Anytime Online-to-Batch, Optimism, and Acceleration

Ashok Cutkosky
Google Research

Stochastic Optimization

First-Order Stochastic Optimization

Find the minimum of some convex function $F : W \to \mathbb{R}$ using a stochastic gradient oracle: given w we can obtain a random variable g where $\mathbb{E}[g] = \nabla F(w)$.

Example: Stochastic Gradient Descent

A popular algorithm is gradient descent:

$$
\begin{aligned}
w_1 &= 0 \\
w_{t+1} &= w_t - \eta_t g_t
\end{aligned}
$$

Example: Stochastic Gradient Descent

A popular algorithm is gradient descent:

$$
\begin{aligned}
w_1 &= 0 \\
w_{t+1} &= w_t - \eta_t g_t
\end{aligned}
$$

How should we analyze its convergence?

Online Optimization

For $t = 1 \ldots T$, repeat:

1. Learner chooses a point w_t.
2. Environment presents learner with a gradient g_t (think $\mathbb{E}[g_t] = \nabla F(w_t)$).
3. Learner suffers loss $\langle g_t, w_t \rangle$.

The objective is minimize *regret*:

$$R_T(w_\star) = \sum_{t=1}^{T} \underbrace{\langle g_t, w_t \rangle}_{\text{loss suffered}} - \underbrace{\langle g_t, w_\star \rangle}_{\text{benchmark loss}}$$

Back to Gradient Descent

$$w_{t+1} = w_t - \eta_t g_t$$

Simplest analysis chooses $\eta_t \propto 1/\sqrt{T}$, but can also do more complicated things like $\eta_t \propto \frac{1}{\sqrt{\sum_{t=1}^{T} \|g_t\|^2}}$.

Back to Gradient Descent

$$w_{t+1} = w_t - \eta_t g_t$$

Simplest analysis chooses $\eta_t \propto 1/\sqrt{T}$, but can also do more complicated things like $\eta_t \propto \frac{1}{\sqrt{\sum_{t=1}^{T} \|g_t\|^2}}$.These yield

$$R_T(w_\star) \leq \|w_\star\|\sqrt{T}$$

$$R_T(w_\star) \leq \|w_\star\|\sqrt{\sum_{t=1}^{T} \|g_t\|^2}$$

Back to Gradient Descent

$$w_{t+1} = w_t - \eta_t g_t$$

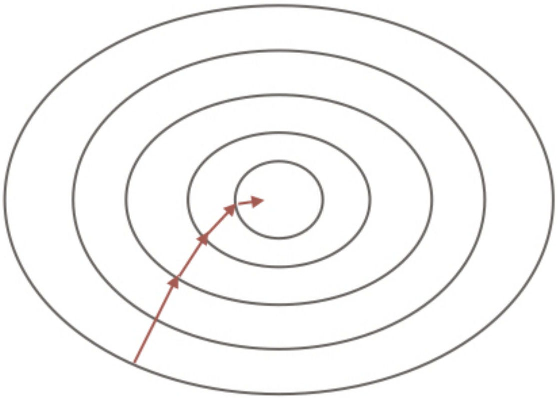
Simplest analysis chooses $\eta_t \propto 1/\sqrt{T}$, but can also do more complicated things like $\eta_t \propto \frac{1}{\sqrt{\sum_{t=1}^{T} \|g_t\|^2}}$.These yield

$$R_T(w_\star) \le \|w_\star\|\sqrt{T}$$

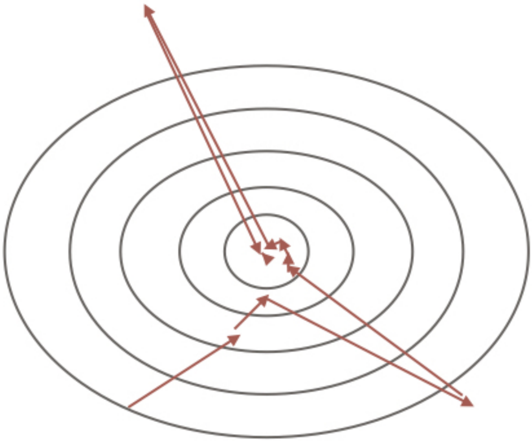
$$R_T(w_\star) \le \|w_\star\|\sqrt{\sum_{t=1}^{T} \|g_t\|^2}$$

We want to use regret bounds to solve stochastic optimization.

What We Hope Happens



What Could Happen Instead



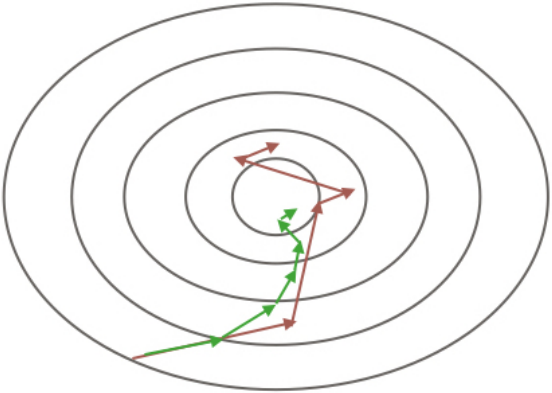
Online-to-Batch Conversion

- Run an online learner for T steps on gradients $\mathbb{E}[g_t] = \nabla F(w_t)$.
- Pick $\hat{w} = \frac{1}{T}\sum_{t=1}^{T} w_t$.
- $\mathbb{E}[F(\hat{w}) - F(w_\star)] \leq \frac{\mathbb{E}[R_T(w_\star)]}{T}$

Online-to-Batch Conversion

- Run an online learner for T steps on gradients $\mathbb{E}[g_t] = \nabla F(w_t)$.
- Pick $\hat{w} = \frac{1}{T}\sum_{t=1}^{T} w_t$.
- $\mathbb{E}[F(\hat{w}) - F(w_\star)] \leq \frac{\mathbb{E}[R_T(w_\star)]}{T}$
- For example: $\frac{\|w_\star\|\sqrt{\sum_{t=1}^{T}\|g_t\|^2}}{T} = O(1/\sqrt{T})$.

Averages Converge



Something That Could Be Better

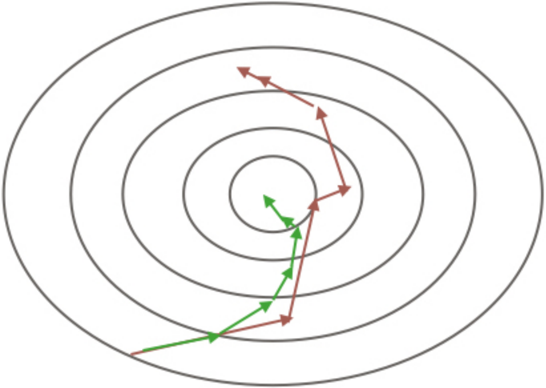
- The conversion is not "anytime": you must stop and average in order to get a convergence guarantee.
- The iterates w_t are still not well-behaved. For example, $\|\nabla F(w_T)\|$ may be much larger than $\|\nabla F(\hat{w})\|$.

Simple Fix

Just evaluate gradients at running averages!

- Let $x_t = \frac{1}{t} \sum_{i=1}^{t} w_i$
- Let g_t be stochastic gradient at x_t.
- Send g_t to online learner and get w_{t+1}.

Using Running Averages



Notation Recap

- x_t: where we evaluate gradients g_t.
- w_t: iterate of online learner (now exists only for analysis).
- $R_T(w_\star) = \sum_{t=1}^T \langle g_t, w_t - w_\star \rangle$.

No longer clear what the relationship is between R_T and the original loss function F since g_t is no longer a gradient at w_t.

Online-To-Batch is unchanged

Theorem

Define

$$R_T(x_\star) = \sum_{t=1}^{T} \langle \alpha_t g_t, w_t - x_\star \rangle$$

$$x_t = \frac{\sum_{i=1}^{t} \alpha_i w_i}{\sum_{i=1}^{t} \alpha_i}$$

Then for all $x_\star$ and all T,

$$\mathbb{E}[F(x_T) - F(x_\star)] \le \mathbb{E}\left[\frac{R_T(x_\star)}{\sum_{t=1}^{T} \alpha_t}\right]$$

Proof Sketch

Suppose $\alpha_t = 1$ for simplicity.

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} F(x_t) - F(x_\star)\right] &\leq \mathbb{E}\left[\sum_{t=1}^{T} \langle g_t, x_t - x_\star\rangle\right] \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T} \langle g_t, \underbrace{x_t - w_t}_{(t-1)(x_{t-1}-x_t)} \rangle + \underbrace{\langle g_t, w_t - x_\star\rangle}_{R_T(x_\star)}\right] \\
&\leq \mathbb{E}\left[R_T(x_\star) + \sum_{t=1}^{T} (t-1)(F(x_{t-1}) - F(x_t))\right]
\end{aligned}$$

Subtract $\sum_{t=1}^{T} F(x_t)$ from both sides, and telescope.

Stability

It's clear that $F(x_t) \to F(x_\star)$. But (in a bounded domain) we also have:

$$x_t - x_{t-1} = \frac{\alpha_t(x_t - w_t)}{\sum_{i=1}^{t-1} \alpha_i} = O(1/t)$$

In contrast, the iterates of the base online learner are less stable: $w_t - w_{t-1} = O(1/\sqrt{t})$ usually (because learning rate $\eta_t \propto 1/\sqrt{t}$).

An Algorithm That Likes Stability

Optimistic online learning algorithms can obtain [RS13; HK10; MY16]:

$$R_T(w_\star) \leq \sqrt{\sum_{t=1}^{T} \|g_t - g_{t-1}\|^2}$$

- This algorithm does better if the *gradients* are stable.

An Algorithm That Likes Stability

Optimistic online learning algorithms can obtain [RS13; HK10; MY16]:

$$R_T(w_\star) \le \sqrt{\sum_{t=1}^{T} \|g_t - g_{t-1}\|^2}$$

- This algorithm does better if the *gradients* are stable.
- When F is smooth, then gradient stability is implied by iterate stability!

Using Optimism with Stability

- With previous conversion, we might hope that $w_t - w_{t-1} = O(1/\sqrt{t})$. This implies

$$\mathbb{E}[F(\hat{w}_T) - F(x_\star)] \le O\left(\frac{1}{T} + \frac{\sigma}{\sqrt{T}}\right)$$

- In the new conversion, $g_t - g_{t-1} \approx x_t - x_{t-1} = O(1/t)$, so we can do much better.

Faster Rates with Optimism

Theorem

Suppose

$$R_T(x_\star) \leq \sqrt{\sum_{t=1}^{T} \alpha_t^2 \|g_t - g_{t-1}\|^2}$$

Set $\alpha_t = t$ for all t. Suppose each g_t has variance at most σ^2, and F is L-smooth. Then

$$\mathbb{E}[F(x_T) - F(x_\star)] \leq O\left(\frac{L}{T^{3/2}} + \frac{\sigma}{\sqrt{T}}\right)$$

Acceleration

The optimal rate is

$$\mathbb{E}[F(x_T) - F(x_\star)] \leq \frac{L}{T^2} + \frac{\sigma}{\sqrt{T}}$$

Acceleration

The optimal rate is

$$\mathbb{E}[F(x_T) - F(x_\star)] \leq \frac{L}{T^2} + \frac{\sigma}{\sqrt{T}}$$

- A small change to the algorithm can get this rate too.
- The algorithm does not know L or σ.
- Unfortunately, the algebra no longer fits on a slide.

Online-to-Batch Summary

- Evaluate gradients at running averages.
- Keeps the same convergence guarantee, but is anytime.
- Stabilizes the iterates $\longrightarrow$ faster rates on smooth problems.

Online-to-Batch Summary

- Evaluate gradients at running averages.
- Keeps the same convergence guarantee, but is anytime.
- Stabilizes the iterates $\longrightarrow$ faster rates on smooth problems.

Thank you!