Differences between GD and SGD:

- 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 or SUBSET of training sample from your training set to do the update for a parameter in a particular iteration. If you use SUBSET, it is called Minibatch Stochastic gradient Descent.
- 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.
-

Comparison how Gradient Descent and Stochastic Gradient Descent differs:

- Convergence (convergence speed)

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 or SUBSET of training sample from your training set to do the update for a parameter in a particular iteration. If you use SUBSET, it is called Minibatch Stochastic gradient Descent.

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.

Use case (when to use which algorithm)

Gradient Descent Algorithm

- if training examples are not linearly separable, the delta rule converges toward a best-fit approximation
- use gradient descent to find the weights that best fit the training examples - basis of Backpropagation Algorithm
- Given above error definition, the error surface must be parabolic with a single global minimum.

Stochastic gradient descent Algorithm

- It is used when the training data size is huge because GD may be infeasible in such case.
- If the training data set has many redundant data instances, stochastic gradients may be so close to the true gradient $\nabla f(\mathbf{x})$∇f(x) that a small number of iterations will find useful solutions to the optimization problem.

Standard Gradient descent updates the parameters only after each epoch i.e. after calculating the derivatives for all the observations it updates the parameters. This phenomenon may lead to the following caveats.

- It can be very slow for very large datasets because only one-time _update_for each epoch so large number of epochs is required to have a substantial number of updates.
- For large datasets, the vectorization of data doesn’t fit into memory.
- For non-convex surfaces, it may only find the local minimums.

Now let see how different variations of gradient descent can address these challenges.

Stochastic gradient descent

The standard gradient descent algorithm updates the parameters $\theta\theta$ of the objective $J(\theta)J(\theta)$ as,

$\theta=\theta-\alpha\nabla\theta E[J(\theta)]\theta=\theta-\alpha\nabla\theta E[J(\theta)]$

where the expectation in the above equation is approximated by evaluating the cost and gradient over the full training set. Stochastic Gradient Descent (SGD) simply does away with the expectation in the update and computes the gradient of the parameters using only a single or a few training examples. The new update is given by,

$\theta=\theta-\alpha\nabla\theta J(\theta;x(i),y(i))\theta=\theta-\alpha\nabla\theta J(\theta;x(i),y(i))$

with a pair $(x(i),y(i))(x(i),y(i))$ from the training set.

Disadvantages of SGD

- Because of the greedy approach, it only approximates (stochastics) the gradient.
- Due to frequent fluctuations, it will keep overshooting near to the desired exact minima.