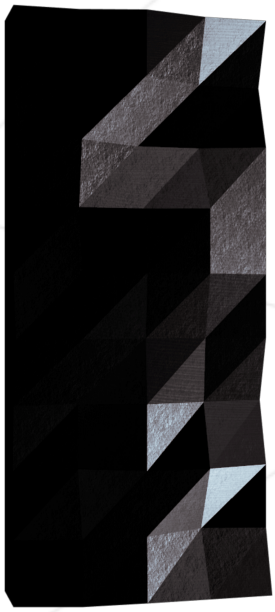


Introduction to Machine Learning

Linear Regression

Jeson Lee ljunzhen@42.us.org
Jesonleejunzhen.com

Summary: Learn the most basic type of regression model, linear regression!

Contents

`Eat, Sleep, Code, Repeat.`

Chapter I

Concepts of Linear Regression

What is linear regression?

Simple Linear regression is the most basic machine learning algorithm. Linear regression in simple terms is a statistical way of measuring the relationships between variables. Such as: as time increases, so does cost. It assesses how successfully a predictive variable does its job in predicting the outcome variable. It operates under the assumption that the two variables form a linear relationship.

There are two types of linear regression - Simple and Multiple.

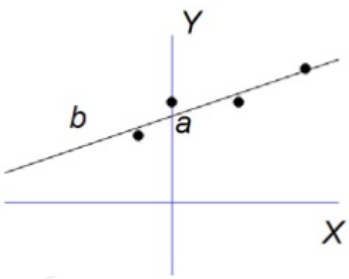
Linear regression equation
(without error)

$$\hat{Y} = bX + a$$

predicted values of *Y*

b = slope = rate of predicted ↑/↓ for Y scores for each unit increase in *X*

Y-intercept = level of *Y* when X is 0

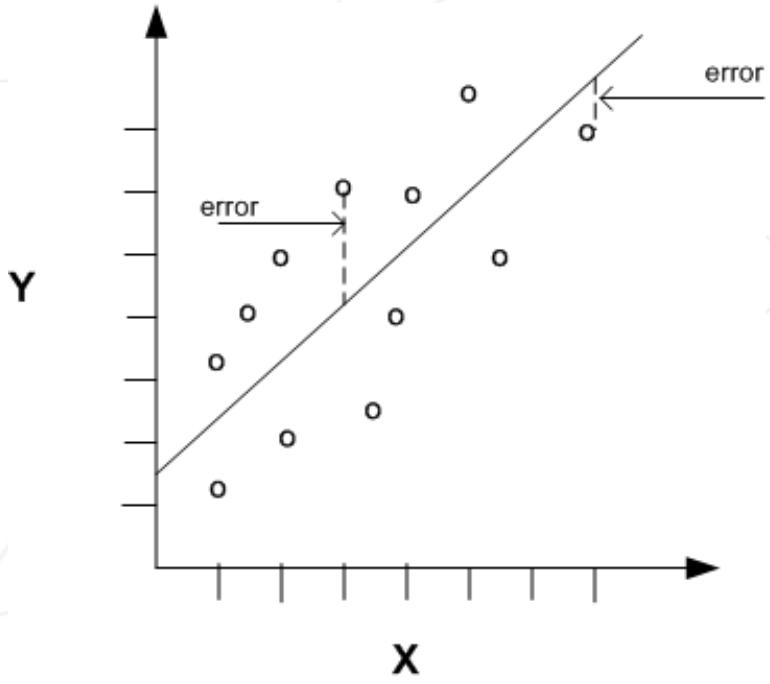


Cost Function

Cost functions are a way to determine how well the machine learning model has performed given the different values of each parameter. We can measure the accuracy of our hypothesis function by using a cost function. This takes an average difference (actually a fancier version of an average) of all the results of the hypothesis with inputs from Xs and the actual output Ys.

The cost function will be the sum of the least square methods.

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$$

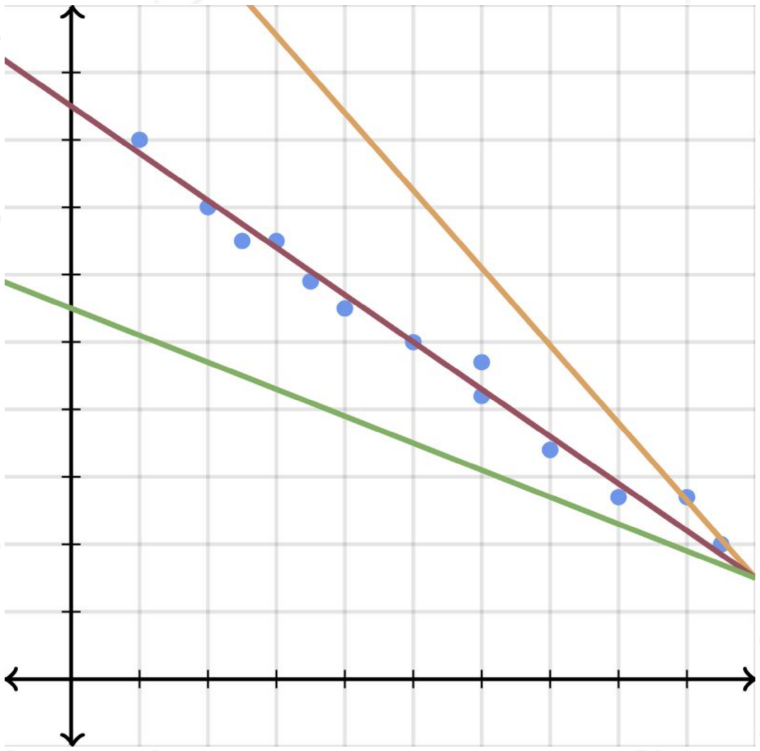


This function is otherwise called the **"Squared error function"**, or **"Mean squared error"**. But how do find the best fitting line with the smallest possible value of squared error function? We will talk about that later.

Gradient Descent

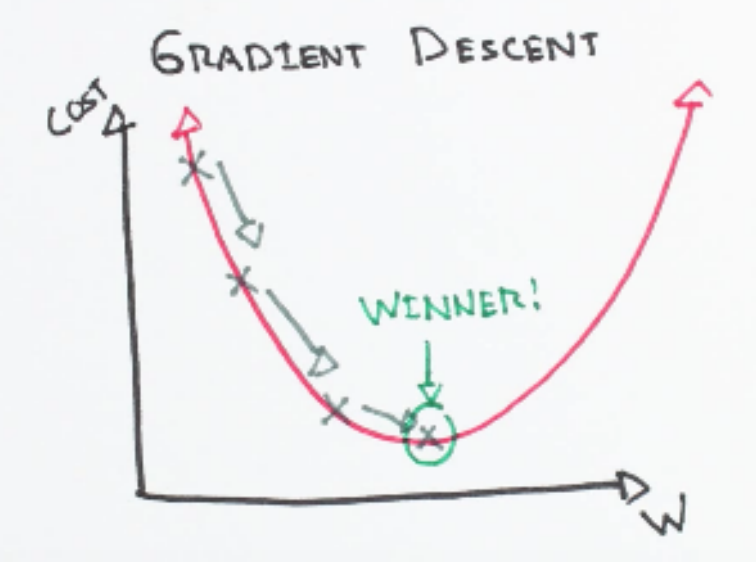
Gradient descent is an optimization algorithm that finds the optimal weights (a,b) that reduces prediction error. In simple terms, gradient descent helps find the best fitting line with the minimum error through iterative steps. In the picture below, try guessing which

color of the line has the minimum error?

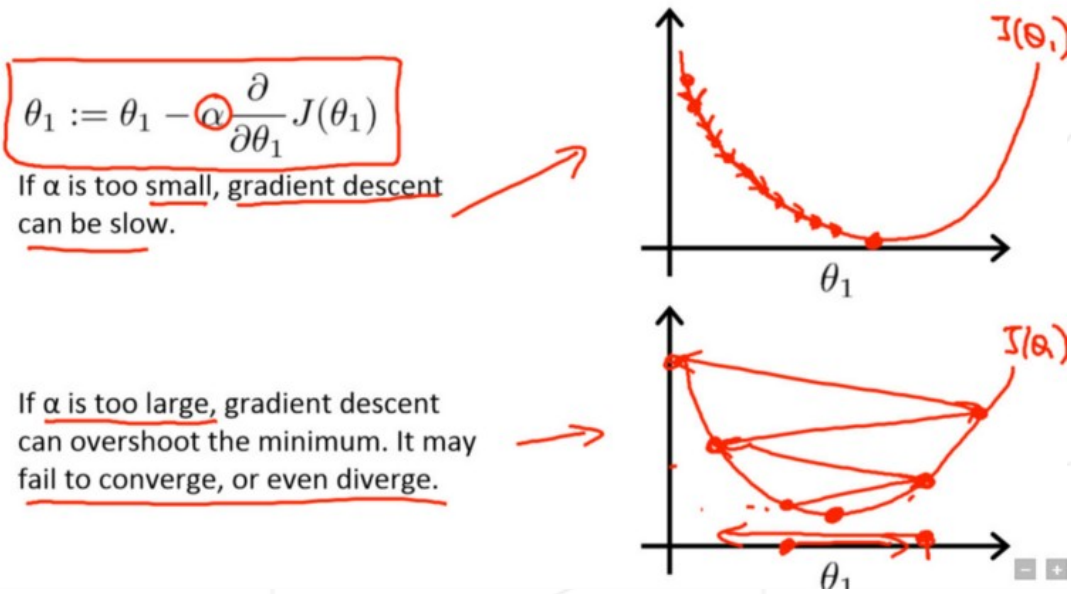


Given a function defined by a set of parameters, gradient descent starts with an initial set of parameter values and iteratively moves toward a set of parameter values that minimize the function. This iterative minimization is achieved using calculus, taking steps in the negative direction of the function gradient.

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$



We make steps down the cost function in the direction with the steepest descent. The size of each step is determined by the parameter , which is called the learning rate. Learning rate gives the rate of speed where the gradient moves during gradient descent. Setting it too high would make your path unstable, too low would make convergence slow. Putting it to zero means your model isn't learning anything from the gradients. The lower the value, the slower we travel along the downward slope.



Play with this interactive page to understand learning rate better:

https://developers.google.com/machine-learning/crash-course/fitter/graph

Variants of Gradient Descent

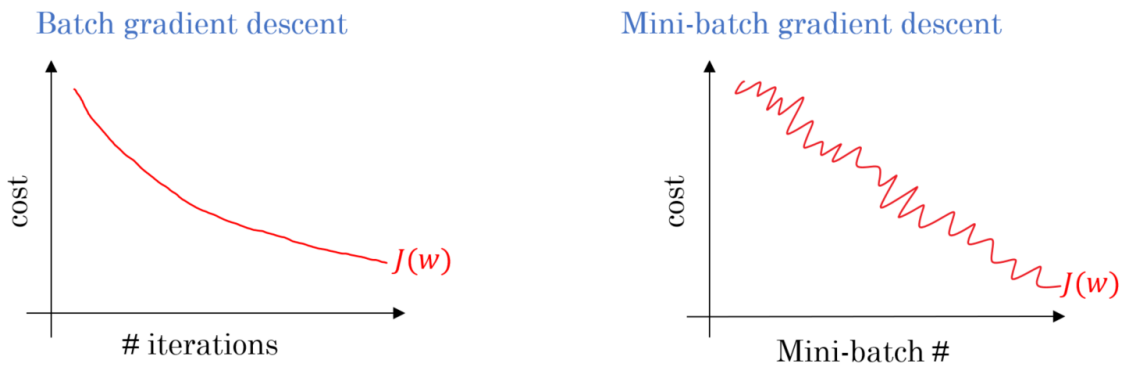
Do you know that there are different types of gradient descent? The main difference between them is the amount of data we use when computing the gradients for each learning step. The trade-off between them is the accuracy of the gradient versus the time complexity to perform each parameter update (learning step). Let's discuss the three variants of gradient descent algorithm:

Batch Gradient Descent

Sum up overall examples on each iteration when performing the updates to the parameters.

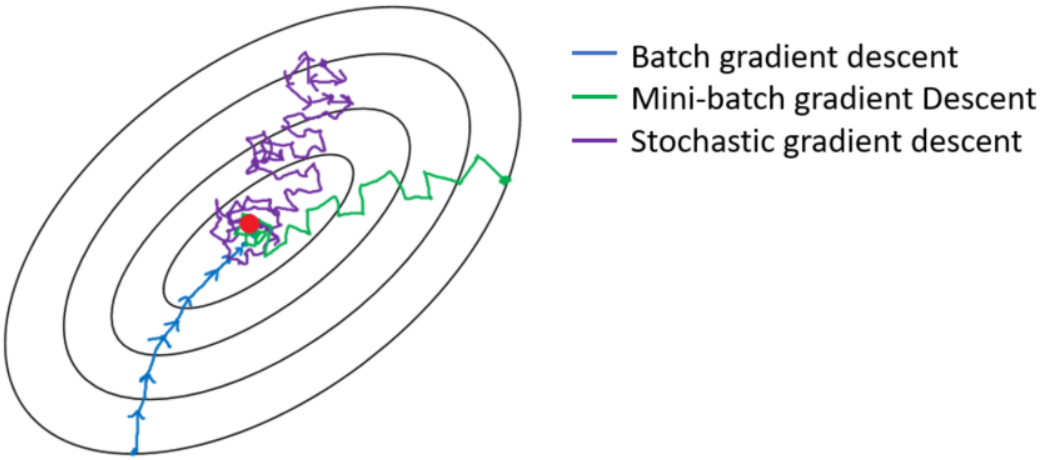
Mini-batch Gradient Descent

A mini-batch is typically between 10 and 1,000 examples, chosen at random. Mini-batch SGD reduces the amount of noise in SGD but is still more efficient than full-batch.



Stochastic Gradient Descent

Uses only a single example (a batch size of 1) per iteration. Given enough iterations, SGD works but is very noisy. The term "stochastic" indicates that the one example comprising each batch is chosen at random.



More information!

- Linear Regression:
  - Machine Learning Model: Simple Linear Regression
  - Linear Regression In Real Life
- Cost Function:
  - Machine Learning week 1: Cost Function, Gradient Desce, and Univariate Linear Regression
  - Understanding and Calculating the Cost Function for Linear Regression
- Gradient Descent:
  - Gradient Descent in a nutshell
  - An Introduction to Gradient Descent and Linear Regression
- Variant of Gradient Descent:
  - Gradient Descent Algorithm and Its Variants
  - Variants of Gradient Descent
- The first week of Andrew Ng's Machine Learning course on Coursera

Chapter II

Ask Your Peers

1. Explain what is a linear regression for and what kind of machine learning does it fall under?

2. What can we use linear regressions for? Give 3 examples.

3. Explain about cost function and the means squared error

4. Explain gradient descent

5. What is the difference between global and local minima?

6. What is a learning rate? Does it matter if we have a large or small learning rate?

7. What is the difference between mini-batch and stochastic gradient descent? What are the advantages and disadvantages of both?

Read more on Google if you aren't able to answer these questions with confidence!

Chapter III

Exercise 00: Linear Regression Project

	Exercise
Linear Regression Project	
Topics to study : `linear regression, scikit-learn, pandas, numpy, matplotlib`	
Files to turn in : `salary.py`	
Forbidden functions : `None`	
Notes : `n/a`	

Scenario:

Imagine you're looking for a job as a software engineer. And you spotted a company that you want to work for. You already have 4 years of experience working as a developer, and you want to see how much you can make working in this company based on the number of years of experience you have.

Well, the good news is you have a dataset of the salary of all the employees in the company and their number of years of experience respectively. This is the perfect opportunity to apply linear regression to predict your salary!

You must be able to visualize the result with a scatter plot by using Matplotlib library. The image below is the end result of the test set that is expected to be displayed.

Using a **Jupyter notebook** to build this project will be highly recommended. Download the dataset to begin. Happy hacking!

`Use the scikit-learn library to perform linear regression.`


Salary vs Experience (Test set)
120000
100000
80000
60000
40000
Salary
2
4
6
8
10
Years of Experience