Certificate Program in AI and Machine Learning	Cohort starts on 21st Nov 2022, Weekend batch	Pune

About the Author

Mayank Banoula

Mayank is a Research Analyst at Simplilearn. He is proficient in Machine learning and Artificial intelligence with python.

View More

Recommended Resources

Supervised and Unsupervised Learning...

Machine Learning Career Guide: A compl...

Disclaimer

PMP, PMI, PMBOK, CAPM, PgMP, PfMP, ACP, PBA, RMP, SP, and OPM3 are registered marks of the Project Management Institute, Inc.

*According to Simplilearn survey conducted and subject to terms & conditions with Ernst & Young LLP (EY) as Process Advisors