The Importance of Regularity Conditions and Other Technicalities in Stochastic Gradient Descent Optimization

Introduction

The utility and applicability of stochastic gradient descent (SGD) and its variants does not necessitate an explanation. Ranging from univariate linear regression to high dimensional deep learning networks, the optimization capabilities of stochastic gradient descent forms the backbone of machine learning that has found an inumerable set of opportunities for use in all scientific and engineering disciplines. Given the vast incorporation of SGD into machine learning and stochastic algorithmic theory, it appears necessary for anyone concerned with the implementation of SGD optimization to be aware of the various mathematical technicalities that leverage the algorithm to correctly predict responses. This paper does exactly that. That is, it analyzes the efficacy of certain regularity conditions, statistical estimators, and real analytical results that promotes a more efficient algorithm.

This paper accomplishes this via two methods. First, it analyzes the convexity and smoothness of the loss function for a simple multivariate linear regression with t-distributed responses. Then, it analyzes how regularity conditions imposed on convexity and smoothness of the loss affect the accuracy of algorithmic parameters (number of epochs, initialization, step size, noise, etc.) in making a sensible prediction. Namely, this paper demonstrates how violations of assumptions regarding convexity, smoothness, and other accuracy guarantees affect performance.

Background

As alluded above, this model analyzes the efficacy of regularity conditions imposed on the loss function that minimizes the expected error of the SGD response. Recall that the framework for SGD optimization can be constructed as follows. Given a set of labeled data points $\mathcal{D} = \{(y_n, \mathbf{z}_n)\}_{n=1}^{N}$, where each y_n is a response to a set (vector) of features $\mathbf{z}_n$ from an experiment, we can construct an algorithm that makes predictions for a new set of features based on the data used to construct it. Assuming there is

some recognizable pattern in the data, we use a loss function $\mathcal{L}(\mathbf{x})$ that penalizes arbitrarily large differences between the training data and the prediction in order to converge to a solution. The goal of the SGD algorithm is to minimize the loss function (the differences between each prediction and its actual result) so that there is a high probability that new data will be correctly predicted according to the pattern described by the model.

Assuming that the data provided in the training set follows a linear pattern, then the loss function can be described as follows,

$$\mathcal{L}(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B}(\beta^T\mathbf{z}_b - y_b)^2 + \lambda||\mathbf{x}||^2$$

where $\mathbf{x} = (\beta_1, ..., \beta_D)$ describes the optimized set of parameters that the algorithm updates and uses to find a "line of best fit" for the responses. Here, $\lambda \geq 0$ controls the strength of the regularization, a parameter that curbs the prediction from over-fitting the data. B is the batch size of the selected data, which is the number of data points selected from $\mathcal{D}$ as opposed to training the entire dataset. Each once the data is fit to the curve, each response can be characterized as

$$y_n = \beta^T\mathbf{z}_n + \epsilon_n$$

where the error term ϵ_n is normally distributed with standard deviation s. In practice, it's best to assume that the errors are t-distributed with scale e^{ψ}, with $\psi = \log s$ and $\nu \geq 0$ degrees of freedom.

Using the above loss function, the SGD algorithm aims to return a set of optimized parameters $\mathbf{x}_*$ that minimizes the loss function (and therefore the total squared differences between each response and its predicted value). Of course, the SGD algorithm is generally. That is, the loss function does not need to be described by linear regression but can be constructed to minimize differences between predictions and realized values. That is, the set of optimized parameters satisfies

$$\mathbf{x}_* = \arg\min \mathcal{L}(\mathbf{x})$$

where $\mathbf{x}_*$ is chosen from an optimized range of vectors given by regularity conditions.

Once the loss function is obtained, an initial set of parameters is initialized, denoted $\mathbf{x}_0$, that SGD must optimize. Using the notion of the gradient, the optimal set of parameters $\mathbf{x}_*$ can be determined by updating the initial set of parameters over a number of time steps. That is, the stochastic gradient descent update is given by

$$\mathbf{x}_{k+1} \to \mathbf{x}_k - \eta_k \nabla \mathcal{L}_k(\mathbf{x}_k)$$

where η_k is the step size of the update. SGD is often noisy since it only uses B samples of dataset of size N, an iterate average of the optimized set of vectors is computed as

$$\bar{\mathbf{x}}_{k/2:k+1} = \frac{1}{\lceil k/2+1 \rceil} \sum_{l=\lfloor k/2 \rfloor}^{k} \mathbf{x}_l$$

which discards the first half of the iterates since we expect the early iterates to be far from the minimize.

In order to accurately converge to a solution $\mathbf{x}_k$, we impose certain regularity conditions that ensures there are no properties of the loss function that could prevent the optimizer from converging. Namely, we assert that a loss function must be μ-convex and L-strongly smooth over its domain of $\mathbf{x}$-vectors. Roughly speaking, convexity ensures that there are no sudden ascents in the loss function that prevent the optimizer from converging towards a minimum. The smoothness of the loss keeps the minimizer converging at sharper rates.

Depending on the chosen step size parameter η, an upper bound can be determined for the expected difference between the optimal set of parameters $\mathbf{x}_*$ and that determined by SGD at iteration k, $\mathbf{x}_k$. For instance, if we choose a constant step size $\eta \in (0, \frac{1}{2L})$, then the expected difference is

$$\mathbb{E}[||\mathbf{x}_k - \mathbf{x}_*|| \leq \alpha^{k/2} ||\mathbf{x}_0 - \mathbf{x}_*||_2 + \frac{2^{1/2}\eta^{1/2}}{\mu^{1/2}}\sigma$$

where $\alpha = 1 - 2\eta\mu(1-\eta L)$ and σ is the standard deviation of our t-distribution that characterizes the variation in our responses.

Similarly, if we use $\bar{\mathbf{x}}_{k/2:k+1}$ (the optimized set of parameters computed by discarding the first half of iterates), we find that the expected difference is also bounded. Assuming a constant step size, define $\delta = M/\mu$, where M is defined as $\mathbb{E}_{k-1}[||\nabla f'(\mathbf{x})||_2] \leq M$. Then the expected error for SGD with constant step size and iterate-averaging at iteration k is

$$[||\bar{\mathbf{x}}_{k/2:k+1} - \mathbf{x}_*||] \leq \frac{\delta\eta\sigma^2}{\mu} + \frac{\sigma}{\mu k^{1/2}}(1 + 2^{3/2}\rho^{1/2}) + \frac{2\delta\sigma^2}{\mu^2 k} + \frac{3\rho^{1/2}\sigma}{\mu\eta k}(2 + \rho^{1/2})$$

$$+\alpha^{k/2}\frac{||\mathbf{x}_0 - \mathbf{x}_*||}{\mu\eta k}(2(1 + \rho^{1/2}) + \frac{\delta}{2\mu}\alpha^{k/2}||\mathbf{x}_0 - \mathbf{x}_*||)$$

where $\rho = L/\mu$.

Finally, we have an error bound for SGD with decreasing step size. Let $\eta_k = \eta/k^a, a \in (1/2, 1)$, $\eta \in (0, \frac{1}{2L})$. Then our iterates satisfy

$$\mathbb{E}[||\mathbf{x}_k - \mathbf{x}_*||^2] \leq \exp\{-\frac{\mu\eta}{2}(k^{1-a} - 1)\}(||\mathbf{x}_0 - \mathbf{x}_*||^2 + \frac{4a\sigma^2\eta^2}{2a - 1}) + \frac{2\sigma^2\eta}{\mu k^a}$$

We discussed earlier how regularity conditions – namely μ-convexity and L-smoothness – ensure a tractable path for SGD to converge. In practice, the idealized regularity conditions may not always be present for the set of all parameter combinations. We will see how certain accommodations will need to be made in order for the SGD regularity conditions to hold.

Methods

Having summarized the basic premises of stochastic gradient descent and its error bounds for variants on the step size and iterate-averaging, we now describe the experiment specifics for the linear regression we wish to optimize. Using the same notation from before, we are given a dataset $\mathcal{D} = \{(y_n, \mathbf{z}_n)\}_{n=1}^N$, where each y_n is a response to a set (vector) of features $\mathbf{z}_n$. Our model for the data is the linear regression $\beta \in \mathbb{R}^D$,

$$y_n = \beta^T\mathbf{z}_n + \epsilon_n$$

, where ϵ_n is t-distributed with scale e^ψ and $\nu > 0$ degrees of freedom. Here $\psi = \log s$, where s is the standard deviation (which describes the variation of

the responses). As $\nu \to \infty$, the t-distribution converges to $\mathcal{N}(0, e^{\psi}) = \mathcal{N}(0, s)$.

We treat ν as fixed, so $\mathbf{x} = (\psi, \beta_1, ..., \beta_D) \in \mathbb{R}^{D+1}$ represents the vector of parameters to be optimized. Given that our responses are t-distributed with scale s, location $\widehat{y}$, and ν degrees of freedom, we write this as $\mathcal{T}(\widehat{y}, s, \nu)$. The density of $\mathcal{T}(\widehat{y}, s, \nu)$ is

$$p(y|\widehat{y}, s, \nu) = \frac{c(\nu)}{s}(1 + \frac{1}{\nu}(\frac{y - \widehat{y}}{s})^2)^{-(\nu+1)/2}$$

where $c(\nu)$ is a normalization term that only depends on ν.

The log loss of the nth observation is

$$l_{(n)}(\mathbf{x}) = \frac{\nu + 1}{2}\log\{1 + \frac{e^{-2\psi}}{\nu}(y - \widehat{y})^2\} + \psi$$

, given $\nu < \infty$. Finally, we define the overall loss function as

$$\mathcal{L}(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N} l_{(n)}(\beta) + \frac{1}{2N}||\mathbf{x}||^2$$

For the remainder of the experiment, we have that the covariates $\mathbf{z}_n$ are t-distributed (multivariately) with mean zero and 10 degrees of freedom ($\nu = 10$). We also take that $D = 10$ and $N = 10000$.

We stated in the previous section that the first regularity condition to guarantee the success of SGD is that the loss function is μ-strongly convex. We can state this as

$$\nabla^2 l_{(n)}(\beta) \geq \mu I$$

for twice differentiable $l_{(n)}$, which implies that we compute the Hessian of each loss for the nth observation and ensure that it is greater than or equal to $\mu > 0$. That is, we must find the range of values of $\widehat{y}$ and ψ that ensure that the loss if μ-convex.

Define $g(\widehat{y}, \psi, y) = l_{(n)}(\mathbf{x})$. Taking the second derivative, we get

$$\frac{\partial^2 g}{\partial \widehat{y}^2} = (\nu + 1)e^{-2\psi}(\nu - e^{-2\psi}(y - \widehat{y})^2) > 0$$

Rearranging, we obtain

$$\nu(\nu+1)e^{-2\psi} > (y-\widehat{y})^2 e^{-2\psi}(\nu+1)e^{-2\psi}$$

implying that the log-loss of the nth observation is $(\nu+1)e^{-2\psi}$-strongly convex with respect to $\widehat{y}$. Rearranging the above and solving for $\widehat{y}$, we get

$$\widehat{y} \in (y - e^{\psi}\sqrt{\nu}, y + e^{\psi}\sqrt{\nu})$$

or

$$\widehat{y} \in (y - s\sqrt{\nu}, y + s\sqrt{\nu})$$

as the range of values for which $\widehat{y}$ is $(\nu+1)e^{-2\psi}$-strongly convex. Given that our loss function will be computed as $l_{(n)}(\mathbf{x}) = g(\mathbf{z}_n^T\beta, \psi, y_n)$, we have

$$\mathbf{z}_n^T\beta \in (y_n - s\sqrt{\nu}, y_n + s\sqrt{\nu})$$

Following the same procedure for the loss function with respect to ψ, we obtain

$$\frac{\partial^2 g}{\partial \psi^2} = 2(\nu+1)\nu(y-\widehat{y})^2 e^{-2\psi} > 0$$

We can once again rearrange the above to ensure that our log loss is once again $(\nu+1)e^{-2\psi}$-strongly convex on the interval

$$(y_n - s\sqrt{\nu}, y_n + s\sqrt{\nu})$$

or

$$\mu = (\nu+1)e^{-2\psi} = 11s^{-2}$$

Because of the $e^{-2\psi}$ term, we see that the $g(\widehat{y}, \psi, y)$ is convex for all $\psi \in \mathbb{R}$ since the term never dips below zero. Nonetheless, it may not be optimal for ψ to range over all values of $\mathbb{R}$ and so it would be best if the domain were restricted.

Having resolved the first regularity condition, we now show that $g(\widehat{y}, \psi, y)$ is L-strongly smooth on the domain

$$(y_n - s\sqrt{\nu}, y_n + s\sqrt{\nu})$$

Recall that a function ϕ is L-strongly smooth for any $\mathbf{x}$,$\mathbf{y} \in \mathcal{A}$ if

$$||\phi'(\mathbf{x}) - \phi'(\mathbf{y})|| \leq L||\mathbf{x} - \mathbf{y}||$$

which implies that

$$L := \sup ||\phi''(\mathbf{x})||$$

Hence we have that the smoothness coefficient L for the loss with respect to $\widehat{y}$ is

$$L := \sup ||\frac{\partial^2 g}{\partial \widehat{y}^2}|| = \sup |(\nu+1)e^{-2\psi}(\nu - e^{-2\psi}(y-\widehat{y})^2)||$$

$$= \sup |(\nu+1)s^{-2}(\nu - s^{-2}(y-\widehat{y})^2)|$$

We can readily see that this function encapsulated in the supremum is concave up with a maximum at the location parameter $\widehat{y}$. Therefore, we conclude that L is finite and that

$$L = 110e^{-2\psi} = 110s^{-2}$$

Determining the smoothness coefficient L_ψ for the loss with respect to ψ is more tricky. By the same procedure,

$$L_\psi := \sup ||2\nu(\nu+1)(y-\widehat{y})^2 e^{-2\psi}|| = \sup 220e^{-2\psi}(y-\widehat{y})^2$$

As $\psi \to -\infty, e^{-2\psi} \to \infty$ which implies that our smoothness coefficient L_ψ is not bounded from above. In other words, as the standard deviation approaches $-\infty$, L_ψ approaches infinity and we have a greater bound for SGD's rate of convergence towards an optimal solution.

To avoid initializing values for ψ that are arbitrarily far from the solution, we can equate the smoothness coefficient L obtained with respect to $\widehat{y}$ to the objective inside the supremum and solve for the range of values ψ that are less than the smoothness factor. That is, all ψ that satisfy

$$L = 110s^{-2} > 220(y-\widehat{y})^2 e^{-2\psi}$$

Doing so yields that ψ is L-strongly smooth if

$$-\frac{1}{2}\ln 2 < \ln(y-\widehat{y})$$

Experiments

After considering the regularity conditions and technicalities that optimize SGD, we can simulate the effectiveness of our linear regression model using various hyperparameters. Namely, we consider three linear regression models: standard SGD with constant step size η, SGD with constant step size η and iterate averaging, and SGD with decreasing step size, where $\eta_k = \eta/k^a$, η fixed. For all experiments, we use a batch size $B = 10$ from a dataset of size $N = 10000$, regression dimension of $D = 10$, and $\nu = 5$ degrees of freedom for the probability distribution $\mathcal{T}(\mathbf{z}_n^T\beta, \psi, y)$ describing the variation in our responses from the prediction.

Recall that the set of our parameters can written as

$$\mathbf{x} = (\psi, \beta_1, ...\beta_{10}) \in \mathbb{R}^{D+1}$$

. We initialize our initial parameters as $\mathbf{x}_0 = \mathbf{0}$ where the distribution of our responses $\mathcal{T}(\mathbf{z}_n^T\beta, \psi, y)$ has an initial standard deviation $s = 1$, implying that $\psi = 0$. Finally, we set our initial step size as $\eta_0 = 5$ and set $\eta = 0.2$. For the decreasing step size variant, we set $a = 0.51$.

The true set of optimal regression parameters, or true-β, is $\beta_* = (1, ..., D) = (1, ..., 10)$. Considering the first model, standard SGD with constant step size $\eta = 0.2$, we obtain results as described by **Figure 1** and **Figure 2**. Clearly, we see that SGD converges very quickly to the optimal solution $\mathbf{x}_*$ in just 2 epochs. We find that the convergence to $_*$ is largely unrelated to the initial set of parameters x_0 as SGD with constant step size manages to converge rather quickly regardless in just about the same time. This is mostly likely due to our large initial step size of $\eta_0 = 5$

Considering SGD with constant step size and 50% iterate averaging, we find that our algorithm converges approximately twice as slowly but results in greater precision (less gradient noise). Observing **Figure 3** and **Figure 4**, we find that although the algorithm is slower due, we see introducing iterate-averaging significantly reduces gradient noise since the algorithm discards the initial, error-prone parameter updates. Once again, initialization has little effect for arbitrarily far (but not too far) initialized parameters from the optimal. Once again, this is mostly likely due to the large initial step size η_0. We would find that if we keep extending the distance between

$\mathbf{x}_0$ and $\mathbf{x}_k$ towards infinity, then the magnitude of η_0 has little effect in promoting performance.

Our final variant, SGD with decreasing step size $\eta_k = \eta/k^a$, converges towards the optimal $\mathbf{x}_*$ at the slowest rate. Observing **Figure 5** and **Figure 6**, we see that it takes approximately 300 epochs for SGD to converge towards $\mathbf{x}_*$. Note, however, that the limit of the error between $\mathbf{x}_k$ and $\mathbf{x}_*$ is the least and most stable of the three algorithms. Therefore, it is clear that a decreasing step size is desired for any application that requires the most accurate convergence.

One final metric that we aim to analyze is the t-distribution of our responses. If we set $\eta = 15$, we find that the error between $\mathbf{x}_k$ and $\mathbf{x}_*$ flattens as compared to the distribution with 5 degrees of freedom. Generally, we see that letting $\eta \to \infty$ towards the normal distribution $\mathcal{N}(0, s)$, we find that our error flattens rather being minimized at approximately 4 epochs and then slowly increasing.

Conclusion

This paper analyzed the efficacy of regularity conditions pertaining mainly to convexity and smoothness of the loss function for three variants on stochastic gradient descent: constant step size, constant step size with iterate-averaging, and decreasing step size. We find that there is an obvious tradeoff between speed and accuracy. That is, SGD with constant step size converged to its optimal $\mathbf{x}_*$ at the fastest rate but with the greatest noise. Introducing 50% iterate-averaging into this model decreased the gradient noise between responses and and predictions but it converged approximately twice as slowly. Finally, we saw that SGD with decreasing step size provided the best performance with the least noise at the expense of substantial training times. In order to augment the accuracy of our SGD variants, we saw that increasing the batch size, initializing the initial set of parameters closer to the optimum $\mathbf{x}_*4$, and decreasing the step size were vital tools to constructing a more accurate performance.

Despite the efficacy of our analysis in determining suitable ranges for the hyperparameters for this simple multivariate linear regression model, we urge

that the framework is general and can be applied towards almost all types of machine learning applications. For instance, we could have developed a more intricate model by fitting a logistic regression instead of a linear one. Furthermore, we could have tried different optimizers instead of SGD, such as ADAM, AdaMax, RMSProp, etc. Perhaps those variants could be the subject of future research to showcase just how extensive is the literature of machine learning and stochastic optimization.

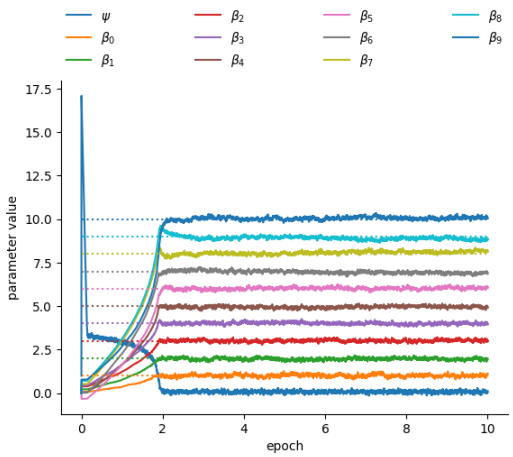

Figure 1: Convergence of parameters $\mathbf{x}_k$ to β_* and ψ_* for SGD with constant step size η

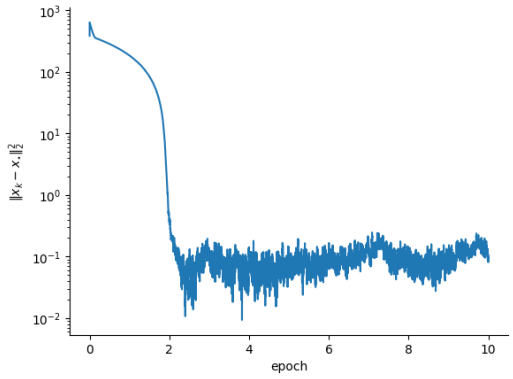

Figure 2: Error analysis of SGD with constant step size η over the number of epochs

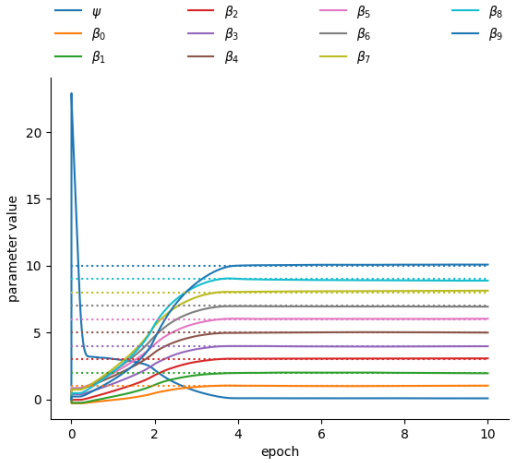

Figure 3: Convergence of parameters $\mathbf{x}_k$ to β_* and ψ_* for SGD with constant step size η and 50% iterate averaging

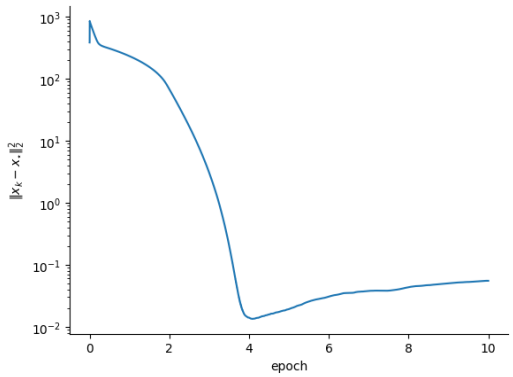

Figure 4: Error analysis of SGD with constant step size η and 50% iterate averaging over the number of epochs

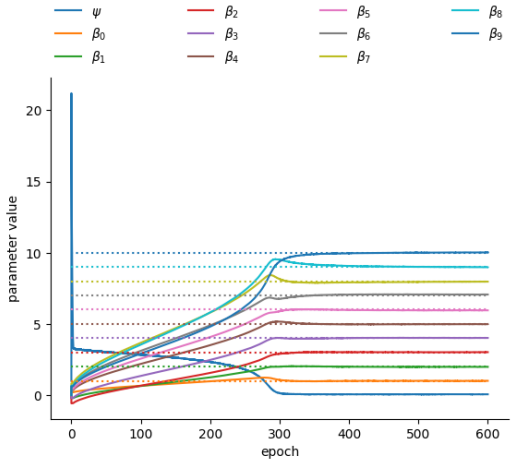

Figure 5: Convergence of parameters $\mathbf{x}_k$ to β_* and ψ_* for SGD with decreasing size η_k

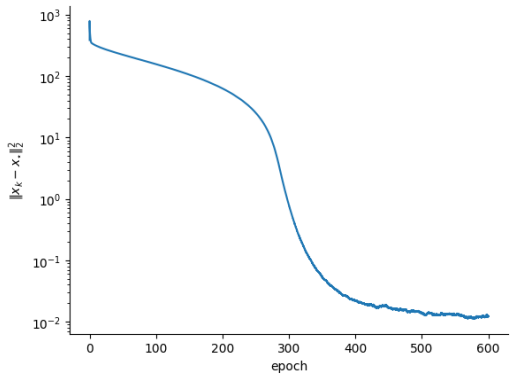

Figure 6: Error analysis of SGD with decreasing step size η_k over the number of epochs