Distributed Optimization for Machine Learning

Lecture 10 - Variance reduction and momentum for SGD

Tianyi Chen

School of Electrical and Computer Engineering
Cornell Tech, Cornell University

September 29, 2025

Review: Empirical risk minimization

Let $\{\mathbf{a}_i, y_i\}_{i=1}^n$ be n random samples, and consider

$$\min_{\mathbf{x}} \underbrace{F(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}; \{\mathbf{a}_i, y_i\})}_{\text{empirical risk}}$$

e.g. quadratic loss $f(\mathbf{x}; \{\mathbf{a}_i, y_i\}) = (\mathbf{a}_i^\top \mathbf{x} - y_i)^2$.

If one draws index $j \sim \mathrm{Unif}(1, \ldots, n)$ uniformly at random, then

$$F(\mathbf{x}) = \mathbb{E}_j[f(\mathbf{x}; \{\mathbf{a}_j, y_j\})]$$

The problem of variance in SGD

From previous lectures, we know the SGD update rule is:

$$\mathbf{x}^{t+1} = \mathbf{x}^t - \eta_t g(\mathbf{x}^t; \xi^t)$$

where $g(\mathbf{x}^t; \xi^t)$ is an unbiased estimate of $\nabla F(\mathbf{x}^t)$.

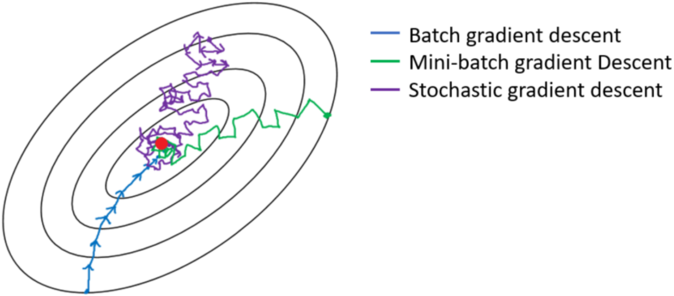
We established that this stochastic gradient has bounded variance:

$$\mathbb{E}[\|g(\mathbf{x}^t; \xi^t)\|_2^2] \leq \sigma_g^2 + c_g \|\nabla F(\mathbf{x}^t)\|_2^2$$

The term σ_g^2 (intrinsic noise) remains non-negligible even when $\mathbf{x}^t$ is close to the optimum $\mathbf{x}^*$ (where $\nabla F(\mathbf{x}^t) \approx \mathbf{0}$). This causes:

- Oscillations around the minimum.
- Slow convergence, requiring very small learning rates.

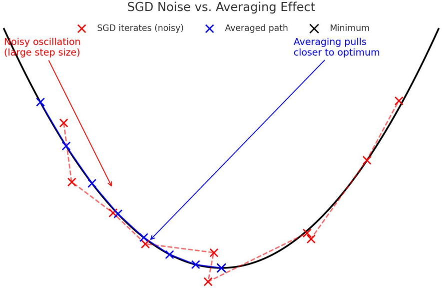
Recall the comparison between GD and SGD



Takeaway: Acceleration by averaging the stochastic gradients (high cost)

Table of Contents

Acceleration by averaging the iterates



Iterate averaging returns

$$\bar{\mathbf{x}}^t := \frac{1}{t} \sum_{i=0}^{t-1} \mathbf{x}^i$$

with larger stepsizes $\eta_t = t^{-\alpha}$, $\alpha < 1$.

Last iterate vs. Averaged iterates in SGD

$$\min_{\mathbf{x}\in\mathbb{R}^d} \ \frac{1}{2}||\mathbf{x}||_2^2$$

Last iterate. $\mathbf{x}^t = (1-\eta)^t\mathbf{x}^0 - \eta\sum_{k=0}^{t-1}(1-\eta)^{t-1-k}\xi^k$

$$\lim_{t\to\infty}\mathbb{E}\|\mathbf{x}^t\|^2 = \frac{\eta}{2-\eta} \qquad \Rightarrow \text{ with } \eta = 1, \text{ variance floor } \mathcal{O}(1).$$

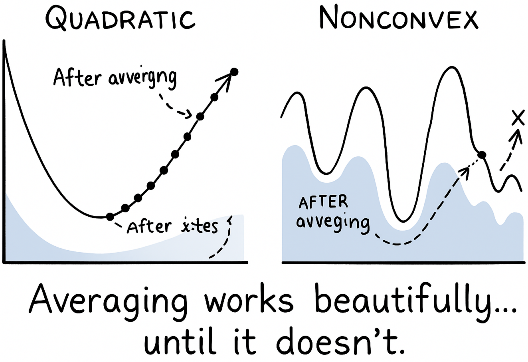
Averaged iterate. $\bar{\mathbf{x}}^t = \frac{1}{t}\sum_{j=0}^{t-1}\mathbf{x}^j \ \approx \ -\frac{1}{t}\sum_{k=0}^{t-1}\xi^k$

$$\sqrt{t}\,\bar{\mathbf{x}}^t \xrightarrow{d} \mathcal{N}(0,\mathbf{I}) \quad \Rightarrow \quad \mathbb{E}\|\bar{\mathbf{x}}^t\|^2 \approx \tfrac{d}{t}.$$

Takeaway:

- Last iterate: variance $\approx \mathcal{O}(1)$ (does not vanish).
- Averaged iterate: variance $\approx \mathcal{O}(1/t)$ (vanishes).

Averaged iterates may fail



(Credit: Generated by ChatGPT5)

Averaging may fail for non-quadratic objectives

Consider a non-quadratic function:

$$f(x) = \frac{1}{4}x^4 \quad \Rightarrow \quad \nabla f(x) = x^3.$$

SGD with constant stepsize $\eta_t \equiv \eta$:

$$x^{t+1} = x^t - \eta\big((x^t)^3 + \xi^t\big), \quad \text{where } \mathbb{E}[\xi^t] = 0.$$

Observation: Unlike the quadratic case, the dynamics are *nonlinear and biased*. The stochastic term interacts with $(x^t)^3$, so averaging iterates no longer cancels the noise cleanly.

Key message: For non-quadratic f, the iterate distribution is asymmetric, so $\bar{x}^t$ is not an unbiased estimator of the optimum.

When iterate averaging helps - and when it does not

Quadratic case ($f(x) = \frac{1}{2}x^2$):

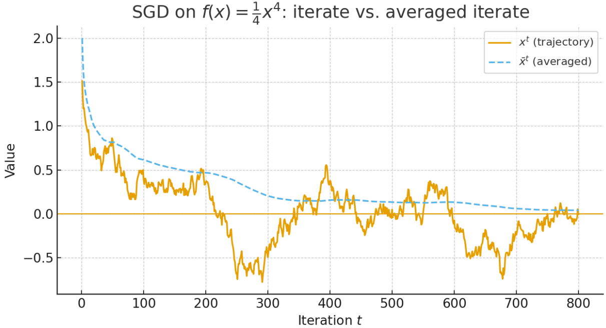
- SGD dynamics are linear: $x^{t+1} = (1-\eta)x^t - \eta\xi^t$.
- Averaging cancels zero-mean noise $\Rightarrow$ variance $\sim 1/t$.

Non-quadratic case ($f(x) = \frac{1}{4}x^4$ or nonconvex f):

- Dynamics are nonlinear: noise interacts multiplicatively with x^t.
- The iterates are *asymmetrically distributed*, leading to a biased $\bar{x}^t$.
- Variance may decrease, but bias dominates $\Rightarrow$ **no true acceleration.**

Takeaway: Iterate averaging works beautifully for quadratics (linear dynamics), but can fail or even slow convergence for nonlinear cases.

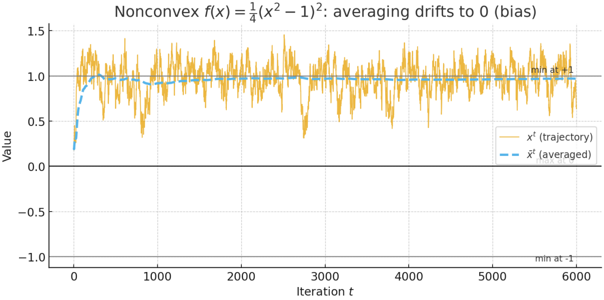
Example: $f(x) = \frac{1}{4}x^4$ with noisy gradients



(Illustration: SGD trajectories and their averages)

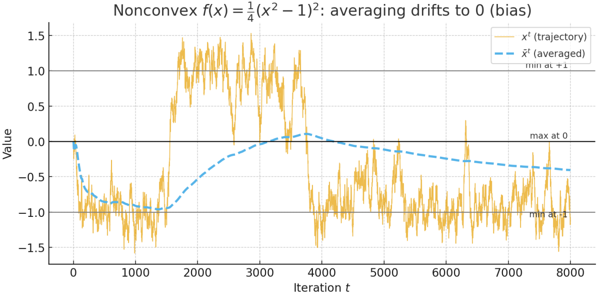
Why: For large $|x|$, the gradient magnitude $|x^3|$ is large, so SGD spends less time far from the origin, making the time-averaged $\bar{x}^t$ **not representative** of the stationary point.

Example: Nonconvex non-quadratic with noisy gradients



(Illustration: SGD trajectories and their averages)

Example: Nonconvex non-quadratic with noisy gradients



(Illustration: SGD trajectories and their averages)

Why: This produces a trajectory that spends time in both minima, so the mean of the iterates sits near 0, even though 0 is not a minimizer - exactly the failure mode in nonconvex landscapes.

Table of Contents

A simple idea of variance reduction

Imagine we take some $\mathbf{e}^t$ with $\mathbb{E}[\mathbf{e}^t] = \mathbf{0}$ and set stochastic gradient as

$$\mathbf{v}^t = g(\mathbf{x}^t; \xi^t) - \mathbf{e}^t$$

— so $\mathbf{v}^t$ is still an unbiased estimate of $\nabla F(\mathbf{x}^t)$

Question: how to reduce variability (i.e. $\mathbb{E}[\|\mathbf{v}^t\|_2^2] < \mathbb{E}[\|g(\mathbf{x}^t; \xi^t)\|_2^2]$)?

Answer: find some zero-mean $\mathbf{e}^t$ that is positively correlated with $g(\mathbf{x}^t; \xi^t)$ (i.e. $\langle \mathbf{e}^t, g(\mathbf{x}^t; \xi^t) \rangle > 0$) (why? whiteboard)

Example: Reducing variance using control variates

Goal: Estimate $\mu = \mathbb{E}[Y]$ where $Y = X^2$ and $X \sim \text{Uniform}[0, 1]$.

- The true mean is $\mathbb{E}[Y] = \int_0^1 x^2 dx = \frac{1}{3} \approx 0.333$.

We'll use Monte Carlo sampling with $n = 5$ samples:

Sample	X_i	$Y_i = X_i^2$
1	0.1	0.01
2	0.3	0.09
3	0.7	0.49
4	0.9	0.81
5	0.5	0.25

$$\hat{\mu}_{\text{naive}} = \frac{1}{5} \sum_{i=1}^{5} Y_i = 0.33$$

Observation: The estimate is close to the truth, but has high variance.

Question: Can we reduce variance without introducing bias?

Example: Reducing variance using control variates

We introduce a correlated variable $Z_i = X_i$ with $\mathbb{E}[Z_i] = 0.5$.

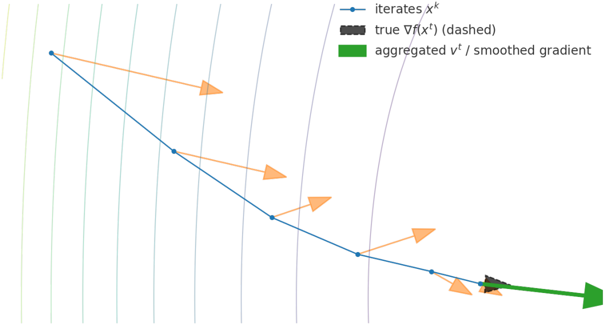
$$\hat{\mu}_{\text{cv}} = \frac{1}{5}\sum_{i=1}^{5}\big(Y_i - c(Z_i - \mathbb{E}[Z])\big)$$

Sample	X_i	$Y_i = X_i^2$	$Z_i = X_i$	$Y_i - c(Z_i - \mathbb{E}[Z])$
1	0.1	0.01	0.1	0.01 - 0.6(0.1 - 0.5) $=$ 0.25
2	0.3	0.09	0.3	0.09 - 0.6(0.3 - 0.5) $=$ 0.21
3	0.7	0.49	0.7	0.49 - 0.6(0.7 - 0.5) $=$ 0.37
4	0.9	0.81	0.9	0.81 - 0.6(0.9 - 0.5) $=$ 0.57
5	0.5	0.25	0.5	0.25 - 0.6(0.5 - 0.5) $=$ 0.25
			$\hat{\mu}_{\text{cv}} = \frac{1}{5}\sum_{i=1}^{5}(\cdot)$	**0.33**

Table: Control variate estimator with $c = 0.6$ and $\mathbb{E}[Z] = 0.5$.

The mean remains unchanged - the estimator is **unbiased**.

Reducing variance via gradient aggregation



Main idea: If the current iterate is not too far away from previous iterates, then historical gradients might be useful in producing $\mathbf{e}^t$

Recall the heavy-ball method

Recall the Heavy-ball method as

$$\mathbf{x}^{t+1} = \mathbf{x}^t - \eta_t \nabla F(\mathbf{x}^t) + \theta_t(\mathbf{x}^t - \mathbf{x}^{t-1})$$

Here, $\theta_t(\mathbf{x}^t - \mathbf{x}^{t-1})$ is the momentum term, proportional to the step.

Imagine a heavy ball rolling down a hilly landscape.

- The gradient $(\nabla F(\mathbf{x}^t))$ acts like gravity, pulling it downhill.
- The momentum term $(\theta_t(\mathbf{x}^t - \mathbf{x}^{t-1}))$ acts like inertia, helping the ball continue in its previous direction, smoothing out sharp turns and accelerating through flat regions.

From Heavy-ball to Momentum SGD: equivalent views

Heavy-ball update can be rewritten by introducing a **velocity variable**:

$$\mathbf{v}^{t+1} := \mathbf{x}^{t+1} - \mathbf{x}^{t} \quad \Longleftrightarrow \quad \mathbf{v}^{t+1} = \theta_t \mathbf{v}^{t} - \eta_t \nabla F(\mathbf{x}^{t})$$

- The "momentum" $\mathbf{v}^{t+1}$ stores information from past updates.
- Each new gradient $\nabla F(\mathbf{x}^{t})$ modifies this moving direction.

When we move to the stochastic setting, we use the stochastic gradient $g(\mathbf{x}^{t}; \xi^{t})$ and maintain an exponentially weighted moving average:

$$\mathbf{v}^{t+1} = \theta \mathbf{v}^{t} + (1-\theta) g(\mathbf{x}^{t}; \xi^{t})$$

By recursively substituting $\mathbf{v}^{t}$, we can see that $\mathbf{v}^{t+1}$ is a weighted average of all past stochastic gradients (assuming $\mathbf{v}^{-1} = \mathbf{0}$ for simplicity):

$$\begin{aligned}\mathbf{v}^{t+1} =& (1-\theta) g(\mathbf{x}^{t}; \xi^{t}) + \theta(1-\theta) g(\mathbf{x}^{t-1}; \xi^{t-1}) + \theta^2 (1-\theta) g(\mathbf{x}^{t-2}; \xi^{t-2}) \\ & + \cdots + \theta^{t}(1-\theta) g(\mathbf{x}^{0}; \xi^{0}) + \theta^{t+1} \mathbf{v}^{-1}\end{aligned}$$

Momentum SGD as a weighted average of past gradients

Main idea: If the current iterate is not too far away from previous iterates, then historical gradients might be useful in producing $\mathbf{v}^t$

The update for $\mathbf{x}^{t+1}$ can be rewritten as:

$$\mathbf{v}^{t+1} = \theta \mathbf{v}^t + (1-\theta) g(\mathbf{x}^t; \xi^t)$$
$$\mathbf{x}^{t+1} = \mathbf{x}^t - \eta_t \mathbf{v}^{t+1}$$

where $\theta \in [0, 1)$ is the momentum coefficient.

The momentum term $\theta \mathbf{v}^t$ acts as a "control variate" that effectively subtracts out some of the noise in the current stochastic gradient $g(\mathbf{x}^t; \xi^t)$, guiding the update in a more stable direction.

Momentum SGD is thus the stochastic version of Heavy-ball.

Momentum term's unbiasedness

Taking the expectation conditional on $\mathbf{x}^t$:

$$\mathbb{E}[\mathbf{v}^{t+1}|\mathbf{x}^t] = \theta\mathbb{E}[\mathbf{v}^t|\mathbf{x}^t] + (1-\theta)\mathbb{E}[g(\mathbf{x}^t;\xi^t)|\mathbf{x}^t].$$

If we start with $\mathbf{v}^0 = \nabla F(\mathbf{x}^0)$ (or $\mathbf{0}$ and warm up), and if we consider $\mathbf{x}^t$ to be fixed, then $\mathbb{E}[g(\mathbf{x}^t;\xi^t)|\mathbf{x}^t] = \nabla F(\mathbf{x}^t)$; that is

$$\mathbb{E}[\mathbf{v}^{t+1}|\mathbf{x}^t] = \theta\mathbb{E}[\mathbf{v}^t|\mathbf{x}^t] + (1-\theta)\nabla F(\mathbf{x}^t)$$

In a stationary setting ($\mathbf{x}$ constant), $\mathbf{v}$ would converge to $\nabla F(\mathbf{x})$.

In the context of variance reduction, we treat the momentum update direction $\mathbf{v}^{t+1}$ itself as our new gradient estimate. $\mathbb{E}[\mathbf{v}^{t+1}] \approx \nabla F(\mathbf{x}^t)$ (This is an approximation due to changing $\mathbf{x}^t$, but holds for small η).

Noise reduction

Consider the noise component $\epsilon^t = g(\mathbf{x}^t; \xi^t) - \nabla F(\mathbf{x}^t)$, where $\mathbb{E}[\epsilon^t] = \mathbf{0}$ and $\mathbb{E}[\|\epsilon^t\|_2^2] \leq \sigma_g^2 + (c_g - 1)\|\nabla F(\mathbf{x}^t)\|_2^2$. The momentum becomes:

$$\mathbf{v}^{t+1} \approx \theta \mathbf{v}^t + (1-\theta)(\nabla F(\mathbf{x}^t) + \epsilon^t)$$

which is a moving average of the gradients and the noise components.

Because $\mathbb{E}[\epsilon^t] = \mathbf{0}$, averaging several ϵ^t terms together (which is what $\mathbf{v}^{t+1}$ does over time) tends to reduce the overall magnitude of the noise.

$$\mathbb{E}\left[\left\|\sum_{j=0}^{t} \theta^j (1-\theta)\epsilon^{t-j}\right\|_2^2\right] \propto \frac{(1-\theta)^2}{1-\theta^2}\sigma_g^2 \approx \frac{1-\theta}{1+\theta}\sigma_g^2$$

Vanilla SGD vs. SGD with momentum

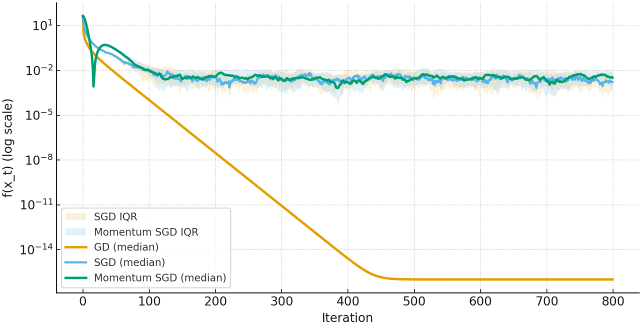
Vanilla SGD ($B = 1$):

- $g(\mathbf{x}^t; \xi^t) = \nabla f_{i_t}(\mathbf{x}^t)$ (single sample)
- $\mathbb{E}[\|g(\mathbf{x}^t; \xi^t)\|_2^2] \leq \sigma_g^2 + c_g \|\nabla F(\mathbf{x}^t)\|_2^2$
- Prone to high variance in update, especially when $\nabla F(\mathbf{x}^t)$ is small.

SGD with Momentum

- $\mathbf{v}^{t+1} = \theta \mathbf{v}^t + (1 - \theta) g(\mathbf{x}^t; \xi^t)$
- The effective variance σ_{mom}^2 of $\mathbf{v}^{t+1}$ is much smaller than σ_g^2.
- $\mathbb{E}[\|\mathbf{v}^{t+1}\|_2^2] \approx \frac{(1-\theta)^2}{1-\theta^2} \sigma_g^2 + \ldots \quad$ (simplified)
- **Result:** Faster, more stable convergence, especially in the tail phase when approaching the minimum.

Comparison between GD, SGD and momentum SGD



Takeaway: momentum SGD does not fundamentally eliminate variance.

Table of Contents

Momentum helps... but only to a certain extent

Momentum smooths the gradient noise by averaging recent updates:

$$\mathbf{v}^{t+1} = \theta \mathbf{v}^t + (1-\theta) g(\mathbf{x}^t; \xi^t).$$

- It reduces the short-term fluctuations of stochastic gradients.
- But it does **not eliminate the bias** from using noisy samples - the expected gradient still fluctuates around $\nabla F(\mathbf{x}^t)$.
- Moreover, once the iterates move far from previous ones, the accumulated information in $\mathbf{v}^t$ becomes stale.

Momentum helps... but only to a certain extent

Variance reduction via momentum is implicit and local.

Question: Can we achieve *explicit* variance reduction - not just smooth the noise, but actually construct a lower-variance gradient estimator?

Variance reduction for finite-sum minimization

Idea: Instead of smoothing gradients implicitly, can we **correct** noisy stochastic gradients using information from the full dataset?

In our discussion so far, we focus on the following stochastic problem

$$F(\mathbf{x}) = \mathbb{E}_\xi[f(\mathbf{x};\xi)]$$

which we call it as the "online" problem hereafter.

But we in fact first collect n offline training samples in $\{\xi_i\}_{i=1}^n$, and solve

$$\min_{\mathbf{x}} \ \underbrace{F(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^n f(\mathbf{x};\xi_i)}_{\text{empirical risk}}$$

which we call it as the "offline" problem hereafter.

SVRG's variance-reduced gradient estimator

Key difference: SVRG replaces noisy gradients with a corrected version that re-centers them around the full gradient at a *snapshot point*.

- Periodically calculate the **full gradient** at a "snapshot" point $\tilde{\mathbf{x}}$.
- Use this full gradient as a "low-variance anchor" to correct

$$\mathbf{v}^t_{\mathrm{SVRG}} = \nabla f_{i_t}(\mathbf{x}^t) - \nabla f_{i_t}(\tilde{\mathbf{x}}) + \nabla F(\tilde{\mathbf{x}})$$

- f_{i_t}: gradient for the current random sample i_t.
- $\tilde{\mathbf{x}}$: snapshot point (updated every epoch).
- $\nabla F(\tilde{\mathbf{x}})$: full gradient at $\tilde{\mathbf{x}}$.

The SVRG algorithm

for epoch $s = 1, 2, \ldots$

- Compute the full gradient at the snapshot:

$$\nabla F(\tilde{\mathbf{x}}) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\tilde{\mathbf{x}}) \quad \text{set snapshot } \tilde{\mathbf{x}} \text{ from the last epoch}$$

- **for** inner iteration $t = 1, \ldots, m$
  - Choose a sample i_t uniformly at random and compute

$$\mathbf{v}^t_{\mathrm{SVRG}} = \nabla f_{i_t}(\mathbf{x}^t) - \nabla f_{i_t}(\tilde{\mathbf{x}}) + \nabla F(\tilde{\mathbf{x}})$$

  - Run the gradient descent update: $\mathbf{x}^{t+1} = \mathbf{x}^t - \eta \mathbf{v}^t_{\mathrm{SVRG}}$

  end for

end for

- Full gradient calculation is expensive but only done once per epoch.
- Inside the epoch, computations are two stochastic gradients.

SVRG: How it reduces variance

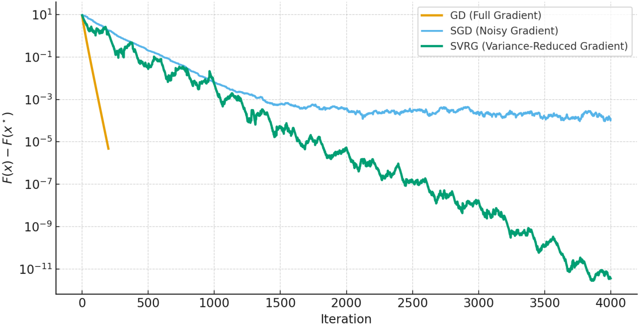
Let's look at the "noise" of the SVRG gradient:

$$\mathbf{v}^t_{\text{SVRG}} - \nabla F(\mathbf{x}^t) = (\nabla f_{i_t}(\mathbf{x}^t) - \nabla F(\mathbf{x}^t)) - (\nabla f_{i_t}(\tilde{\mathbf{x}}) - \nabla F(\tilde{\mathbf{x}}))$$

- **Unbiased:** Yes, $\mathbb{E}[\mathbf{v}^t_{\text{SVRG}}] = \nabla F(\mathbf{x}^t)$.
- **Key:** Both $\nabla f_{i_t}(\mathbf{x}^t)$ and $\nabla f_{i_t}(\tilde{\mathbf{x}})$ use the **same random sample** i_t.
  - This makes them highly correlated.
  - The "common noise" associated with sample i_t tends to cancel out in the difference term $\nabla f_{i_t}(\mathbf{x}^t) - \nabla f_{i_t}(\tilde{\mathbf{x}})$.
- **Variance bound:** The variance of $\mathbf{v}^t_{\text{SVRG}}$ is bounded by:

$$\text{Var}(\mathbf{v}^t_{\text{SVRG}}) \leq L^2 \|\mathbf{x}^t - \tilde{\mathbf{x}}\|_2^2$$

Comparison between GD, SGD and SVRG



Takeaway: Acceleration by explicit variance reduction

The SVRG's advantages and online challenge

- **SVRG's merits:** Leverages the finite-sum structure to introduce *explicit variance reduction*:
  - Able to use constant stepsize as GD.
  - Able to converge as the same convergence rate as GD.
  - Average per-iteration cost of SVRG is comparable to that of SGD

- **SVRG's limitation:** Relies on periodic *full gradient computations*, which can be expensive or impossible in:
  - Online learning (data arrives as a stream).
  - Extremely large datasets where a full pass is too slow.

Recap and fine-tuning

- What we have talked about today?
  - $\Rightarrow$ How to reduce variance by averaging iterates?
  - $\Rightarrow$ How to reduce variance by momentum?
  - $\Rightarrow$ How to reduce variance by using the finite-sum structure?

Welcome anonymous survey!