

Stochastic gradient descent Gradient Descent Machine Learning +1

What's the difference between gradient descent and stochastic gradient descent?

Answer

Request ▾

Follow 99 Comment Share 2 Downvote

Promoted by Udacity.com

Launch a career in machine learning today.

Get your Machine Learning Engineer Nanodegree in this course, co-created by Google.

Learn More at Udacity.com

9 Answers

**Abhishek Shivkumar**, Research Engineer - Deep Learning

Updated Sep 20, 2013

For a quick simple explanation:

In both gradient descent (GD) and stochastic gradient descent (SGD), you update a set of parameters in an iterative manner to minimize an error function.

While in GD, you have to run through **ALL** the samples in your training set to do a single update for a parameter in a particular iteration, in SGD, on the other hand, you use **ONLY ONE** training sample from your training set to do the update for a parameter in a particular iteration.

Thus, if the number of training samples are large, in fact very large, then using gradient descent may take too long because in every iteration when you are updating the values of the parameters, you are running through the complete training set. On the other hand, using SGD will be faster because you use only one training sample and it starts improving itself right away from the first sample.

SGD often converges much faster compared to GD but the error function is not as well minimized as in the case of GD. Often in most cases, the close approximation that you get in SGD for the parameter values are enough because they reach the optimal values and keep oscillating there.

If you need an example of this with a practical case, check Andrew NG's notes here where he clearly shows you the steps involved in both the cases. <http://cs229.stanford.edu/notes/...>

53.7k Views · View Upvotes

Upvote 301

Downvote Comments 5+

**Sebastian Raschka**, Author of Python Machine Learning, researcher applying ML to computational bio.

Written Nov 18, 2015

In order to explain the differences between alternative approaches to estimating the parameters of a model, let's take a look at a concrete example: Ordinary Least Squares (OLS) Linear Regression. The illustration below shall serve as a quick reminder to recall the different components of a simple linear regression model:

Related Questions

[Why is the trajectory of the estimated parameters in each iteration of gradient descent zig-zagged?](#)[What is the difference between online gradient descent and stochastic gradient descent?](#)[Can you mention relatively recent usage of gradient descent in ML research?](#)[Difference between stochastic gradient descent and online learning?](#)[How do we decide minibatch size when doing stochastic gradient descent for neural network? Does big minibatch hurt generalization? How do we e...](#)[How do I draw steps in a gradient descent by MATLAB or Python?](#)[Is stochastic gradient descent online? How do I intuitively prove that it converges?](#)[In batch gradient descent with regularization, how should I compute the gradient?](#)[What is the formal relationship between batch gradient descent and a fixed-point theorem?](#)[How does stochastic gradient descent work?](#)

More Related Questions

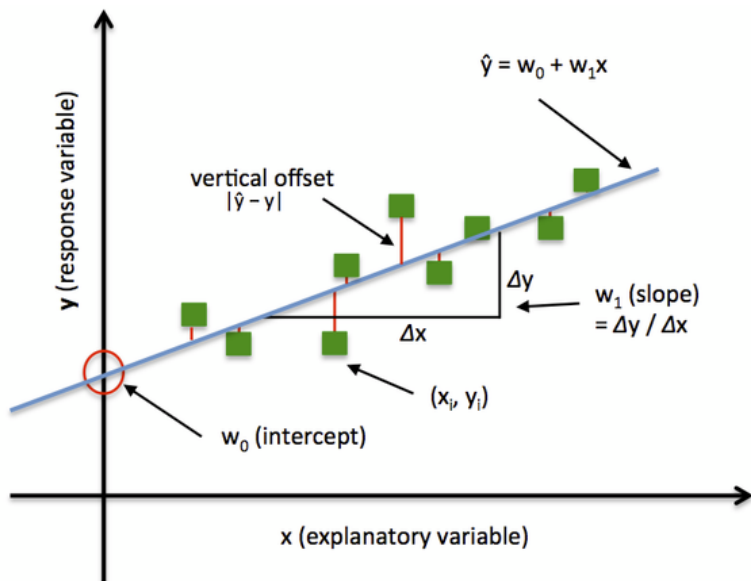
Question Stats

99 Followers

64,553 Views

Last Asked Aug 8, 2015

Edits



with

$$y = w_0x_0 + w_1x_1 + \dots + w_mx_m = \sum_{i=1}^n = \mathbf{w}^T \mathbf{x}$$

In Ordinary Least Squares (OLS) Linear Regression, our goal is to find the line (or hyperplane) that minimizes the vertical offsets. Or, in other words, we define the best-fitting line as the line that minimizes the sum of squared errors (SSE) or mean squared error (MSE) between our target variable (y) and our predicted output over all samples i in our dataset of size n .

$$SSE = \sum_i (\text{target}^{(i)} - \text{output}^{(i)})^2$$

$$MSE = \frac{1}{n} \times SSE$$

Now, we can implement a linear regression model for performing ordinary least squares regression using one of the following approaches:

- Solving the model parameters analytically (closed-form equations)
- Using an optimization algorithm (Gradient Descent, Stochastic Gradient Descent, Newton's Method, Simplex Method, etc.)

GRADIENT DESCENT (GD)

Using the Gradient Decent (GD) optimization algorithm, the weights are updated incrementally after each epoch (= pass over the training dataset).

The cost function $J(\cdot)$, the sum of squared errors (SSE), can be written as:

$$J(\mathbf{w}) = \frac{1}{2} \sum_i (\text{target}^{(i)} - \text{output}^{(i)})^2$$

The magnitude and direction of the weight update is computed by taking a step in the opposite direction of the cost gradient

$$\Delta w_j = -\eta \frac{\partial J}{\partial w_j},$$

where η is the learning rate. The weights are then updated after each epoch via the following update rule:

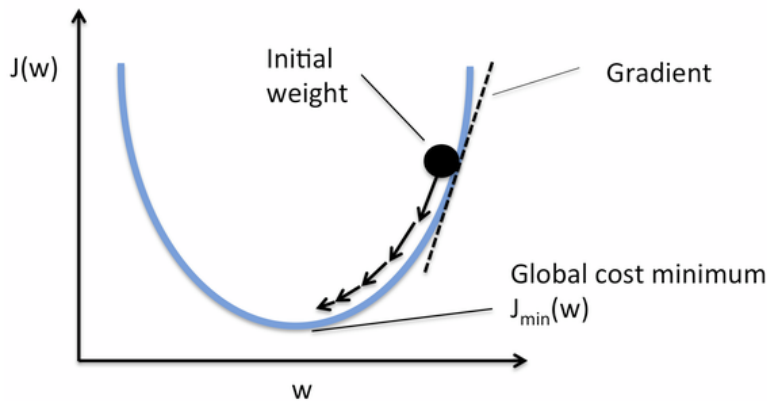
$$\mathbf{w} := \mathbf{w} + \Delta \mathbf{w},$$

where $\Delta \mathbf{w}$ is a vector that contains the weight updates of each weight coefficient w , which

are computed as follows:

$$\begin{aligned}\Delta w_j &= -\eta \frac{\partial J}{\partial w_j} \\ &= -\eta \sum_i (\text{target}^{(i)} - \text{output}^{(i)}) (-x_j^{(i)}) \\ &= \eta \sum_i (\text{target}^{(i)} - \text{output}^{(i)}) x_j^{(i)}.\end{aligned}$$

Essentially, we can picture GD optimization as a hiker (the weight coefficient) who wants to climb down a mountain (cost function) into a valley (cost minimum), and each step is determined by the steepness of the slope (gradient) and the leg length of the hiker (learning rate). Considering a cost function with only a single weight coefficient, we can illustrate this concept as follows:



STOCHASTIC GRADIENT DESCENT (SGD)

In GD optimization, we compute the cost gradient based on the complete training set; hence, we sometimes also call it *batch GD*. In case of very large datasets, using GD can be quite costly since we are only taking a single step for one pass over the training set -- thus, the larger the training set, the slower our algorithm updates the weights and the longer it may take until it converges to the global cost minimum (note that the SSE cost function is convex).

In Stochastic Gradient Descent (SGD; sometimes also referred to as *iterative* or *on-line* GD), we don't accumulate the weight updates as we've seen above for GD:

- for one or more epochs:
 - for each weight j
 - $w_j := w + \Delta w_j$, where: $\Delta w_j = \eta \sum_i (\text{target}^{(i)} - \text{output}^{(i)}) x_j^{(i)}$

Instead, we update the weights after each training sample:

- for one or more epochs, or until approx. cost minimum is reached:
 - for training sample i :
 - for each weight j
 - $w_j := w + \Delta w_j$, where: $\Delta w_j = \eta (\text{target}^{(i)} - \text{output}^{(i)}) x_j^{(i)}$

Here, the term "stochastic" comes from the fact that the gradient based on a single training sample is a "stochastic approximation" of the "true" cost gradient. Due to its stochastic nature, the path towards the global cost minimum is not "direct" as in GD, but may go "zig-zag" if we are visualizing the cost surface in a 2D space. However, it has been shown that SGD almost surely converges to the global cost minimum if the cost function is convex (or pseudo-convex)[1]. Furthermore, there are different tricks to improve the GD-based learning, for example:

... (more)

Upvote 247

Downvote Comments 12+



Sean Currey, Aerospace Engineer

Written Jan 19, 2013

To add on to Abhishek's excellent response:

Gradient descent is *deterministic*, which means that every time you run GD for a given training set, you will get the same optimum in the same number of iterations. Stochastic gradient descent is, well, *stochastic*. Because you are no longer using your entire training set at once, and instead picking one or more examples at a time in some likely random fashion, each time you run SGD you will obtain a different optimum and a unique cost vs. iteration history.

20.8k Views · View Upvotes

Upvote 76

Downvote Comment 1



Sathya Narayanan Ravi, Regularizer, Dualizer and a casual Parallelizer

Updated Aug 9 · Upvoted by Justin Rising, MSE in CS, PhD in Statistics

In the Gradient Descent method, one computes the direction that decreases the objective function the most in the case of minimization problems. But sometimes this can be quite costly. In most Machine Learning for example, the objective function is often the cumulative sum of the error over the training examples. But the size of the training examples set might be very large and hence computing the actual gradient would be computationally expensive.

In Stochastic Gradient (Descent) method, we compute an estimate or approximation to this direction. The most simple way is to just look at one training example (or subset of training examples) and compute the direction to move only on this approximation. It is called as Stochastic because the approximate direction that is computed at every step can be thought of a random variable of a stochastic process. This is mainly used in showing the convergence of this algorithm.

There might be many reason but one reason as to why SG is preferred in Machine Learning is because it helps the algorithm to skip some local minima. Though this is not a theoretically sound reason in my opinion, the optimal points that are computed using SG are empirically better than the GD method often.

17.1k Views · View Upvotes

Upvote 27

Downvote Comments 2



Arvind Rapaka, CEO SpotDy Inc

Written May 8

In the standard gradient descent algorithm, the parameters are updated on the entire data set iteratively. On a large data set, standard gradient descent algorithm not very efficient to find the global minimum of parameters (weights).

Algorithm:

Repeat till you converge {

$$\theta_j := \theta_j + \alpha \sum_{i=1}^m (y^{(i)} - h_{\theta}(x^{(i)})) x_j^{(i)} \quad (\text{for every } j).$$

}

Where α is the learning rate. Smaller α is recommended so that we don't overshoot the global minimum while calculating the parameter.

It evident that on every step, the entire training set is used and that is the reason it is also called batch gradient descent. Such calculation is expensive on a huge dataset as it has to store the intermediate values in memory or read from disk on each iteration.

Following is the graphical interpretation.

Ask or Search Quora

Ask Question

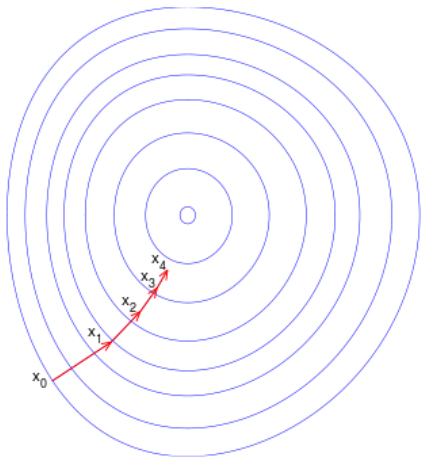
Read

Answer¹

Notifications¹



Ashutosh



In stochastic gradient descent, the algorithm uses a single or a few training examples to calculate the parameters.

Algorithm:

1> Randomly shuffle the training set.

2> Iteratively Calculate the parameters:

```

Loop {
  for i=1 to m, {
     $\theta_j := \theta_j + \alpha (y^{(i)} - h_{\theta}(x^{(i)})) x_j^{(i)}$     (for every  $j$ ).
  }
}

```

In SGD the learning rate α is much smaller compared to the standard gradient descent due to higher variance.

Random shuffle is required because order data set can bias the gradient.

3.4k Views · View Upvotes

Upvote 2 Downvote Comment 1



Kim Xu, Data Scientist

Written Nov 11

stochastic gradient descent is only a batch version of the gradient descent. In SGD, the update was done after a single example or a small batch of example.

This is superior when you have large sample size and somehow redundant samples.

133 Views

Upvote Downvote Comment



Atul Kulkarni, I dabble with ML algorithms...

Written Sep 26, 2013

The video below gives exactly what you are looking for with the algorithm.

Ask or Search Quora

Ask Question

Read

Answer¹

Notifications¹




Ashutosh

Also, what Abhishek wrote is spot on.

11.3k Views · View Upvotes


Upvote 9DownvoteComment

**Vivek Poonia**, pursuing research in ML.
Written Jul 30, 2015

Abhishek has explained perfectly. Generally stochastic GD is preferred for being faster as it is optimizing parameter on one training example at a time till it converges. On the other hand, gradient descent(called Batch GD) optimizes parameter on whole training set every iteration till convergence. This makes Batch GD slow but deterministic. As Abhishek has again pointed out, Andrew Ng's Lecture notes are elaborate.

6.2k Views · View Upvotes

Upvote 1DownvoteComment

**Humoyun Ahmedov**, I am passionate learner and lover of ML
Written Feb 29


I think Andrew NG's video on this will enhance your understanding much better.

3.1k Views · View Upvotes


Upvote 9DownvoteComment

1 Answer Collapsed (Why?)

Top Stories from Your Feed

Mohit Ahuja upvoted this · Mon

What happens if a person with lots of black money in ₹500/1000 notes uses many persons' bank accounts to convert it to white?

**Bala Senthil Kumar**
Written Nov 10 · Upvoted by Mohit Ahuja

Of course people will try this, and of course a lot of people with bank accounts and hardly any money in it will be prime targets. They will even be offered money for their services

Jessica Su upvoted this · Tue


What's the most embarrassing misconception you've ever held?

**Sameer Bobade**, Enjoy the life and laugh at the past.
Updated Jul 29 · Upvoted by Jessica Su




Answer written · Apr 21

What was the most ruthless spoiler someone gave you?

**Crystal Ng**, I watch movies..
Written Apr 21



Ask or Search Quora	Ask Question	Read	Answer ¹	Notifications ¹	 Ashutosh
Read In Feed	you need petrol you drive to them, pay up	Read In Feed	**SPOILER ALERT** I was planning to	Read In Feed	