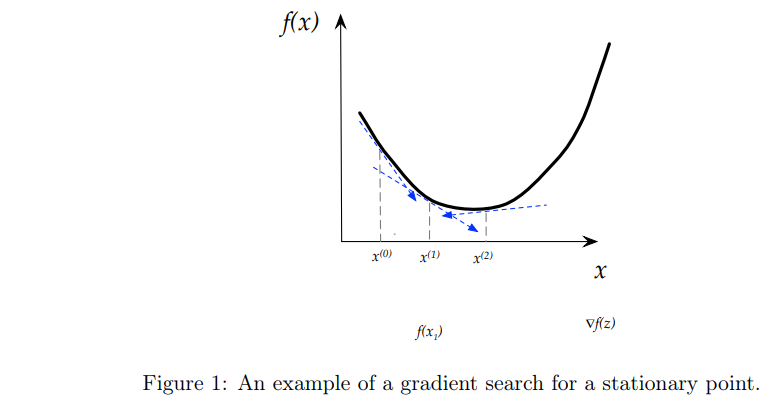
# **The Gradient Descent Algorithm**

In order to minimize a convex function, we need to ﬁnd a stationary point. There are diﬀerent methods and heuristics to ﬁnd a stationary point. One possible approach is to start at an arbitrary point, and move along the gradient at that point towards the next point, and repeat until (hopefully) converging to a stationary point. We illustrate this in the ﬁgure below.



## **Direction and step size**

In general, one can consider a search for a stationary point as having two components: **the direction and the step size**. The direction decides which direction we search next, and the step size determines how far we go in that particular direction. Such methods can be generally described as starting at some arbitrary point and then at every step iteratively moving at direction by step size to the next point . In gradient descent, the direction we search is the negative gradient at the point, i.e. Thus, the iterative search of gradient descent can be described through the following recursive rule:

## **Choosing a step size**

?

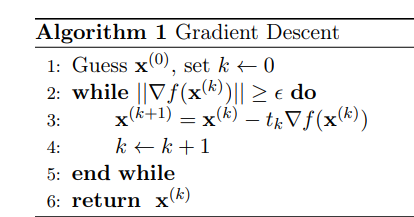
Given that the search for a stationary point is currently at a certain point x(k), how should we choose our step size tk? Since our objective is to minimize the function, one reasonable approach is to choose the step size in manner that will minimize the value of the new point, i.e. ﬁnd the step size that minimizes Since the step size of this approach is:

with assumption that can be computed analytically.

?

## **The algorithm**

Formally, given a desired precision we deﬁne the gradient descent as described below.



## **Implement Gradient Descent**

Implementing gradient descent algorithm for one feature, you will need three functions.

* Compute gradient implementing equation
* Compute cost implementing equation
* Gradient descent by utilizing compute gradient and compute cost

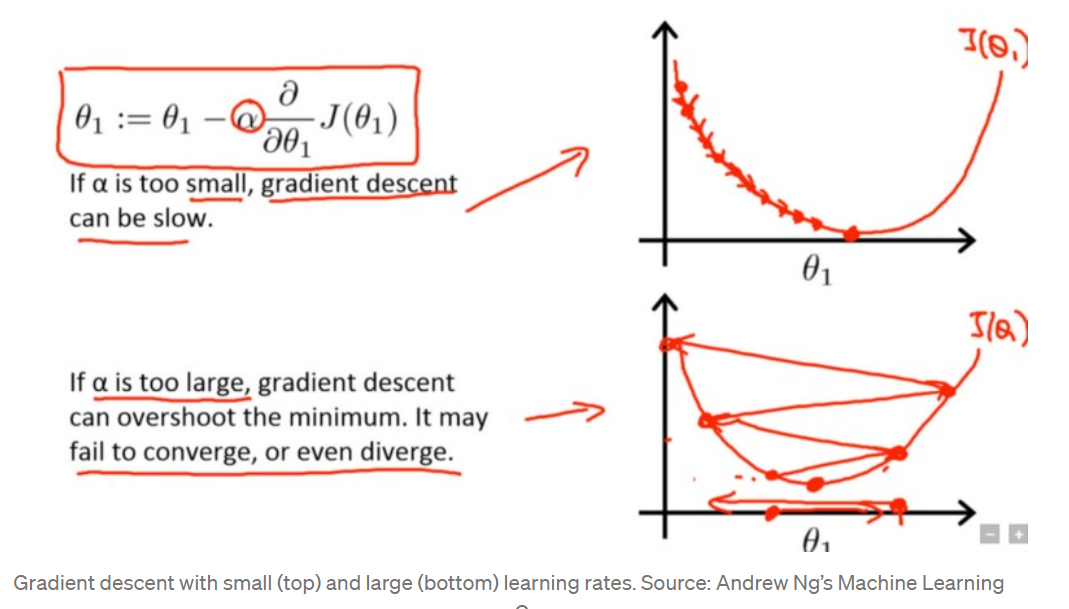
# **2. LEARNING RATE**

In machine learning there are two sort of parameters: machine learnable parameters and hyper-parameters.

Machine learnable parameters are the parameters that the algorithms learn/estimate on their own during training for a particular dataset.

Hyper-parameters are variables that machine learning engineers or data scientists provide precise values to regulate how algorithms learn and modify the model’s performance.

The learning rate, denoted by the symbol α, is a hyper-parameter used to govern the pace at which an algorithm updates or learns the values of a parameter estimate. In other words,the learning rate regulates the weights of ournetwork concerning the loss gradient. It indicates how often the network refreshes the notions it has learned.



# **3. Overfitting**

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.

# **4. Underfitting in Machine Learning**

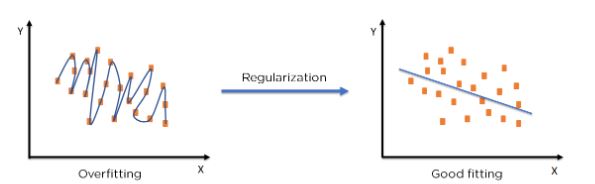
Underfitting refers to a model that can neither model the training data nor generalize to new data.

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.

# **5. REGULARIZATION**

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. Using Regularization, we can fit our machine learning model appropriately on a given test set and hence reduce the errors in it.  There are two main types of regularization techniques: Ridge Regularization and Lasso Regularization.



# **6. CLUSTERING**

This is a machine learning technique, which groups the unlabeled dataset. It can be defined as "A way of grouping the data points into different clusters, consisting of similar data points. The objects with the possible similarities remain in a group that has less or no similarities with another group." It does it by finding some similar patterns in the unlabeled dataset such as shape, size, color, behavior, etc., and divides them as per the presence and absence of those similar patterns. It is an unsupervised learning method, hence no supervision is provided to the algorithm, and it deals with the unlabeled dataset. The below diagram explains the working of the clustering algorithm. We can see the different fruits are divided into several groups with similar properties.

The clustering methods are broadly divided into Hard clustering (datapoint belongs to only one group) and Soft Clustering (data points can belong to another group also). But there are also other various approaches of Clustering exist. Below are the main clustering methods used in Machine learning:

1. Partitioning Clustering

2. Density-Based Clustering

3. Distribution Model-Based Clustering

4. Hierarchical Clustering

5. Fuzzy Clustering

## **Uses of Clustering**

Clustering has a myriad of uses in a variety of industries. Some common applications for clustering include the following:

• market segmentation

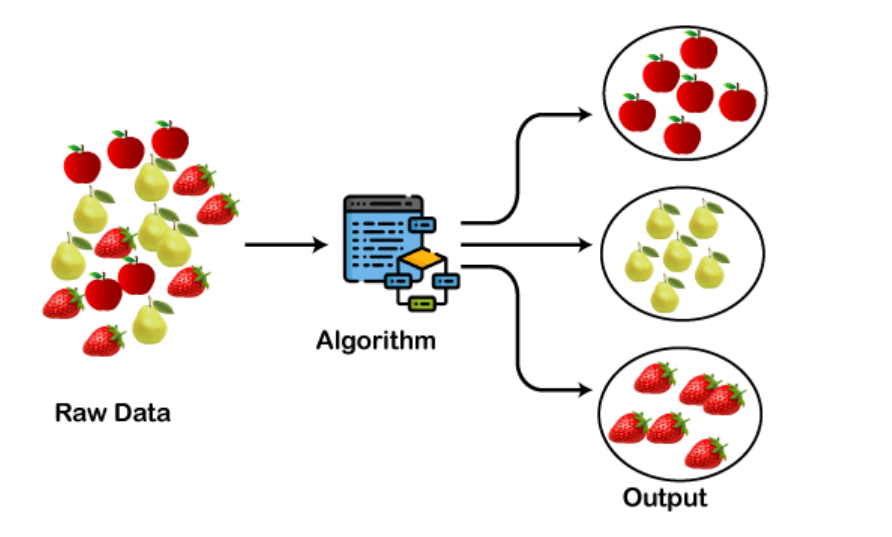
• social network analysis

• search result grouping

• medical imaging

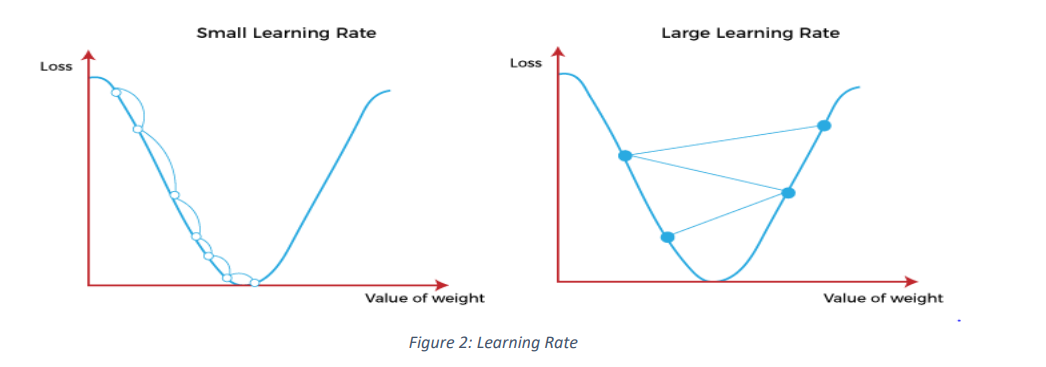
• image segmentation

• anomaly detection



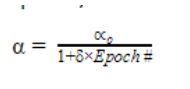
# **7. LEARNING RATE**

The learning rate is defined as the number of steps required to reach the minimum or lowest point. This is typically a small value that is evaluated and updated based on the cost function's behavior. When the learning rate is high, larger steps are taken, but there is a risk of exceeding the minimum. At the same time, a low learning rate demonstrates small step sizes, which reduces overall efficiency but provides the benefit of greater precision.

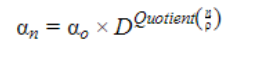


The learning rate, denoted by the symbol , is a hyper-parameter used to govern the pace at which an algorithm updates or learns the values of a parameter estimate. In other words, the learning rate regulates the weights of our neural network concerning the loss gradient. It indicates how often the network refreshes the notions it has learned. There are, however, various sorts of learning rate approaches:

• Decaying Learning Rate – The learning rate drops as the number of epochs/iterations increases in this learning rate technique.



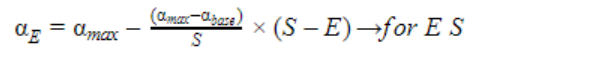
• Scheduled Drop Learning rate – The learning rate is lowered by a specified proportion at a specified frequency in the drop learning rate method, as opposed to the decay technique, where the learning rate declines repetitively.



• Cycling learning rate – The learning rate cyclically changes between a base rate and a maximum rate in this methodology. At a constant frequency, the learning rate varies in a triangular pattern between the maximum and base rates.

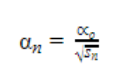


In the above equation, E is the learning rates for a given epoch, E is the epoch number, max and base are respectively the maximum and the base learning rates. S is the step size. Note that the above equation valid when E>S. for E≤S below equation can be used



• The Gradient Descent Method – is a well-known optimization approach for estimating model parameters in machine learning. The value of each parameter is originally assumed or assigned random values when training a model. The cost function is generated using the initial values, and the parameter estimations are improved over time so that the cost function eventually assumes a minimum value.

* Adaptive Learning rate- In this approach, the learning rate increases or decreases based on the gradient value of the cost function. For higher gradient value, the learning rate will be smaller and for lower gradient value, the learning rate will be larger. Hence, the learning decelerates and accelerates respectively at steeper and shallower parts of the cost function curve. The formula used in this approach is shown in the below equation.





In the above equation,𝛾 is a hyperparameter whose value is typically between 0.7 and 0.9

# 8. OVERFITTING

Overfitting is a bad machine learning behavior that occurs when a machine learning model predicts accurately for training data but not for new data. When data scientists use machine learning models to make predictions, the model is first trained on a known data set. The model then tries to predict outcomes for new data sets based on this information. An overfit model can produce inaccurate predictions and is unsuitable for all types of new data.

# 9. UNDERFITTING

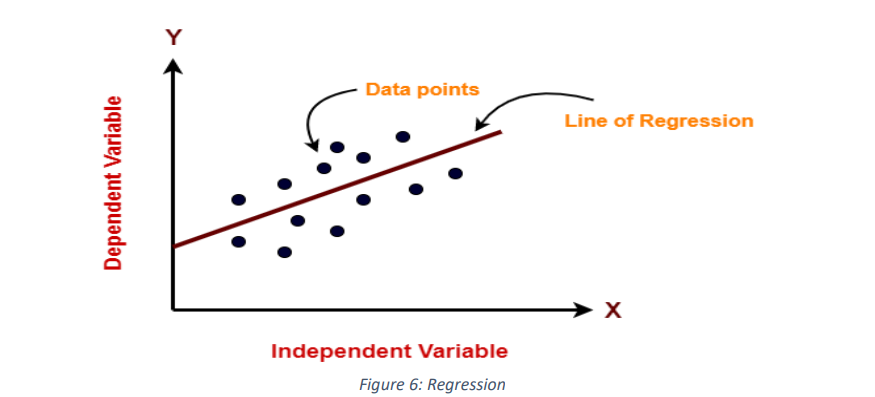
Underfitting is a data science scenario in which a data model is unable to accurately capture the relationship between the input and output variables, resulting in a high error rate on both the training set and unseen data. It happens when a model is overly simple, which can happen when a model requires more training time, more input features, or less regularization.



# **10. REGRESSION**

The technique of regression is used to investigate the relationship between independent variables or features and a dependent variable or outcome. It is used in machine learning as a method for predictive modeling, in which an algorithm is used to predict continuous outcomes. Some important types of regression which are given below:

* Linear Regression
* Logistic Regression
* Polynomial Regression
* Support Vector Regression
* Decision Tree Regression
* Forest Regression
* Ridge Regression
* Lasso Regression



# **Hyper-parameters tuning**

hyper-parameter tuning or optimization involves choosing a set of optimal hyper-parameters for a machine learning algorithm, while applying the optimized algorithm to any data set.it involves finding the configuration of hyperparameters that result in the best performance.

# **Feature Engineering:**

feature engineering or feature extraction is the process of using domain knowledge to extract features from raw data. It consists of four main steps which are feature creation, transformations, extraction and selection of features that are most conducive to create an accurate machine learning algorithm.

1. **Accuracy score:**

Accuracy score is a performance metrics for classification model that is defined as the ratio of true positives and true negatives to all positive and negative observations. It tells us how often our machine learning model will correctly predict an outcome from the total number of times it made predictions.

1. **Cross validation**

Cross validation is a technique for validating the model efficiency by training it on the subset of input data and testing it on previously unseen subset of input data. Some basic steps involved in cross validation are;

* Reserve a subset of the dataset as a validation set.
* Train the model using the training dataset.
* Evaluate the model performance using the validation set, if the model performs well with the validation set, go to the next step, else check for the issues.

# **One hot encoding**

one hot encoding is a technique that transforms categorical data into numerical data, it transforms strings into numbers so they can be applied to machine learning algorithms directly without any problems. One hot encoding creates new columns as much as the unique number of objects in the original column and assigns values of 0’s and 1’s where applicable.

1. **Train and test split**

Train test split is a technique used for evaluating the performance of a machine learning algorithm, it is used for both classification and regression problems and can be used for any supervised learning algorithm. The procedure involves dividing a dataset into two subsets, the first subset is used to fit the model and is referred to as the training dataset, while the second subset is used to evaluate the performance of the trained model on the output and is referred to as the test dataset. The scikit-learn python machine learning library provides an implementation of the train\_test\_split procedure.

# **REFERENCE**

[**Jason Brownlee**](https://machinelearningmastery.com/author/jasonb/)(2016) Overfitting and Underfitting With Machine Learning Algorithms,[**Machine Learning Algorithms**](https://machinelearningmastery.com/category/machine-learning-algorithms/)

For further reading on gradient descent and general descent methods please see Chapter 9 of the Convex Optimization book by Boyd and Vandenberghe.

https://www.simplilearn.com/tutorials/machine-learning-tutorial/regularization-in-machine-learning#:~:text=EngineerExplore%20Course-,What%20is%20Regularization%20in%20Machine%20Learning%3F,and%20prevent%20overfitting%20or%20underfitting

https://www.mygreatlearning.com/blog/understanding-learning-rate-in-machine-learning/

# **CONTRIBUTORS**

Margaret Oluwadare

Okoro Oluchi

Olubusayo Solola

Monsurat Onabajo

Olayemi Victoria Oloyede

Oluwadunsin Olajide

Olayemi Ibiloye

Anishere Mariam Adrola

Amaefula Maryann

Loveth Osuagwu