Lecture 5: Minibatching and Decreasing Step Sizes

CS4787 — Principles of Large-Scale Machine Learning Systems

Where we left off: we looked at how stochastic gradient descent performs on both convex and objectives. For non-convex objectives, we assumed that our function was $L$-Lipschitz continuous, i.e. for any objective component $i$, and points $x$, and $y$, 
$$
\| \nabla f_i(x) - \nabla f_i(y) \| \leq L \cdot \| x - y \| .
$$
We also assumed that the step size $\alpha$ is bounded such that $0 < \alpha < 1 / L$,

and that the mean-squared-error of the gradient samples is, for any $w \in \mathbb{R}^d$, bounded by
$$
\frac{1}{n} \sum_{i=1}^{n} \left\| \nabla f_i(w) - \nabla f_j(w) \right\|^2 = \mathbb{E}_i \left[ \| \nabla f_i(w) - \nabla f_j(w) \|^2 \right] \leq \sigma^2.
$$

For convex objective functions $f$, we additionally assumed $\mu$-strong convexity, i.e.
$$
f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{\mu}{2} \cdot \|x - y\|^2.
$$

Under these conditions, we got for the non-convex case that if $w_t$ is the $t$th iterate of SGD with constant step size $\alpha$, after running for $T$ timesteps
$$
\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \left[ \| \nabla f(w_t) \|^2 \right] \leq \frac{2 (f(w_0) - f^*)}{\alpha T} + \frac{\alpha \sigma^2 L}{2}.
$$

For the strongly convex case, we got that
$$
\mathbb{E} \left[ f(w_T) - f^* \right] \leq \exp(-\mu \alpha T) \cdot (f(w_0) - f^*) + \frac{\alpha \sigma^2 L}{2 \mu}.
$$

Notice that even if we run for a large number of iterations, this is not going to necessarily go to zero!

Previously, with gradient descent, if we wanted to get a solution of a desired level of accuracy (either small gradient or small objective gap) we could just keep running until we observed a gradient small enough to satisfy our desires. Now though, this won’t necessarily happen.

One way to achieve a desired level of error is to choose an $\alpha$ and $T$ as a function of the error level. For example, for non-convex SGD, if for some $\epsilon > 0$ we want to guarantee that we will get
$$
\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \left[ \| \nabla f(w_t) \|^2 \right] \leq \epsilon,
$$
it suffices to pick $\alpha$ and $T$ such that
$$
\frac{2 (f(w_0) - f^*)}{\alpha T} = \frac{\alpha \sigma^2 L}{2} = \epsilon.
$$
This happens when
$$
\alpha = \frac{\epsilon}{\sigma^2 L} \quad \text{and} \quad T = \frac{4 \sigma^2 L (f(w_0) - f^*)}{\epsilon^2}.
$$

1
This can be compared with our results from gradient descent (Lecture 2) where we could get the same guarantee with
\[\alpha = \frac{1}{L} \quad \text{and} \quad T \leq \frac{2L(f(w_0) - f^*)}{\epsilon} .\]
Similarly, for strongly convex SGD, if we want to guarantee that
\[\mathbb{E} [f(w_T) - f^*] \leq \epsilon,\]
it suffices to pick \(\alpha\) and \(T\) such that
\[\exp(-\mu\alpha T) \cdot (f(w_0) - f^*) = \frac{\alpha \sigma^2 L}{2\mu} = \frac{\epsilon}{2}.\]
This happens when (letting \(\kappa = L/\mu\) as usual)
\[\alpha = \frac{\epsilon}{\sigma^2 \kappa} \quad \text{and} \quad T = \frac{\sigma^2 \kappa}{\epsilon} \log \left( \frac{2 (f(w_0) - f^*)}{\epsilon} \right) .\]
In comparison, gradient descent (Lecture 2) had
\[T \geq \kappa \cdot \log \left( \frac{f(w_0) - f^*}{\epsilon} \right) .\]

What can we conclude from this? Here's one thing that we can get: the asymptotic runtime used by these algorithms. For each of non-convex GD/SGD and strongly convex GD/SGD, write a big-\(O\) expression for the total amount of compute that would be done by the algorithm to achieve error \(\epsilon\). Give your result in terms of \(\epsilon, \kappa\) (for strongly-convex), \(n\), and \(\sigma^2\), treating all other expressions (such as \(f(w_0) - f^*\)) as constant.

When might one algorithm be better than the other?

**Minibatching.** One way to make all these rates smaller is by decreasing the value of \(\sigma^2\). A simple way to do this is by using **minibatching.** With minibatching, we use a sample of the gradient examples of size larger than 1. That is, our update rule looks like
\[w_{t+1} = w_t - \alpha_t \sum_{b=1}^{B} \nabla f_{t,b}(w_t) .\]
If the batch size is \(B\), this results in an estimator with variance \(B\) times smaller.

**How does this trade off work for faster convergence?**

**Diminishing Step Size Rules.** We will see how we can get an “optimal” step size from the analysis of convex SGD, starting with the expression (from the Lecture 4 notes)
\[\mathbb{E} [f(w_{t+1}) - f^*] \leq (1 - \mu \alpha) \mathbb{E} [f(w_t) - f^*] + \frac{\alpha^2 \sigma^2 L}{2}.\]