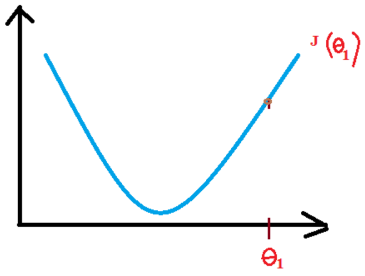
Gradient Descent

1. Gradient Descent

- Gradient descent is an optimization algorithm used to find the values of parameters (coefficients) of a function (f) that minimizes a cost function ($cost$).
- It is best used when the parameters cannot be calculated analytically (e.g. using linear algebra) and must be searched for by an optimization algorithm.



- The goal of gradient descent algorithm is to try different values of the coefficients theta to minimize the cost. For example: the minimum of the cost function above.

2. Gradient Descent Procedure

- Start off with an initial guess of the coefficient θ. For example: $\theta = 0$
- The cost of the coefficient is evaluated by computing the function f using coefficient θ.

$$cost = eval(f(\theta))$$

- In order to know the direction to move the coefficient value to minimize the cost, we need to compute the partial derivative of the cost function.

$$delta = derivative(cost\ function)$$

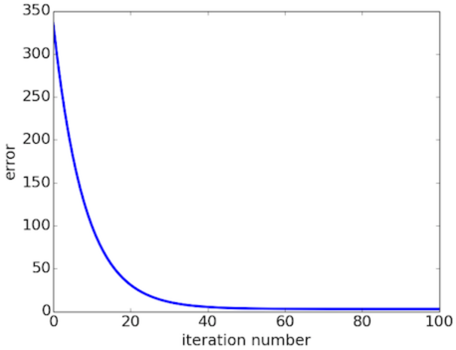
- To get the next coefficient that is closer to the minimum of the cost function, we need a learning rate parameter ($alpha$) that controls how much the coefficients can change on each update.

$$coefficient = coefficient - (alpha \times delta)$$

- This procedure is repeated until an optimal solution is found.

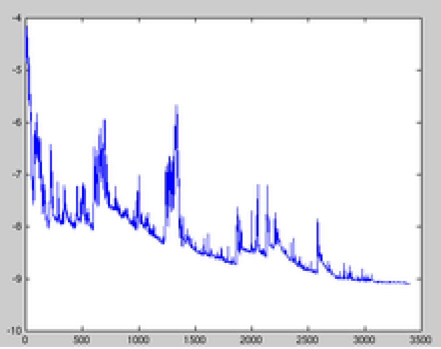
3. Batch Gradient Descent

Gradient Descent



- Batch gradient descent refers to calculating the derivative from all training data before calculating an update.
- Different algorithms have different representations and different coefficients. Many of them require a process of optimization to find the set of coefficients that result in the best estimate of the target function
- The cost function involves evaluating the coefficients in the machine learning model by calculating a prediction for each training instance in the dataset and comparing the predictions to the actual output values then calculating a sum or average error.
- From the cost function a derivative can be calculated for each coefficient
- The cost is calculated for a machine learning algorithm over the entire training dataset for each iteration of the gradient descent algorithm
- One iteration of the algorithm is called one batch and this form of gradient descent is referred to as batch gradient descent.
- Batch gradient descent is the most common form of gradient descent described in machine learning.

4. <u>Stochastic Gradient Descent</u>



- Gradient descent can be slow to run on very large datasets
- In situations when you have large amounts of data, you can use a variation of gradient descent called stochastic gradient descent.
- Stochastic gradient descent refers to calculating the derivative from each training data instance and calculating the update immediately.

Gradient Descent

- In this variation, the gradient descent procedure described above is run but **the update to the coefficients is performed for each training instance, rather than at the end of the batch of instances.**
- The first step of the procedure requires that the order of the training dataset is randomized to mix up the order that updates are made to the coefficients.
- The update procedure for the coefficients is the same as that above, except **the cost is not summed or averaged over all training patterns, but instead calculated for one training pattern**
- The learning can be much faster with stochastic gradient descent for very large training datasets and often you only need a small number of passes through the dataset to reach a good or good enough set of coefficients