Stochastic vs Batch Gradient Descent

Divakar Kapil
Jan 7, 2019 · 5 min read

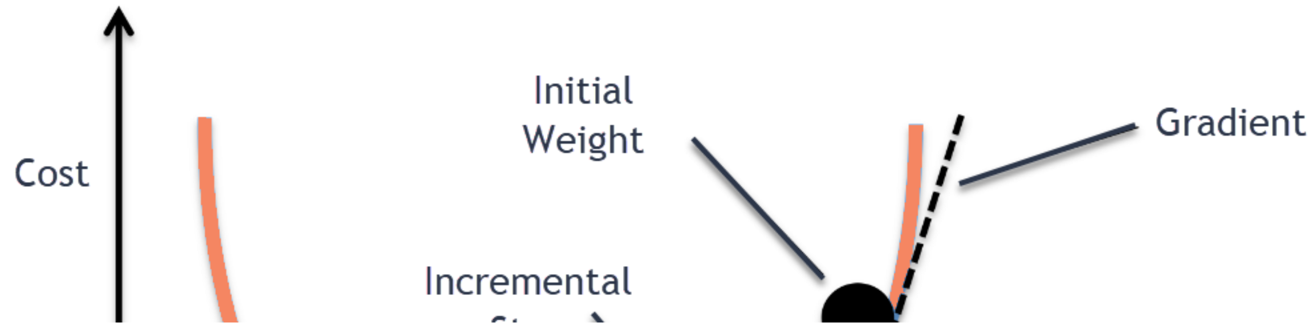
One of the first concepts that a beginner comes across in the field of deep learning is gradient descent followed by various ways in which it can be implemented. Gradient descent is one of the most important concept used in the training of neural networks for supervised learning. Hence, it is important to understand it and the different ways in which it is to be carried out on the training sets.

This post mostly deals with the different ways in which gradient descent is implemented on a training set. Thus, I will briefly go over the definition of the concept and then explain the advantages and disadvantages of all the possible ways.

Gradient Descent

This is an iterative optimization algorithm for finding the minimum of a function. The algorithm takes steps proportional to the negative gradient of the function at the current point [1]. In deep learning neural networks are trained by defining a loss function and optimizing the parameters of the network to obtain the minimum of the function. the optimization is dne using the gradient descent algorithm which operates in these two steps:

1. Compute the slope (gradient) that is first order derivative of the function at the current point
2. Move in the opposite direction of the slope increase from the current point by the computed amount



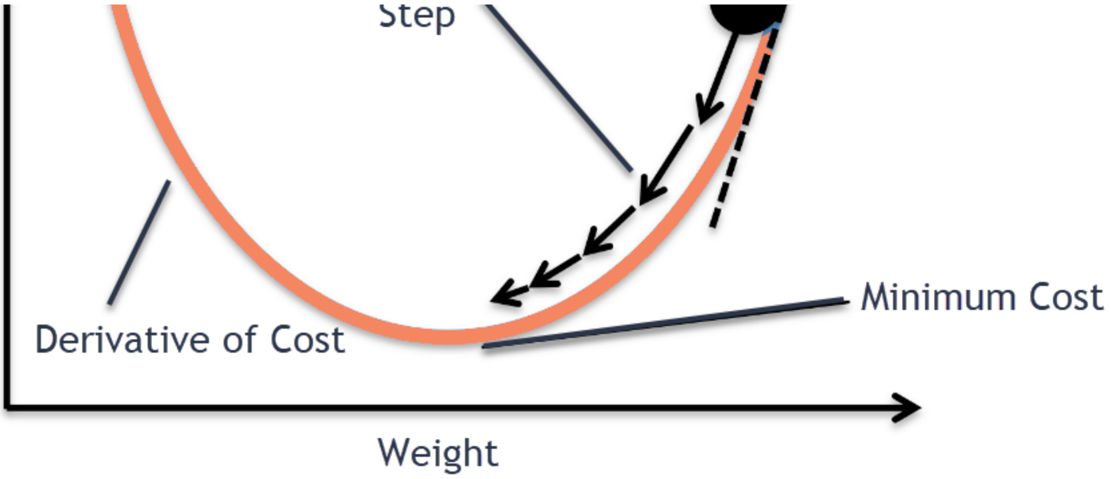


Figure1: Gradient Descent Algorithm [2]

So, the idea is to pass the training set through the hidden layers of the neural network and then update the parameters of the layers by computing the gradients using the training samples from the training dataset. This procedure can be done in the following ways:

Stochastic Gradient Descent

In this method **one training sample (example)** is passed through the neural network at a time and the parameters (weights) of each layer are updated with the computed gradient. So, at a time a single training sample is passed through the network and its corresponding loss is computed. The parameters of all the layers of the network are updated after every training sample. For example, if the training set contains 100 samples then the parameters are updated 100 times that is one time after every individual example is passed through the network. Following is the gradient descent equation and for stochastic gradient descent it is iterated over 'n' times for 'n' training samples in the training set.

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta).$$

Figure2: Gradient Descent Equation [3]

Here, (Theta(j)) corresponds to the parameter, (alpha) is the learning rate that is the step size multiplied by the derivative of the function by which to move on the loss function curve toward the minima.

Advantages of Stochastic Gradient Descent

1. It is easier to fit into memory due to a single training sample being processed by the network

2. It is computationally fast as only one sample is processed at a time

3. For larger datasets it can converge faster as it causes updates to the parameters more frequently

4. Due to frequent updates the steps taken towards the minima of the loss function have oscillations which can help getting out of local minimums of the loss function (in case the computed position turns out to be the local minimum)

Disadvantages of Stochastic Gradient Descent

1. Due to frequent updates the steps taken towards the minima are very noisy. This can often lead the gradient descent into other directions.

2. Also, due to noisy steps it may take longer to achieve convergence to the minima of the loss function

3. Frequent updates are computationally expensive due to using all resources for processing one training sample at a time

4. It loses the advantage of vectorized operations as it deals with only a single example at a time

Batch Gradient Descent

The concept of carrying out gradient descent is the same as stochastic gradient descent. The difference is that instead of updating the parameters of the network after computing the loss of every training sample in the training set, the parameters are updated once that is after all the training examples have been passed through the network. For example, if the training dataset contains 100 training examples then the parameters of the neural network are updated once. The equation in Figure2 is iterated over only once.

Advantages of Batch Gradient Descent

1. Less oscillations and noisy steps taken towards the global minima of the loss function due to updating the parameters by computing the average of all the training samples rather than the value of a single sample

2. It can benefit from the vectorization which increases the speed of processing all training samples together
3. It produces a more stable gradient descent convergence and stable error gradient than stochastic gradient descent
4. It is computationally efficient as all computer resources are not being used to process a single sample rather are being used for all training samples

Disadvantages of Batch Gradient Descent

1. Sometimes a stable error gradient can lead to a local minima and unlike stochastic gradient descent no noisy steps are there to help get out of the local minima
2. The entire training set can be too large to process in the memory due to which additional memory might be needed
3. Depending on computer resources it can take too long for processing all the training samples as a batch

Mini Batch Gradient Descent Batch : A Compromise

This is a mixture of both stochastic and batch gradient descent. The training set is divided into multiple groups called batches. Each batch has a number of training samples in it. At a time a single batch is passed through the network which computes the loss of every sample in the batch and uses their average to update the parameters of the neural network. For example, say the training set has 100 training examples which is divided into 5 batches with each batch containing 20 training examples. This means that the equation in figure2 will be iterated over 5 times (number of batches).

This ensures the following advantages of both stochastic and batch gradient descent are used due to which Mini Batch Gradient Descent is most commonly used in practice.

1. Easily fits in the memory
2. It is computationally efficient
3. Benefit from vectorization
4. If stuck in local minimums, some noisy steps can lead the way out of them
5. Average of the training samples produces stable error gradients and convergence

. . .

In this post I briefly went over the gradient descent algorithm with detailed explanations on the various methods of gradient descent. I hope this post provides some clarity on the differences between stochastic gradient descent and batch gradient descent.

If you like this post or found it useful please leave a clap!

If you see any errors or issues in this post, please contact me at divakar239@icloud.com and I will rectify them.

References

[1] https://en.wikipedia.org/wiki/Gradient_descent

[2] https://www.oreilly.com/library/view/learn-arcore-/9781788830409/e24a657a-a5c6-4ff2-b9ea-9418a7a5d24c.xhtml

[3] http://eric-yuan.me/linear-regression/

Machine Learning Neural Networks Gradient Descent Optimization Beginner

Medium About Help Legal