Running batch gradient descent can be computationally quite costly in such scenarios which have a very large dataset since we need to reevaluate the whole training dataset each time we take one step towards the global minimum.

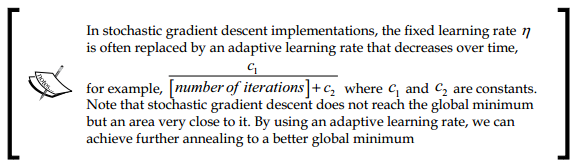
A popular alternative to the batch gradient descent algorithm is stochastic gradient descent, sometimes, also called iterative or on-line gradient descent. Instead of updating the weights based on the sum of the accumulated errors over all samples :



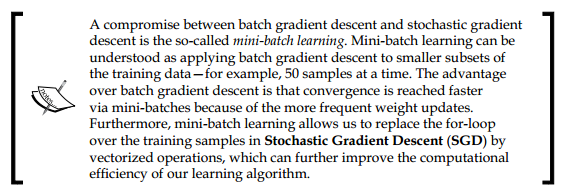
We update the weights incrementally for each training sample:



To obtain accurate results via stochastic gradient descent, it is important to present it with data in a random order, which is why we want to shuffle the training set for every epoch to prevent cycles.



Another advantage of stochastic gradient descent is that we can use if for online learning. In online learning, our model is trained on-the-fly as new training data arrive.



Source code is AdalineSGD at Perceptron and Neural Networks.

If we want to update our model—for example, in an on-line learning scenario with streaming data—we could simply call the partial\_fit method on individual samples—for instance, ada.partial\_fit(X\_std[0, :], y[0]).