Thant Zin Oo (Andy)

705172144

Aaron Zhou Qian

Stochastic Gradient Descent

The prescribed article outlines a basic overview of stochastic gradient descent as a loss minimization algorithm. With the intuition of the loss function as an n-dimensional optimization landscape, the article considers different ways of reaching the global minimum through iterative refinement. Some proposed methods are random search, random local search, and, finally, gradient descent. The article then considers the computational cost of batch gradient descent, which is mainly caused by infrequent weight updates when computing over the entire training set. A solution is proposed, suggesting a reduction of the training batch size along with a random selection of said mini-batch per iteration. Ultimately, stochastic gradient descent is defined as a chosen batch size of 1.

I thought the article was informative, albeit slightly verbose. I found myself backtracking to the slides from previous sections to follow along. Most of my issues weren't from a lack of fundamental understanding, as I have dabbled in machine learning myself, but rather from terminology. I ended up Googling the terms: CIFAR-10, SVM, as well as the differences between numerical and analytic gradients. Aside from these slight technical hiccups, however, I found the article very readable. In particular, the derivation of the convex loss curve via the combination of one and two-dimensional slices was explained very well.

The possible applications of stochastic gradient descent (and machine learning in general) are ever expanding. Image recognition algorithms currently enable autonomous vehicles, deep fake detection software, and even Google Photo's image organization of friends and family. Natural language processing techniques power Google Translate, sentiment analysis engines, and Facebook chat bots. Applications extend into healthcare, enabling diagnostic and predictive algorithms capable of detecting cancers amongst an amalgam of other diseases. There are even full-fledged AI engines for complex strategy games, such as AlphaGo for go, AlphaZero for chess and OpenAI for DotA2. The widespread popularity of stochastic/mini-batch gradient descent techniques to facilitate machine learning is a result of its computational efficiency over batch gradient descent. Instead of updating the weights after deriving the gradient for every training sample, weights are updated based on the gradient of a subset of the training samples. This results in faster iterations, which facilitate faster weight updates, and, hence, faster convergence within a given error bound.