Lecture 5: Stochastic Gradient Descent

CS4787 — Principles of Large-Scale Machine Learning Systems

Combining two principles we already discussed into one algorithm.

- Principle: Write your learning task as an optimization problem and solve it with a scalable optimization algorithm.
- Principle: Use subsampling to estimate a sum with something easier to compute.

Recall: we *parameterized* the hypotheses we wanted to evaluate with parameters $w \in \mathbb{R}^d$, and want to solve the problem

$$\text{minimize: } R(h_w) = \frac{1}{n}\sum_{i=1}^{n} L(h_w(x_i), y_i) = f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w) \text{ over } w \in \mathbb{R}^d.$$

Stochastic gradient descent (SGD). Basic idea: **in gradient descent, just replace the full gradient (which is a sum) with a single gradient example**. Initialize the parameters at some value $w_0 \in \mathbb{R}^d$, and decrease the value of the empirical risk iteratively by sampling a random index $\tilde{i}_t$ uniformly from $\{1, \ldots, n\}$ and then updating

$$w_{t+1} = w_t - \alpha_t \cdot \nabla f_{\tilde{i}_t}(w_t)$$

where as usual w_t is the value of the parameter vector at time t, α_t is the *learning rate* or *step size*, and ∇f_i denotes the gradient of the loss function of the ith training example. Compared with gradient descent and Newton's method, SGD is simple to implement and runs each iteration faster.

A potential objection: **this is not necessarily going to be decreasing the loss at every step!** So we can't demonstrate convergence by using a proof like the one we used for gradient descent, where we showed that the loss decreases at every iteration of the algorithm. The fact that SGD doesn't always improve the loss at each iteration motivates the question: **does SGD even work? And if so, why does SGD work?**

Demo. Gradient descent versus stochastic gradient descent on linear regression.

Why might it be fine to get an approximate solution to an optimization problem for training?

Takeaway:

Why does SGD work? Unlike GD, SGD does not necessarily decrease the value of the loss at each step. Let's just try to analyze it in the same way that we did with gradient descent and see what happens. But first, we need some new assumption that characterizes *how far* the gradient samples can be from the true gradient. Assume that, for some constant $G > 0$, the magnitude of our gradient samples are bounded, for all $w \in \mathbb{R}^d$, by

$$\|\nabla f_i(x)\| \leq G.$$

In other words, this is a global bound on the magnitude of the gradient samples, and is similar in spirit to the global bound on the magnitude used in Hoeffding's inequality. As before, we will also assume that for some constant $L > 0$, for all x in the space and for any vector $u \in \mathbb{R}^d$,

$$\left|u^T \nabla^2 f(x) u\right| \leq L \|u\|^2.$$

From here, we can analyze SGD like we did with gradient descent, first *without assuming convexity*. From Taylor's theorem, there exists a ξ_t such that

$$\begin{aligned}
f(w_{t+1}) &= f(w_t - \alpha_t \nabla f_{\tilde{i}_t}(w_t)) \\
&= f(w_t) - \left(\alpha_t \nabla f_{\tilde{i}_t}(w_t)\right)^T \nabla f(w_t) + \frac{1}{2}\left(\alpha_t \nabla f_{\tilde{i}_t}(w_t)\right)^T \nabla^2 f(\xi_t)\left(\alpha_t \nabla f_{\tilde{i}_t}(w_t)\right) \\
&\leq f(w_t) - \alpha_t \nabla f_{\tilde{i}_t}(w_t)^T \nabla f(w_t) + \frac{\alpha_t^2 L}{2}\left\|\nabla f_{\tilde{i}_t}(w_t)\right\|^2 \\
&\leq f(w_t) - \alpha_t \nabla f_{\tilde{i}_t}(w_t)^T \nabla f(w_t) + \frac{\alpha_t^2 G^2 L}{2}.
\end{aligned}$$

Now we're faced with a problem. The term

$$-\alpha_t \nabla f_{\tilde{i}_t}(w_t)^T \nabla f(w_t)$$

is not necessarily nonnegative, so we're not necessarily making any progress in the loss. The key insight: we are making progress **in expectation**. If we take the expected value of both sides of this expression (where the expectation is taken over the randomness in the sample selection $\tilde{i}_t$), we get

$$\begin{aligned}
\mathbf{E}\left[f(w_{t+1})\right] &\leq \mathbf{E}\left[f(w_t) - \alpha_t \nabla f_{\tilde{i}_t}(w_t)^T \nabla f(w_t) + \frac{\alpha_t^2 G^2 L}{2}\right] \\
&= \mathbf{E}\left[f(w_t)\right] - \alpha_t \mathbf{E}\left[\nabla f_{\tilde{i}_t}(w_t)^T \nabla f(w_t)\right] + \frac{\alpha_t^2 G^2 L}{2}.
\end{aligned}$$

Now, the expected value of $\nabla f_{\tilde{i}_t}(w_t)$ given w_t is

$$\mathbf{E}\left[\nabla f_{\tilde{i}_t}(w_t)\middle| w_t\right] = \sum_{i=1}^n \nabla f_i(w_t) \cdot \mathbf{P}\left(\tilde{i}_t = i\middle| w_t\right) = \sum_{i=1}^n \nabla f_i(w_t) \cdot \frac{1}{n} = \nabla f(w_t),$$

so

$$\mathbf{E}\left[f(w_{t+1})\right] \leq \mathbf{E}\left[f(w_t)\right] - \alpha_t \mathbf{E}\left[\|\nabla f(w_t)\|^2\right] + \frac{\alpha_t^2 G^2 L}{2}.$$

Rearranging the terms, summing up over T iterations, and telescoping the sum,

$$\begin{aligned}
\sum_{t=0}^{T-1} \alpha_t \mathbf{E}\left[\|\nabla f(w_t)\|^2\right] &\leq \sum_{t=0}^{T-1} \left(\mathbf{E}\left[f(w_t)\right] - \mathbf{E}\left[f(w_{t+1})\right]\right) + \sum_{t=0}^{T-1} \frac{\alpha_t^2 G^2 L}{2} \\
&= f(w_0) - f(w_T) + \frac{G^2 L}{2} \sum_{t=0}^{T-1} \alpha_t^2 \\
&\leq f(w_0) - f^* + \frac{G^2 L}{2} \sum_{t=0}^{T-1} \alpha_t^2
\end{aligned}$$

where f^* is the global optimum of f. This is a pretty nice expression, but we still need to do something useful with the term we bounded on the left. Here's one thing we can do: run SGD for a random number of iterations τ, where we run for $\tau = t$ iterations with probability

$$\mathbf{P}\left(\tau = t\right) \propto \frac{\alpha_t}{\sum_{k=0}^{T-1} \alpha_k}.$$

Then the expected squared-norm of the gradient of $\nabla f(w_\tau)$ is

$$\mathbf{E}\left[\left\|\nabla f(w_\tau)\right\|^2\right] = \sum_{t=0}^{T-1} \mathbf{E}\left[\left\|\nabla f(w_t)\right\|^2\right] \cdot \mathbf{P}\left(\tau = t\right) = \left(\sum_{k=0}^{T-1} \alpha_k\right)^{-1} \sum_{t=0}^{T-1} \alpha_t \mathbf{E}\left[\left\|\nabla f(w_t)\right\|^2\right],$$

and so we can bound this with

$$\mathbf{E}\left[\left\|\nabla f(w_\tau)\right\|^2\right] \leq \left(\sum_{k=0}^{T-1} \alpha_k\right)^{-1} \left(f(w_0) - f^* + \frac{G^2 L}{2} \sum_{t=0}^{T-1} \alpha_t^2\right).$$

For example, if the learning rate is a constant $\alpha_t = \alpha$, then

$$\mathbf{E}\left[\left\|\nabla f(w_\tau)\right\|^2\right] \leq (T\alpha)^{-1} \left(f(w_0) - f^* + \frac{G^2 L}{2} T\alpha^2\right) = \frac{f(w_0) - f^*}{\alpha T} + \frac{\alpha L G^2}{2}.$$

This second term is our *noise ball* term: the term that is in some sense "causing" SGD to converge not to a point with zero gradient but rather to some reason nearby. We can avoid this by **decreasing the learning rate over time**. The norm of the gradient of the output w_τ will be guaranteed to go to zero if

$$\sum_{t=0}^{T-1} \alpha_t \text{ grows much faster than } \sum_{t=0}^{T-1} \alpha_t^2.$$

One example of such a step size rule is $\alpha_t = (t+1)^{-1/2}$. Then we have (these are approximations, but can be made rigorous with a bit of work)

$$\sum_{t=0}^{T-1} \alpha_t = \sum_{t=0}^{T-1} \frac{1}{\sqrt{t+1}} \approx \int_0^T \frac{1}{\sqrt{x}}\, dx = 2\sqrt{T}$$

and

$$\sum_{t=0}^{T-1} \alpha_t^2 = \sum_{t=0}^{T-1} \frac{1}{t+1} \approx \int_1^{T+1} \frac{1}{x}\, dx = \log(T+1).$$

As a result, with this $\alpha_t = (t+1)^{-1/2}$ decreasing learning rate, we get

$$\begin{aligned}\mathbf{E}\left[\left\|\nabla f(w_\tau)\right\|^2\right] &\lessapprox \left(2\sqrt{T}\right)^{-1} \left(f(w_0) - f^* + \frac{G^2 L}{2}\left(\log(T+1)\right)\right) \\ &= \frac{2(f(w_0) - f^*) + G^2 L \log(T+1)}{4\sqrt{T}} = \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{T}}\right).\end{aligned}$$

This is, indeed, going to zero as the number of steps T increases.

How does this compare to the expression that we got for gradient descent?

Gradient descent for strongly convex objectives. This was without assuming strong convexity. But how does SGD perform on strongly convex problems? As before, we start from this sort of expression

$$\mathbf{E}\left[f(w_{t+1})\right] \leq \mathbf{E}\left[f(w_t)\right] - \alpha_t \mathbf{E}\left[\left\|\nabla f(w_t)\right\|^2\right] + \frac{\alpha_t^2 G^2 L}{2}$$

and apply the Polyak–Lojasiewicz condition,

$$\left\|\nabla f(x)\right\|^2 \geq 2\mu\left(f(x) - f^*\right);$$

this gives us

$$\mathbf{E}\left[f(w_{t+1})\right] \leq \mathbf{E}\left[f(w_t)\right] - 2\mu\alpha_t \mathbf{E}\left[f(w_t) - f^*\right] + \frac{\alpha_t^2 G^2 L}{2}.$$

Subtracting f^* from both sides, we get

$$\mathbf{E}\left[f(w_{t+1}) - f^*\right] \leq (1 - 2\mu\alpha_t)\mathbf{E}\left[f(w_t) - f^*\right] + \frac{\alpha_t^2 G^2 L}{2}.$$

From here we can use the same analysis that we used for Gradient descent to analyze the convergence of this algorithm. As before, it will converge to a noise ball if the learning rate is fixed.