Stochastic Gradient Descent

Dreycey Albin

October 2019

The key need for the stochastic gradient descent (SGD) method was a learning framework that scales well for learning from big data sets. Traditionally, machine learning algorithms are trained on all of the samples in an input dataset, in effort to update each parameter weight, w_i. This is the typical method used for updating of the parameter weights. This uses the average gradient for the loss of each training set (Equation 1). This is a very costly optimization if there are many training data, as the entire dataset needs to be iterated during each epoch, which will cause serious running time problems when many data are used for training [1]. NOTE: equations 1 and 2 specifically focus on the second order gradient descent.

$$w_t = w_{t-1} - \frac{1}{n}\Gamma_{t-1}\sum_{i=0}^{n}\nabla_w Q(z_i, w_{t-1}) \tag{1}$$

This paper extends the weight updates shown in equation 1 to a stochastic method, only updating on one sample per weight update (equation 2). This is a very efficient method for large datasets, as it significantly reduces the time complexity for training. The stochastic nature of the algorithm comes from the fact that the training sample per iteration is chosen at random. Here the gradient is approximated using one sample, $\nabla_w Q(z_i, w_{t-1}$, which results in a noisy approximation. This continues until there is a convergence between iterations $abs(w_t - w_{t-1}) < X$ [1]. Because this is not trained on all of the training examples, this is an approximate approach for determining the parameter weights.

$$w_t = w_{t-1} - \gamma_{t-1}\Gamma_{t-1}\nabla_w Q(z_i, w_{t-1}) \tag{2}$$

There were several subtle limitations of the SGD method mentioned in the paper. One example is the potential convergence problem that arises due to the noisy approximation of the true gradient. If the gains decrease too slowly, then the variance of w_t decreases equally slowly. In contrast, if it decreases too quickly, then it takes a very long time for convergence to occur. Under optimal conditions, the authors note that $\gamma_t \approx t^{-1}$ [1]. It should be noted that this is highly depended on the randomly chosen training data per iteration, so each time the model is trained could be slightly different with respect to convergence.

In particular, something that has me interested is the potential to use a probabilistic method pulling a subset of samples, instead of using only one training sample per iteration. For example, what about using 2 or 3 different training examples per weight update? Could this help reduce the noisy gradient effect? In addition, what would happen if bad data was chosen for the optimization? Could this mean that the model was incorrectly parameterized with respect to the global data? Extending off of these questions, because the algorithm is stochastic, I wonder if it helps to train several times to make sure that the is convergence between output models too [1].

This article uses SGD for bioinformatics applications, focusing on learning parameters for kinetic models using a stochastic gradient descent. This is important because most parameters are not yet known through experimental means (though this is possible). The gradient descent algorithm is used to find the rate constants of the ordinary differential equations describing the stochastic system. A mix of flux balance analysis, using information of metabolic networks, along with an MCMC-like algorithm were combined to predict the parameters for the stochastic model. The gradient of the likelihood function is used to update the direction of the parameters for the updating of the model [2].

References

[1] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[2] Yuanfeng Wang, Scott Christley, Eric Mjolsness, and Xiaohui Xie. Parameter inference for discretely observed stochastic kinetic models using stochastic gradient descent. *BMC systems biology*, 4(1):99, 2010.