Gradient Descent, Methods and variations

Mariam Ali

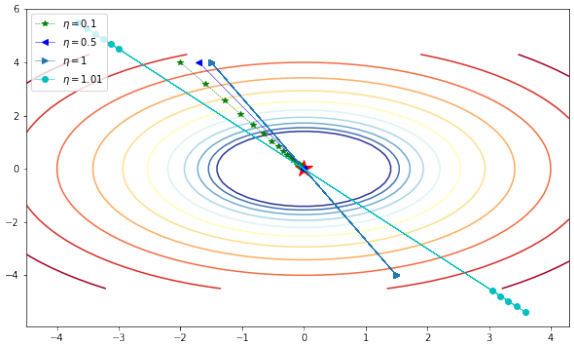
March 2019

1 Motivation

Gradient descent is an iterative algorithm that aims to minimize a function by taking increments or steps proportional to the negative of the derivative -more generally the gradient- of that function with respect to certain parameters and in the scope of Machine learning this function is the lost function or simply, the Error E(θ) between the predicted points and the labels. The main difference between gradient descent and usual minimizing techniques is the update step, as the main parameter of the predicting model θ is updated in the following manner:

$$v_t = \eta_t \nabla_\theta E(\theta)$$
$$\theta_{t+1} = \theta_t - v_t(1)$$

in the previous equation η_t is called the learning rate, which is the parameters that monitors how would the gradient affect the increments taken as the smaller the η the smaller the effect of the gradient value and more steps would be taken to approach the minima. And vice vers, if the η is too large the algorithm might have large steps which could cause to skip the minima and might even lead to divergence of the method. the effect of η shows in the following figure generated by python notebooks of the review of by Mehta et al.



Learning rate Variations

A final note on this point is that the learning rate η has an upper bound of the optimal learning rate $\eta_o pt$ which is the rate that leads the algorithm to converge directly to the minima, if the η is a value greater than the optimal value but less than twice it it causes the gradient descent to oscillate above and below the minima till it converges but if the η skips the limit of 2 $\eta_o pt$ i.e. becomes larger than twice that value, the Gradient descent diverges. (LeCun et al., 1998b).

2 Variations of Gradient descent

The previously mentioned method is the simplest version of gradient descent commonly known as the batch gradient descent because of the usual incremental manner, another version of gradient descent is:

2.1 stochastic gradient descent

which is applied to the type of data that have a random pattern (random probability distribution). One huge drawback of batch (usual) gradient is that under very large number of data points (m) it's very computationally expensive as usually the cost function is a sum from 1 to m which can go to very high orders meaning that for each iteration there's a sum of very large number of terms then for the next step the whole process is conducted again. The main difference between the usual gradient descent and the stochastic gradient descent is the way the cost function is defined differently, in GD (gradient descent) it's only a function of the model's parameter θ while in SGD (stochastic gradient descent the cost function is defined to measure how well the model predicts with respect to a single example (x_i, y_i). So what SGD basically does is that it goes through each training example and find the minimum parameter of the cost function which is much faster and less computationally heavy. A final difference is that the data is already random and being randomly shuffled further more helps diverging to the minima in an even faster manner. A third type of gradient descent that has even a better convergence method is:

2.2 Mini-Batch gradient descent

It's conducted through batching the data into small batches on which the algorithm is done but also with going through each pair of data (x_i, y_i), thus the minimizing parameters are reached in an easy quick manner and also the highly computational procedures are avoided.

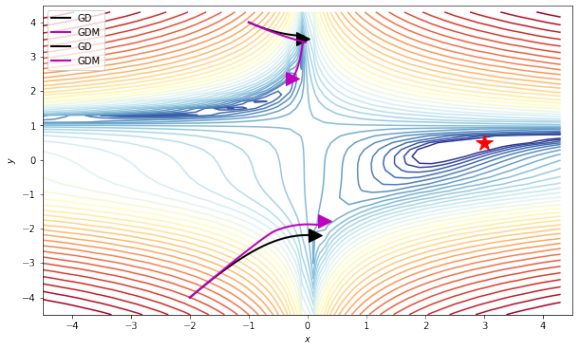
2.3 optimization methods

Gradient Descent has various optimization methods that could direct the algorithm towards the correct direction of minimization to avoid having the curves to be more steep in a certain direction than the others. A method that will be briefly adopted here is the momentum adding method, it adds a term that helps moving the direction of SGD to be quickly reaching the minimum in the following manner:

$$v_t = \gamma v_{t-1} + \eta_t \nabla_\theta E(\theta)$$

$$\theta_{t+1} = \theta_t - v_t(2)$$

This γ term is usually set between zero and 1 and usually called the momentum parameter. Momentum gradient descent can be visulaized in the following figure generated by python notebooks of the review of by Mehta et al.



Momentum Gradiend descent