ECE 901: Large-scale Machine Learning and Optimization **Spring 2018**

Lecture 5 — 02/08

Lecturer: Dimitris Papailiopoulos *Scribe: Kamini Jodha, Ehsan(Eric) Qasemi*

Note: These lecture notes are still rough, and have only have been mildly proofread.

This is the danger environment.

5.1 Review

In gradient descent (GD), we start with model and calculate the full gradient on the complete training set

$$W_{k+1} = W_k - \gamma \nabla f(W_k) \tag{5.1}$$

Using the property of finite sums ($f(x) = \sum_{i=0}^{N} f_i(x)$) in the gradient we can rewrite the above equation as 5.2

$$\nabla f(W_k) = \frac{1}{n} \sum_i \nabla f_i(W_k)$$
$$W_{k+1} = W_k - \gamma * \frac{1}{n} \sum_i \nabla f_i(W_k) \tag{5.2}$$

The equation 5.3 is base of what we call SGD. In SGD 5.3 we start with model and randomly sample dataset to compute gradient on it ($\nabla f_{Sk}(W_k)$). Finally we update the model parameters based on this partially calculated gradient. SGD have been widely used in different contexts e.g. back propagation algorithm, perceptron algorithm, and linear mean square filter (LMS algorithm).

Note: In this course we consider random sampling with replacement as it is easier to realise (proof), however this method of sampling is rarely used in practice.

$$S_n \sim Unif(1, \cdots, n)$$
$$W_{k+1} = \arg\min_{W} \{f_{Sk}(W_k) - < f_{Sk}(W_k), W_k - W > + \frac{1}{2\gamma} ||W_k - W||_2^2\} \tag{5.3}$$

Remark: In the SGD algorithm, the cost of computing gradient of one data point or sample $\nabla f_{Sk}(W_k) = \nabla f(W_k, S_k)$. Consequently, compared to GD algorithm, SGD has smaller memory/CPU footprint because we calculate just one gradient, is easier to implement (auto grad), and has a algorithmic paradigm that ease the Software development around it.

5.2 Convergence of SGD

Motivating questions:
Q1) Does SGD converge to minimum value?
Q2) How fast it converges?
Q3) When is it faster than GD?

Proof: In this section we are going to calculate the bounds for SGD to give acceptable result (expected value of distance from optimum point be less than ϵ).

Assumptions:

- f is λ-strongly convex.
- expected value of the gradient on sampled set is bounded by M^2 (equation 5.5).

Note: Based on two assumptions it is easy to show that, f is also a Lipschitz function.

Note: In expectation (due to the uniform sampling) we know the gradient in SGD and GD are equal (equation5.4).

$$\mathbb{E}_{Sk}\{\nabla f_{Sk}(W_k)\} = \nabla f(W_k) \tag{5.4}$$

$$\mathbb{E}_{Sk}\{\nabla f_{Sk}(W_k)\} \leq M^2 \tag{5.5}$$

$$\begin{aligned}||W_{k+1} - W^*||_2^2 =& ||W_k - W^*||_2^2 - 2\gamma < f_{Sk}(W_k), W_k - W^* > +\gamma^2||\nabla f_{Sk}(W_k)||\\ \mathbb{E}||W_{k+1} - W^*||_2^2 =& \mathbb{E}||W_k - W^*||_2^2 - 2\gamma\mathbb{E} < f_{Sk}(W_k), W_k - W^* > +\gamma^2\mathbb{E}||\nabla f_{Sk}(W_k)||\end{aligned}$$

In order to simplify the equation, lets substitute $\Delta_k = \mathbb{E}||W_k - W^*||_2^2$. Now, if we apply 5.5 to the equation and push the expectation inside the dot product ($\mathbb{E}_{S_1,\cdots,S_k} = \mathbb{E}_{S_k}(\mathbb{E}_{S_0,\cdots,S_{k-1}})$), we have:

$$\begin{aligned}\mathbb{E} < f_{Sk}(W_k), W_k - W^* > &= \mathbb{E}_{S_1,\cdots,S_{k-1}} < \mathbb{E}_{S_k}\nabla f_{Sk}(W_k), W_k - W^* >\\ \mathbb{E} < f_{Sk}(W_k), W_k - W^* > &= \mathbb{E} < \nabla f(W_k), W_k - W^* >\\ \text{So we have: } \Delta_{k+1} &\leq \Delta_k - 2\gamma\mathbb{E} < f_{Sk}(W_k), W_k - W^* > +\gamma^2 M^2\\ \Delta_{k+1} &\leq \Delta_k - 2\gamma\mathbb{E} < \nabla f(W_k), W_k - W^* > +\gamma^2 M^2 \qquad (5.6)\end{aligned}$$

To further bound 5.6, we are going to use strong convexity property. by definition of strong convexity we have:

$$\begin{aligned}&f(W^*) \geq f(W) + < \nabla f(W), W^* - W > + \frac{\lambda}{2}||W_k - W||_2^2\\ &\Rightarrow < \nabla f(W), W^* - W > \geq \frac{\lambda}{2}||W_k - W||_2^2,\ \forall W\\ &\Rightarrow \nabla f(W), \text{correlate strongly with the direction that we have to take}\\ &\Rightarrow \mathbb{E} < f_{Sk}(W_k), W_k - W^* > \geq \frac{\lambda}{2}\mathbb{E}||W_k - W||_2^2 \qquad (5.7)\end{aligned}$$

By applying 5.7 to 5.6 we have:

$$\begin{aligned}
\Delta_{k+1} &\leq \Delta_k - \gamma\lambda\mathbb{E}||W_k - W||_2^2 + \gamma^2 M^2 \\
\Delta_{k+1} &\leq \Delta_k(1-\gamma\lambda) + \gamma^2 M^2 \\
\Rightarrow \Delta_{k+1} &\leq (1-\gamma\lambda)^{k+1}||W_0 - W^*||^2 + \sum_{i=0}^{k}(1-\gamma\lambda)\gamma^2 M^2 \\
\Rightarrow \Delta_{k+1} &\leq (1-\gamma\lambda)^{k+1}||W_0 - W^*||^2 + \frac{\gamma M^2}{\lambda} \\
\Rightarrow \mathbb{E}||W_T - W^*||_2^2 &\leq (1-\gamma\lambda)^{T}||W_0 - W^*||^2 + \frac{\gamma M^2}{\lambda} \qquad (5.8)
\end{aligned}$$

lets make both part on the right hand side of 5.8 smaller than $\epsilon/2$. So we have:

$$\frac{\gamma M^2}{\lambda} \leq \epsilon/2 \Rightarrow \gamma \leq \frac{\epsilon\lambda}{2M^2} \qquad (5.9)$$

$$\text{we also have:} \qquad (1-\gamma\lambda)^T||W_0 - W^*||^2 = \frac{\epsilon}{2}$$

$$\xrightarrow[||W_0 - W^*||^2 = \Delta_0 = R]{\text{taking log}} \qquad T\log(1-\gamma\lambda) + 2\log R = \log\frac{\epsilon}{2}$$

$$\Rightarrow T = \frac{\log\frac{\epsilon}{2} - 2\log R}{\log(1-\gamma\lambda)}$$

$$\xrightarrow{5.9} T \geq \frac{4M^2}{\lambda^2}\frac{1}{\epsilon}log(\frac{R}{\epsilon}) \qquad (5.10)$$

□

So we showed that, if f is λ-strongly convex and expected value of its gradient is bounded (equation 5.5), then for λ satisfying 5.9, after T steps where T satisfies 5.10, we are expected to be at most ϵ distance away from the optimum point or $\mathbb{E}||W_T - W^*||_2^2 \leq \epsilon$.

5.3 Comparison with GD

On comparing the convergence rate of SGD with GD, we find that GD converges far more faster than SGD.

Let's consider a function which is λ-strongly convex and is upper bounded by M^2, we can say that for ϵ error the iteration complexity for Stochastic Gradient Descent can be given by:

$$T_\epsilon^{SGD} = O(\frac{M^2}{\lambda^2}\frac{1}{\epsilon} * log(\frac{R}{\epsilon})) \qquad (5.11)$$

For a λ-strongly convex and β-smooth function the iteration complexity for Gradient Descent can be given by:

$$T_\epsilon^{GD} = O(\frac{\beta}{\lambda} * log(\frac{R}{\epsilon})) \qquad (5.12)$$

The above complexities of number of iterations for SGD and GD are never going to be equivalent for there will never exist an ϵ which is equal to 1.

For example, let's consider the logistic regression function given by:

$$f(W) = \frac{1}{n}\sum_{i=0}^{n} log(1 + e^{-y_i<W;X_i>}) + \frac{\lambda}{2}||w||_2^2 \tag{5.13}$$

By adding the L_2 norm term in the above mentioned function for logistic regression, it becomes λ-strongly convex.

Assumptions:

- data points are not too large which means that while considering the norm of the data point we can think of each element to be a constant and assuming the norm is $\sqrt{d}$ dimension, i.e.,$||X_i|| = O(\sqrt{d})$.
- only considering the models that are not too big, we can write that $\forall w, ||w|| \le O(\sqrt{d})$. The reason for these assumptions is that L_2 norm is not smooth itself unless the argument is bounded.
- the distance of the first iterate from the optimum is given as: $||w_0 - w^*|| \le O(\sqrt{d})$.
- and the strong convex parameter is given as: $\lambda = O(1)$.

Then, $f(w)$ will be:

- not too much but of the order $O(1)$ - strongly convex, which is a constant.
- $O(\sqrt{d})$ - Lipschitz
- $O(d)$ - Smoothness
- the bound of the gradient will be: $M^2 = O(d)$.

It should be kept in mind that the results above are not constants, the M and β parameter penalize the problem size, sometimes d. In order to understand the behavior of this algorithm, one must have an estimate of how these results scale the problems.

Now, substituting the results above in 5.11 and 5.12, the complexity is given by:

$$T_\epsilon^{SGD} = O(\frac{d}{\epsilon}\ log(\frac{d}{\epsilon})) \tag{5.14}$$

$$T_\epsilon^{GD} = O(d\ log(\frac{d}{\epsilon})) \tag{5.15}$$

From the above 5.14 and 5.15, we can say that the SGD is $\frac{1}{\epsilon}$ times slower than GD but it is also n times faster because of the cost of each iteration which is proved below.

To compute the gradient of one loss, we write:

$$\nabla \ell_i(w) = \nabla \ell(< X_i, W >) = \ell'(< X_i, W >)X_i \tag{5.16}$$

where, ℓ' is the point-wise derivative function evaluated on data points.
The cost of iteration of a GD is proprtional to the non-zero elements of X_i written as $nnz(X_i)$. The cost of one iteration SGD is given as $\frac{nnz(X_i)}{n}$. Therefore, we can say *time* can be given by multiplying the number of iterations required to reach ϵ with cost per iteration. In order to compare the *time* taken by SGD and GD, we can write:

$$\frac{time(GD, \epsilon)}{time(SGD, \epsilon)} = O(\frac{nnz(A)\ d\ log(\frac{d}{\epsilon})}{\frac{nnz(A)}{n}\ \frac{d}{\epsilon}\ log(\frac{d}{\epsilon})}) \tag{5.17}$$

$$= O(\frac{n}{\epsilon}\ log(\frac{1}{\epsilon})) \tag{5.18}$$

$$\tag{5.19}$$

For $\epsilon \gg \frac{1}{n}$; we can write:

$$\frac{time(GD, \epsilon)}{time(SGD, \epsilon)} = O(n\epsilon) \tag{5.20}$$

Thus, we conclude that SGD algorithm is n times faster than GD.
Remarks:

- Because the ERM itself does not concentrates faster than $\frac{1}{\sqrt{n}}$(i.e., the error of Empirical risk with n samples), and there will always be a difference of $\frac{1}{\sqrt{n}}$, this is the best possible result one can achieve.
- The above bounds are all in expectations, is there a way to improve them? One simple idea is to use $Markov's\ Inequality$

$$Pr(|X| \geq a) \leq \frac{E[X]}{a} \tag{5.21}$$

- GD algorithm is trivially parallelizable but SGD algorithm is inherently serial in nature. This is because every single model that is sampled and calculated depends on the past.