Machine Learning and Optimization

Lecture 7: Stochastic Gradient Descent

CSCI 4140/6960 — Spring 2026

Course website: `cs.rpi.edu/~gittea/.../mlandopt.html`

Learning Objectives

By the end of this lecture, you should be able to:

1. Explain why SGD is computationally necessary for large-scale learning and quantify the per-iteration speedup over gradient descent
2. Interpret the fundamental trade-off between progress and noise accumulation, and explain the Robbins–Monro conditions
3. State convergence rates for SGD in strongly convex and convex settings, understand why these rates are optimal, and explain the role of averaging
4. Compare theoretical prescriptions (diminishing step sizes, iterate averaging) with practical implementations (constant steps with decay, last-iterate use)

Outline

Why Stochastic Gradient Descent?

The Accounting Identity

Step-Size Principles

The Strongly Convex Case: A Clean Laboratory

The Convex Case and the Necessity of Averaging

Optimality and Improving Constants

Theory vs Practice

Conclusion

The Computational Barrier in Large-Scale Learning

Recall the empirical risk minimization problem:

$$\min_{x \in \mathbb{R}^d} f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x)$$

Standard gradient descent:

$$x_{t+1} = x_t - \alpha_t \nabla f(x_t) = x_t - \frac{\alpha_t}{n} \sum_{i=1}^{n} \nabla f_i(x_t)$$

Cost per iteration: $O(nd)$ — requires evaluating *all* n gradients.

When $n = 10^9$ (modern datasets), this is *computationally infeasible*.

Concrete Example: Computational Cost

Dataset: $n = 10^6$ examples, $d = 10^6$ parameters (neural network)

One GD iteration:

- Compute 10^6 gradients of dimension 10^6
- Total: 10^{12} floating point operations

For convergence (need ~ 1000 iterations):

- Total: 10^{15} operations

On modern GPU (10 TFLOPS):

- $10^{15}/10^{13} = 100$ seconds ≈ 2 minutes per run

For hyperparameter search (100 configurations):

- 2 min $\times\ 100 = 3$ hours

The computational bottleneck is the *data pass*, not the parameter update.

The Stochastic Approximation

Stochastic gradient estimator: If we sample $i_t \sim \text{Uniform}(\{1, \ldots, n\})$, then

$$\mathbb{E}[\nabla f_{i_t}(x_t)] = \nabla f(x_t)$$

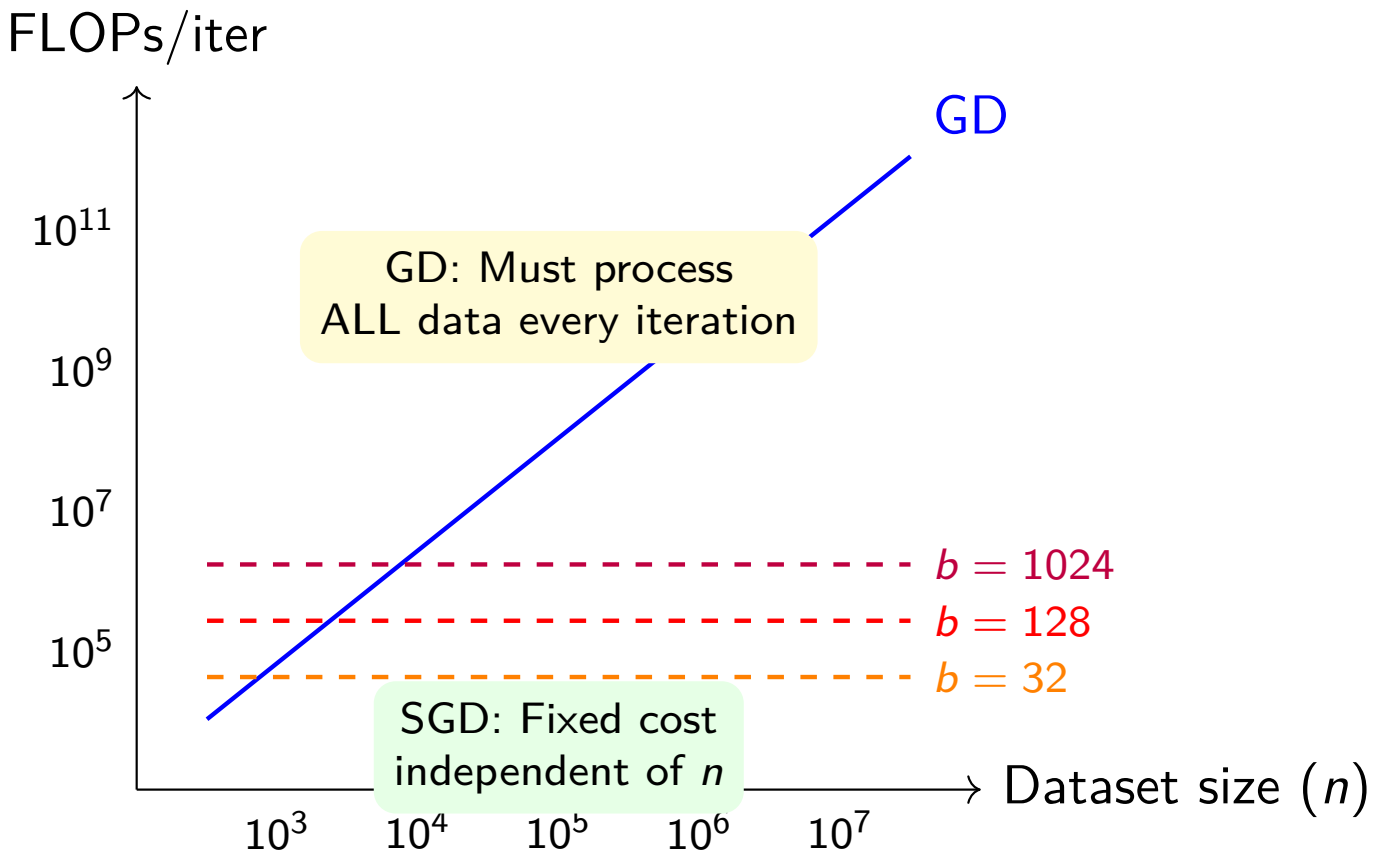
Stochastic gradient descent (SGD):

$$x_{t+1} = x_t - \alpha_t \nabla f_{i_t}(x_t)$$

Cost per iteration: $O(d)$ — **a factor of n speedup!**

The price: We use a *noisy estimate* of the gradient.

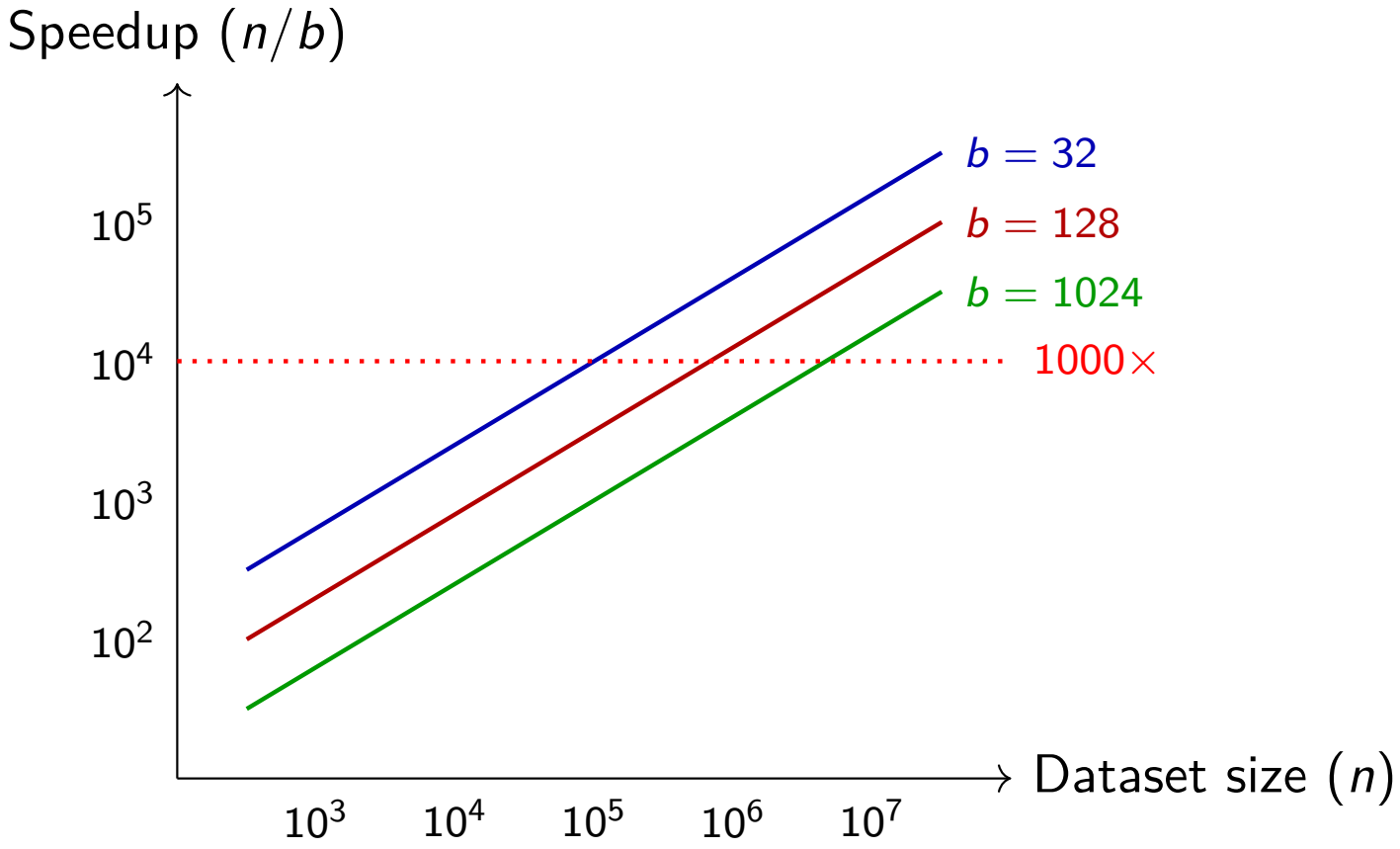
Per-iteration cost comparison



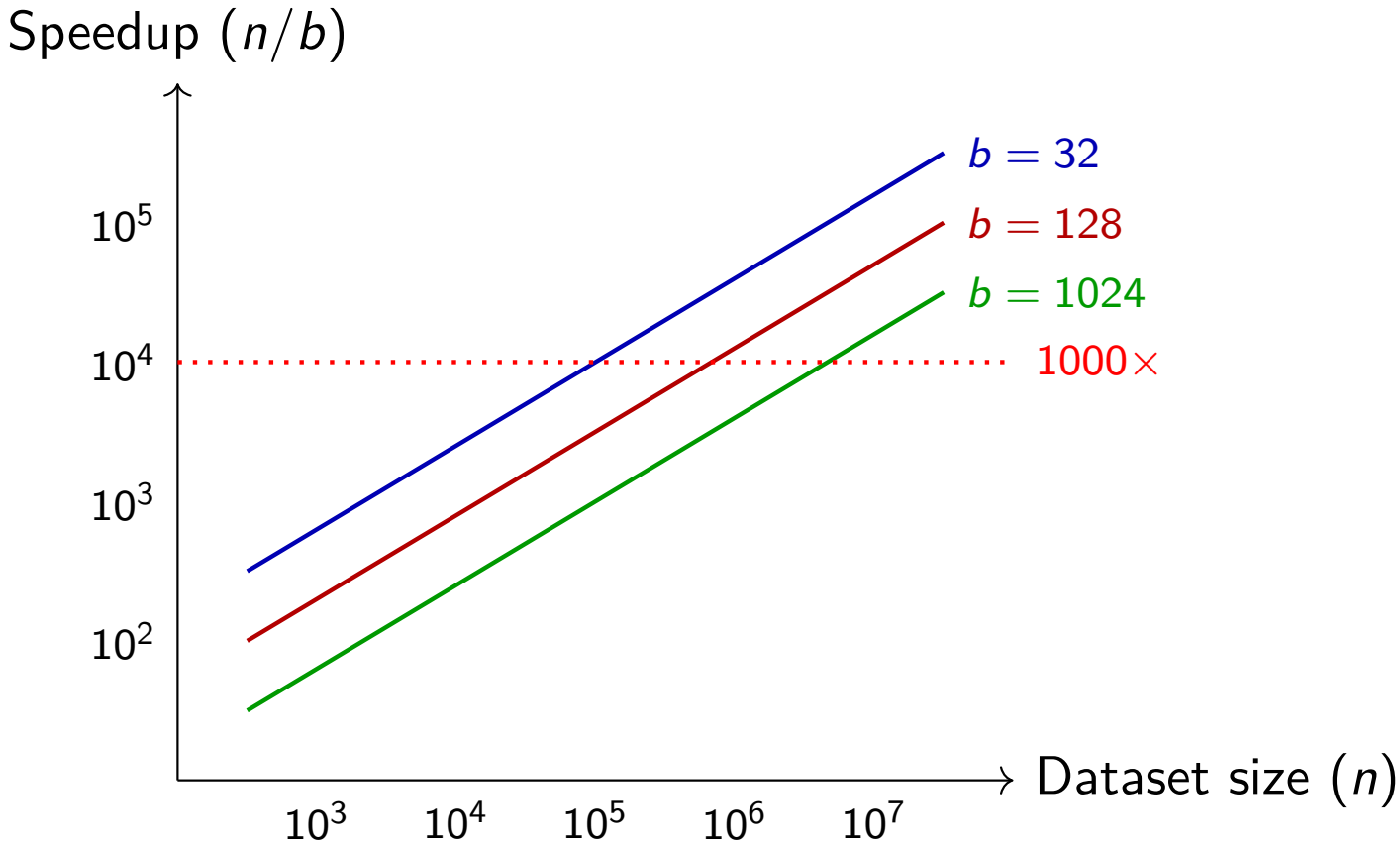
SGD cost is independent of n; GD cost grows linearly with n. For large n, this difference is huge.

Per-iteration speedup of SGD

Per-iteration speedup of SGD

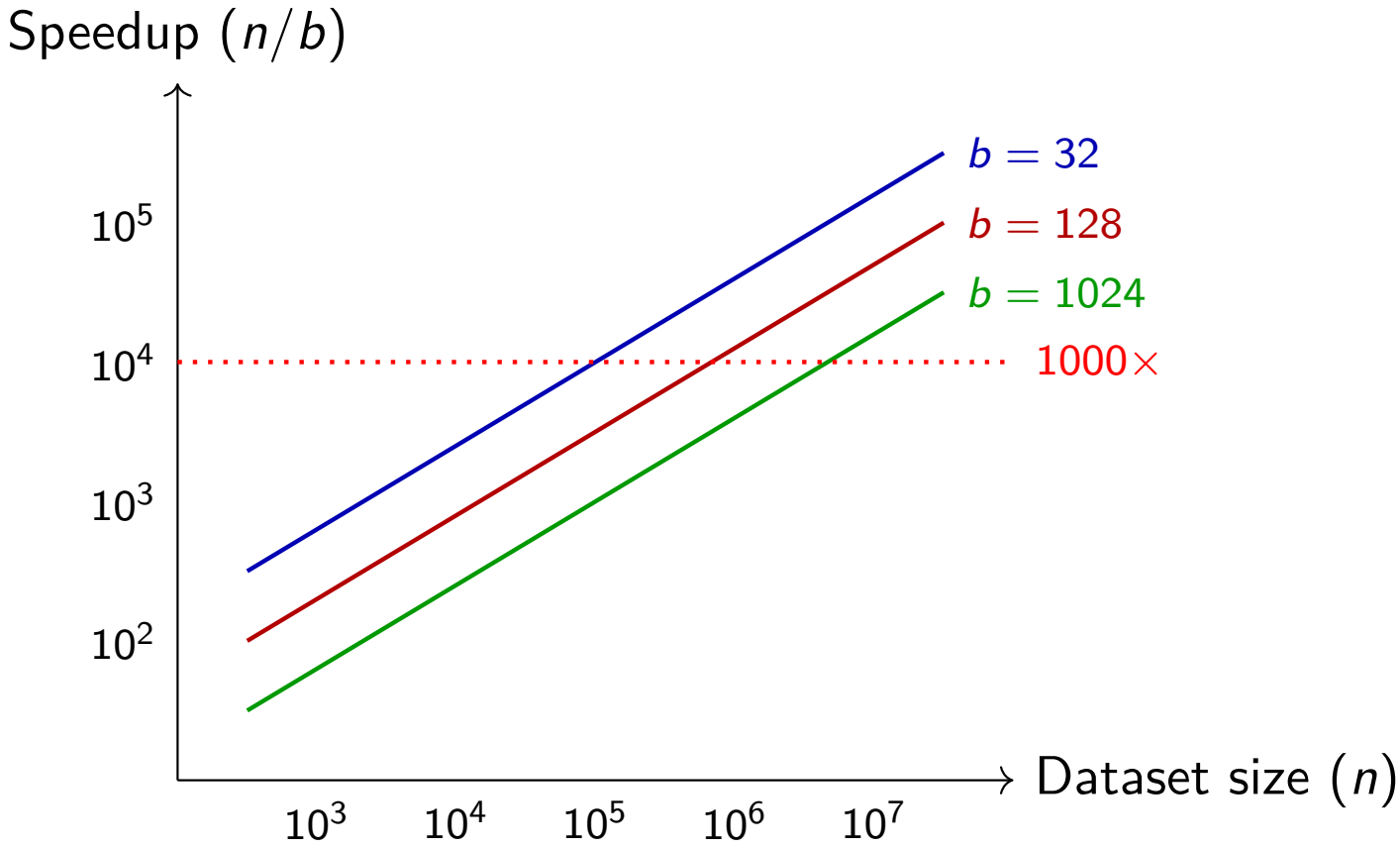


Per-iteration speedup of SGD



For $n = 10^6$, $b = 100$: we get $10^4\times$ speedup per iteration.

Per-iteration speedup of SGD



For $n = 10^6$, $b = 100$: we get $10^4\times$ speedup per iteration.

But this speedup comes at a cost—what is that cost?

What Changes and What Doesn't

Same in expectation[1]**:**

$$\mathbb{E}[x_{t+1} \mid x_t] = x_t - \alpha_t \nabla f(x_t)$$

The *expected* update is identical to gradient descent.

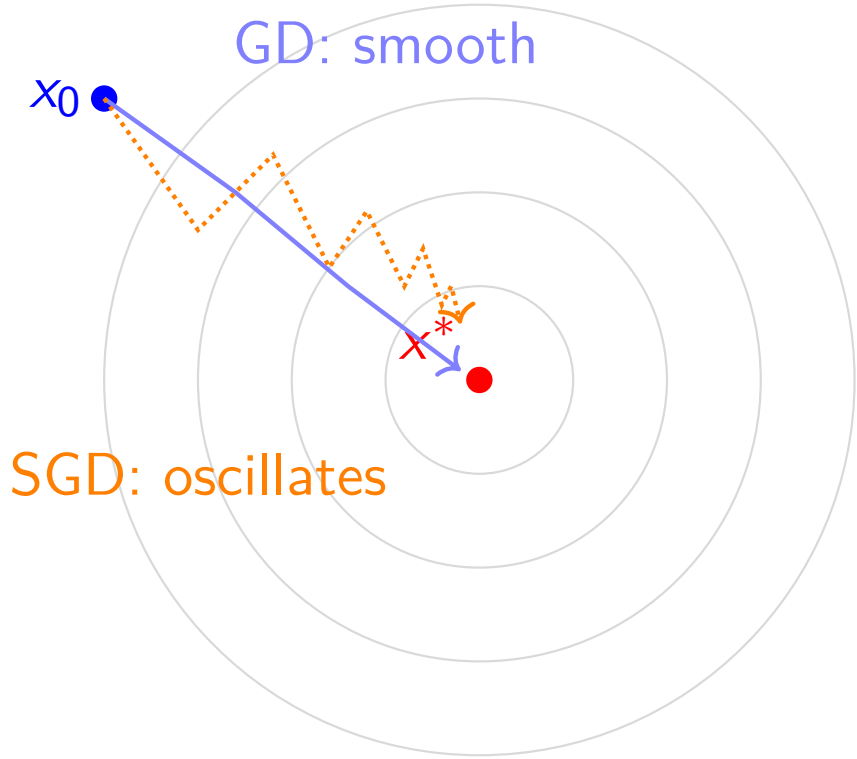
Different in variance:
The actual iterate x_t is a random variable. Each step introduces noise:

$$\underbrace{\nabla f_{i_t}(x_t)}_{\text{stochastic gradient}} = \underbrace{\nabla f(x_t)}_{\text{true gradient}} + \underbrace{\nabla f_{i_t}(x_t) - \nabla f(x_t)}_{\text{noise}}$$

Central tension: Progress toward optimum vs. accumulation of noise.

[1] Same expected step, **not same expected trajectory!**

Visualizing the Dynamics: Noise vs Progress



Observation: SGD makes progress on average but oscillates around the optimum.
Even near x^*, noise persists.

The Noise Floor

With a **constant step size** $\alpha_t = \alpha$:

- Early iterations: noise is small relative to distance from optimum
- Progress dominates

- Later iterations: distance to optimum shrinks
- But noise variance stays bounded: $\mathbb{E}[\|\nabla f_{i_t}(x_t) - \nabla f(x_t)\|^2] \leq \sigma^2$
- Noise prevents convergence to x^*

Result: Iterates oscillate in a ball around x^* of radius $\sim \sqrt{\frac{\alpha}{\mu}}\sigma$ in strongly convex case. Otherwise they can drift arbitrarily far away.

To converge: Must shrink step size to zero or decrease variance.

Measuring Progress: The One-Step Recursion

Let $x^* \in \operatorname{argmin}_x f(x)$ and measure distance $\|x_t - x^*\|^2$.

Expanding the update:

$$\begin{aligned}\|x_{t+1} - x^*\|^2 &= \|x_t - \alpha_t \nabla f_{i_t}(x_t) - x^*\|^2 \\ &= \|x_t - x^*\|^2 - 2\alpha_t \langle \nabla f_{i_t}(x_t), x_t - x^* \rangle + \alpha_t^2 \|\nabla f_{i_t}(x_t)\|^2\end{aligned}$$

This is a **bookkeeping equation**: tracks how distance changes in one step.

Interpreting the Terms

$$\|x_{t+1} - x^*\|^2 = \|x_t - x^*\|^2 - 2\alpha_t \langle \nabla f_{i_t}(x_t), x_t - x^* \rangle + \alpha_t^2 \|\nabla f_{i_t}(x_t)\|^2$$

Three terms:

1. $\|x_t - x^*\|^2$: current distance
2. $-2\alpha_t \langle \nabla f_{i_t}(x_t), x_t - x^* \rangle$: **progress term**
   - When $\langle \nabla f_{i_t}(x_t), x_t - x^* \rangle > 0$, distance decreases
   - Expected value is $-2\alpha_t \langle \nabla f(x_t), x_t - x^* \rangle$
3. $\alpha_t^2 \|\nabla f_{i_t}(x_t)\|^2$: **noise term**
   - Squared step size times squared gradient
   - Always positive: increases distance

Summing the Recursion: Two Competing Quantities

Rearrange and telescope from $t = 0$ to $T - 1$:

$$2\alpha_t \langle \nabla f_{i_t}(x_t), x_t - x^* \rangle = \|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2 + \alpha_t^2 \|\nabla f_{i_t}(x_t)\|^2$$

Sum over t:

$$2 \sum_{t=0}^{T-1} \alpha_t \langle \nabla f_{i_t}(x_t), x_t - x^* \rangle = \|x_0 - x^*\|^2 - \|x_T - x^*\|^2 + \sum_{t=0}^{T-1} \alpha_t^2 \|\nabla f_{i_t}(x_t)\|^2$$

Drop $-\|x_T - x^*\|^2 \leq 0$:

$$\sum_{t=0}^{T-1} \alpha_t \langle \nabla f_{i_t}(x_t), x_t - x^* \rangle \leq \frac{\|x_0 - x^*\|^2}{2} + \frac{1}{2} \sum_{t=0}^{T-1} \alpha_t^2 \|\nabla f_{i_t}(x_t)\|^2$$

The Fundamental Trade-off

$$\underbrace{\sum_{t=0}^{T-1} \alpha_t \langle \nabla f_{i_t}(x_t), x_t - x^* \rangle}_{\text{cumulative progress}} \leq \underbrace{\frac{\|x_0 - x^*\|^2}{2}}_{\text{initial distance}} + \underbrace{\frac{1}{2} \sum_{t=0}^{T-1} \alpha_t^2 \|\nabla f_{i_t}(x_t)\|^2}_{\text{cumulative noise}}$$

Key observations:

- Progress accumulates as $\sum \alpha_t$
- Noise accumulates as $\sum \alpha_t^2$
- To maximize progress while controlling noise: need $\sum \alpha_t \to \infty$ but $\sum \alpha_t^2 < \infty$

This observation dictates which step-size schedules are useful.

The Robbins–Monro Conditions

Robbins and Monro (1951) showed necessary conditions for controlling noise in stochastic approximation:

Step sizes must satisfy:

1. $\sum_{t=0}^{\infty} \alpha_t = \infty$ (infinite progress)

2. $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$ (finite noise accumulation)

Intuition:

- Condition (1): step sizes should not vanish too quickly, or algorithm stalls
- Condition (2): step sizes should vanish fast enough to eliminate noise

Examples of Valid Schedules

Prototypical choice:

$$\alpha_t = \frac{c}{t+1} \quad \text{for } c > 0$$

Check Robbins–Monro:

- $\sum_{t=0}^{\infty} \frac{c}{t+1} = c \sum_{k=1}^{\infty} \frac{1}{k} = \infty$ ✓
- $\sum_{t=0}^{\infty} \frac{c^2}{(t+1)^2} = c^2 \sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2 c^2}{6} < \infty$ ✓

General polynomial schedules:

$$\alpha_t = \frac{c}{(t+1)^\beta}$$

- Valid when $\frac{1}{2} < \beta \leq 1$
- $\beta = 1$: most common in theory
- $\beta < 1$: slower decay, more aggressive early steps

What About Constant Step Sizes?

Constant step size: $\alpha_t = \alpha$ for all t.

Robbins–Monro check:

- $\sum_{t=0}^{\infty} \alpha = \infty \quad \checkmark$
- $\sum_{t=0}^{\infty} \alpha^2 = \infty \quad \times$

Consequence: Noise prevents convergence to a point; iterates converge to a stationary distribution with variance $\frac{\alpha}{\mu}\sigma^2$.
Theory predicts exactly what we see in practice: fast initial progress followed by stochastic oscillation at the noise floor.

Practical use: Constant step sizes are common in practice for speed, followed by manual step-size reduction ("learning rate decay").

Conceptual Summary: Step-Size Principles

What we learned:

- The one-step recursion reveals a **progress vs noise** trade-off
- Summing yields cumulative bounds: $\sum \alpha_t$ (progress) vs $\sum \alpha_t^2$ (noise)
- Robbins–Monro conditions ensure progress with bounded noise
- $\alpha_t = \frac{c}{t+1}$ is the canonical schedule

What's next:

To convert these *cumulative* bounds into *convergence rates*, we need assumptions on f.

We start with the cleanest case: strong convexity.

Why Strong Convexity First?

Strong convexity provides the **cleanest convergence analysis** for SGD:

- Makes the gradient size $\|\nabla f(x_t)\|^2$ **directly related** to suboptimality $f(x_t) - f(x^*)$
- Allows us to obtain **last iterate guarantees**
- Reveals the **fundamental limit** imposed by noise

We'll see precisely how the geometry of strong convexity interacts with noise to let SGD achieve a $O(1/T)$ rate instead of GD's geometric rate

Strong Convexity: Definition and Consequences

Definition

$f : \mathbb{R}^d \to \mathbb{R}$ is μ-**strongly convex** if for all $x, y \in \mathbb{R}^d$:

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2$$

Consequences: Setting $y = x^*$ where $\nabla f(x^*) = 0$:

$$f(x) - f(x^*) \geq \frac{\mu}{2}\|x - x^*\|^2$$

Also, for any x:

$$\langle \nabla f(x), x - x^* \rangle \geq \mu\|x - x^*\|^2$$

Interpretation: Strong convexity means the function grows quadratically away from the optimum, so being far away automatically creates a large restoring force.

Standard Assumptions for SGD Analysis

We assume:

1. **Strong convexity:** f is μ-strongly convex
2. **Smoothness:** Each f_i is L-smooth (gradients are L-Lipschitz):

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|$$

   This implies $\|\nabla f_i(x)\| \leq L\|x - x^*\| + \|\nabla f_i(x^*)\|$

3. **Bounded variance:** The stochastic gradient has bounded variance:

$$\mathbb{E}[\|\nabla f_{i_t}(x) - \nabla f(x)\|^2] \leq \sigma^2 \quad \text{for all } x$$

Note: These are standard but non-trivial. Smoothness controls gradient growth; bounded variance ensures noise is manageable.

The Main Result: $O(1/T)$ Convergence

Theorem (SGD for Strongly Convex Functions)

Let f be μ-strongly convex and L-smooth, and assume unbiased stochastic gradients with bounded variance σ^2. Run SGD with step size[2]

$$\alpha_t = \min\left\{\frac{1}{L}, \frac{2}{\mu(t+1)}\right\}.$$

Then for all sufficiently large T,

$$\mathbb{E}[f(x_T) - f(x^*)] \leq \frac{C}{T} + \frac{2L\sigma^2}{\mu^2 T}.$$

Interpretation:

- Deterministic term: decay from initialization
- Stochastic term: variance-limited accuracy

[2] The cap ensures $\alpha_t \leq 1/L$, so we can use the descent inequality.

Comparing with Gradient Descent

Gradient descent (GD) on μ-strongly convex, L-smooth f:

$$f(x_t) - f(x^*) \leq \left(1 - \frac{\mu}{L}\right)^t [f(x_0) - f(x^*)]$$

- **Geometric** (exponential/linear) convergence
- Rate: $O(e^{-\mu t/L})$
- Reaches ϵ error in $O(\frac{L}{\mu} \log(1/\epsilon))$ iterations
- **Cost per iteration:** $O(nd)$

Stochastic gradient descent (SGD):

- **Sublinear** convergence: $O(1/T)$
- Reaches ϵ error in $O(1/\epsilon)$ iterations
- **Cost per iteration:** $O(d)$

Trade-off: SGD is n times faster per iteration but has worse iteration complexity.

Why the Slower Rate? The Role of Noise

Key difference: Noise prevents geometric convergence.

In GD: once close to x^*, gradient is small and steps shrink naturally.

In SGD: even near x^*, stochastic gradient has variance σ^2.

- This variance **lower bounds** the achievable error
- Cannot converge faster than $O(1/T)$ without additional structure

The $O(1/T)$ rate is optimal for first-order stochastic methods on strongly convex functions.

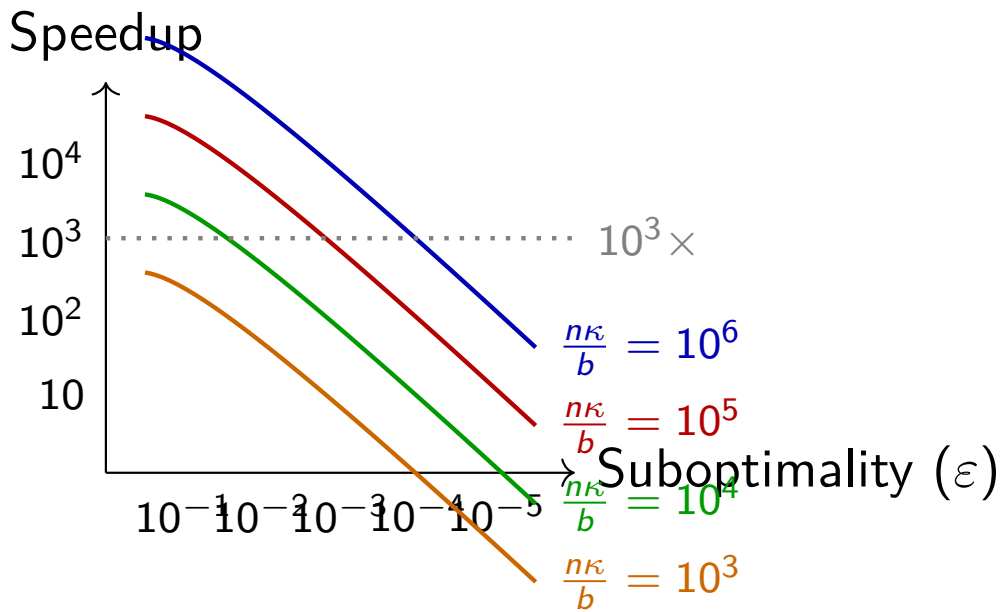
Interpretation: SGD trades iteration quality for iteration cost. For large n, this is favorable in total computational cost.

Overall speedup of SGD over GD

The speedup in the *total cost* of solving the problem to suboptimality ε (This is only valid for ε above the noise floor; we're also ignoring constant factors and descent constraints on the stepsizes):

$$\frac{nd \cdot T_{GD}(\varepsilon)}{bd \cdot T_{SGD}(\varepsilon)} = \frac{nd \cdot \kappa \log(\frac{1}{\varepsilon})}{bd \cdot \frac{1}{\varepsilon}} = \frac{n\kappa}{b}.$$

For strongly convex objectives:



Proof

Step 1: Use the L-smoothness quadratic upper bound at $x_{t+1} = x_t - \alpha_t \nabla f_{i_t}(x_t)$

$$f(x_{t+1}) \leq f(x_t) - \alpha_t \langle \nabla f(x_t), \nabla f_{i_t}(x_t) \rangle + \frac{L\alpha_t^2}{2} \|\nabla f_{i_t}(x_t)\|^2$$

Because of the unbiasedness and bounded variance of the gradient estimator,

$$\begin{aligned} \mathbb{E}[f(x_{t+1}) \mid x_t] &\leq f(x_t) - \alpha_t \|\nabla f(x_t)\|^2 + \frac{L\alpha_t^2}{2} \left(\|\nabla f(x_t)\|^2 + \sigma^2 \right) \\ &= f(x_t) - \alpha_t \left(1 - \frac{L\alpha_t}{2} \right) \|\nabla f(x_t)\|^2 + \frac{L\alpha_t^2}{2} \sigma^2 \end{aligned}$$

Step 2: Using the step-size condition $\alpha_t \leq \frac{1}{L}$ and subtracting $f(x^*)$ from both sides,

$$\mathbb{E}[f(x_{t+1}) \mid x_t] - f(x^*) \leq (f(x_t) - f(x^*)) - \frac{\alpha_t}{2} \|\nabla f(x_t)\|^2 + \frac{L\alpha_t^2}{2} \sigma^2$$

Proof (continued)

Step 4: Recall $\|\nabla f(x_t)\|^2 \geq 2\mu(f(x_t) - f(x^*))$ because of strong convexity, so

$$\mathbb{E}[f(x_{t+1}) \mid x_t] - f(x^*) \leq (1 - \mu\alpha_t)\,(f(x_t) - f(x^*)) + \frac{L\alpha_t^2}{2}\sigma^2$$

This means the objective gap goes down by a constant fraction each iteration when it's above the noise floor. Take the full expectation,

$$\mathbb{E}[f(x_{t+1}) - f(x^*)] \leq (1 - \mu\alpha_t)\,\mathbb{E}\,[f(x_t) - f(x^*)] + \frac{L\alpha_t^2}{2}\sigma^2$$

Proof (continued)

Step 5: Set $\Delta_t = f(x_t) - f(x^*)$ so the recursion becomes

$$\Delta_{t+1} \leq (1 - \mu\alpha_t)\,\Delta_t + \frac{L\alpha_t^2}{2}\sigma^2$$

Exactly solving this recursion is ugly... Suffice it to say, the stepsize cap ensures stability early on; once the iterates enter the regime where the $1/t$ decay in α_t takes over, we can show that

$$\Delta_t \leq \frac{C}{T} + \frac{2L\sigma^2}{\mu^2 T},$$

where C depends on L, μ and Δ_0.

Remarks on the Strongly Convex Result

Step-size choice:

- $\alpha_t = \frac{2}{\mu(t+1)}$ is **tuned** to strong convexity constant μ
- In practice, μ is often unknown; requires tuning of the stepsize schedule

Constants matter! The $L\sigma^2/\mu^2$ term affects practical performance significantly.

Conceptual Summary: Strongly Convex Case

What we learned:

- SGD achieves **sublinear** convergence, slower than GD's geometric rate
- The slowdown is **fundamental**: noise prevents faster convergence
- $O(1/T)$ is optimal for this setting (minimax lower bound)

What's next:

What happens when we *remove* strong convexity?

Convex Case: What Breaks and What Replaces It

Without strong convexity ($\mu = 0$):

- The contraction disappears,
- Gradients do not shrink automatically near x^*
- The last iterate cannot be certified as good

Replacement: bounded gradients + averaging (Jensen)

Assumptions (Convex Case Only):

$$\mathbb{E}[\nabla f_{i_t}(x_t) \mid x_t] = \nabla f(x_t)$$
$$\mathbb{E}\|\nabla f_{i_t}(x_t) - \nabla f(x_t)\|^2 \leq \sigma^2$$
$$\|\nabla f(x)\|^2 \leq G^2 \qquad \text{for all } x$$

- The bounded-gradient assumption replaces the stabilizing role of strong convexity
- Standard in convex SGD analysis

Controlling averaged iterates

Define the averaged iterate:

$$\bar{x}_T = \frac{\sum_{t=0}^{T-1} \alpha_t x_t}{\sum_{t=0}^{T-1} \alpha_t}$$

For constant step size $\alpha_t = \alpha$, this simplifies to:

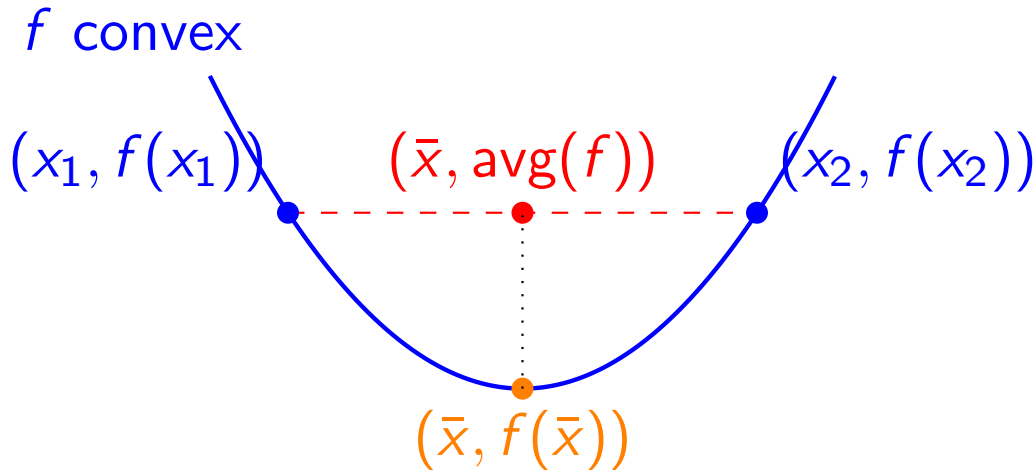
$$\bar{x}_T = \frac{1}{T} \sum_{t=0}^{T-1} x_t$$

Key tool: Jensen's inequality.
For convex f:

$$f(\bar{x}_T) = f\left(\frac{\sum_{t=0}^{T-1} \alpha_t x_t}{\sum_{t=0}^{T-1} \alpha_t} \right) \leq \frac{\sum_{t=0}^{T-1} \alpha_t f(x_t)}{\sum_{t=0}^{T-1} \alpha_t}$$

Intuition: Convexity ensures the function value at the average is at most the average of function values.

Jensen's Inequality: The Conceptual Reason Averaging Works



$$\bar{x} = \frac{x_1 + x_2}{2}$$

Jensen's inequality: $f(\bar{x}) \leq$ average of $f(x_i)$
The function value at the average point lies *below* the average of function values.

The Main Result: $O(1/\sqrt{T})$ Convergence

Theorem (SGD for Convex Functions)

Let f be convex, each f_i be L-smooth, and assume bounded variance σ^2 and bounded gradients,

$$\|\nabla f(x)\|^2 \leq G^2 \quad \text{for all } x.$$

Run SGD with constant step size $\alpha_t = \alpha$ for T iterations. Then:

$$\mathbb{E}[f(\bar{x}_T) - f(x^*)] \leq \frac{\|x_0 - x^*\|^2}{2\alpha T} + \frac{\alpha}{2}(G^2 + \sigma^2)$$

where $\bar{x}_T = \frac{1}{T}\sum_{t=0}^{T-1} x_t$.

With optimal step size choice,

$$\mathbb{E}[f(\bar{x}_T) - f(x^*)] \leq \frac{\sqrt{G^2 + \sigma^2}\|x_0 - x^*\|}{\sqrt{T}}$$

Rate: $O(1/\sqrt{T})$, slower than strongly convex case.

Practical Consideration: Diminishing vs Constant Step Sizes

Diminishing step sizes: $\alpha_t = \frac{c}{(t+1)^\beta}$

- Theoretically sound: satisfies Robbins–Monro
- $\sum \alpha_t = \infty$, $\sum \alpha_t^2 < \infty$
- Achieves $O(1/\sqrt{T})$ without knowing T in advance
- Slower in practice early on

Constant step size with known T: $\alpha = \frac{R}{\sigma\sqrt{T}}$

- Requires knowing T ahead of time
- Tighter constants in finite-time analysis
- More aggressive early steps
- Common in practice with scheduled decay

Hybrid approach: Start with constant α, decay when progress stalls.

Conceptual Summary: Convex Case

What we learned:

- Without strong convexity, cumulative bounds don't give iterate guarantees
- **Averaging is essential:** Jensen's inequality allows controlling the rate of convergence of averaged iterates.
- Convergence rate: $O(1/\sqrt{T})$, slower than strongly convex $O(1/T)$
- Step size must balance initial distance vs noise: $\alpha \sim 1/\sqrt{T}$
- Averaging is *the* mechanism for variance reduction across time

What's next:

Are these rates optimal? Can we do better?

Enter: lower bounds, batching, and Polyak averaging.

Lower Bounds: Are These Rates Optimal?

Fundamental question: Can any algorithm do better than SGD's rates?

Answer (minimax theory):

- For μ-strongly convex f: $O(1/T)$ is **optimal**
- For convex f: $O(1/\sqrt{T})$ is **optimal**

More precisely: there exist problem instances where *any* first-order stochastic method requires $\Omega(1/T)$ or $\Omega(1/\sqrt{T})$ iterations to achieve ϵ error.

Implication: SGD achieves the information-theoretically best possible rate.

We cannot improve the *rate*, but we can improve *constants*.

Two Axes for Improvement

Given that rates are optimal, we can still reduce the constant factors:

1. **Variance reduction in space: Batching**
   - Use multiple samples per iteration
   - Averages out noise within each step
   - Reduces σ^2 at cost of more computation per iteration

2. **Variance reduction in time: Polyak Averaging**
   - Different averaging scheme over iterates
   - Gives better constants without extra gradient computation
   - Particularly effective for strongly convex problems

Batching: Variance Reduction in Space

Mini-batch SGD: Sample a batch $\mathcal{B}_t \subset \{1, \ldots, n\}$ of size b:

$$x_{t+1} = x_t - \alpha_t \cdot \frac{1}{b} \sum_{i \in \mathcal{B}_t} \nabla f_i(x_t)$$

Variance reduction:

If samples are independent, variance of batch gradient:

$$\text{Var}\left[\frac{1}{b} \sum_{i \in \mathcal{B}_t} \nabla f_i(x_t)\right] = \frac{\sigma^2}{b}$$

Variance decreases linearly with batch size.

Trade-off:

- Each iteration costs $O(bd)$ instead of $O(d)$
- Reduces number of iterations by factor $\sim \sqrt{b}$ (from error bound)
- Total cost improvement: $\sqrt{b}$ iterations $\times$ b cost $= \sqrt{b}$ worse
- Parallelization makes batching efficient on GPUs

From Mini-batch to Shuffling SGD

From Mini-batch to Shuffling SGD

So far: Mini-batch SGD with i.i.d. sampling.

From Mini-batch to Shuffling SGD

So far: Mini-batch SGD with i.i.d. sampling.

In practice: **Shuffling SGD** is used ubiquitously.

From Mini-batch to Shuffling SGD

So far: Mini-batch SGD with i.i.d. sampling.

In practice: Shuffling SGD is used ubiquitously.

Algorithm 4: Shuffling SGD (Practical Version)

Input: Dataset $\mathcal{D} = \{z_1, \ldots, z_n\}$, batch size b, epochs E

for *epoch* $e = 1, \ldots, E$ **do**

  Randomly shuffle: $\mathcal{D}_e = \text{shuffle}(\mathcal{D})$;

  for *batch* $j = 1, \ldots, \lceil n/b \rceil$ **do**

    $B_j \leftarrow$ next b examples from $\mathcal{D}_e$ (or remaining if $< b$);

    Compute $\tilde{\nabla}_j = \frac{1}{|B_j|} \sum_{z \in B_j} \nabla \ell(z; \mathbf{w})$;

    Update: $\mathbf{w} \leftarrow \mathbf{w} - \alpha \tilde{\nabla}_j$;

  end

end

From Mini-batch to Shuffling SGD

So far: Mini-batch SGD with i.i.d. sampling.

In practice: Shuffling SGD is used ubiquitously.

Algorithm 5: Shuffling SGD (Practical Version)

Input: Dataset $\mathcal{D} = \{z_1, \ldots, z_n\}$, batch size b, epochs E

for *epoch* $e = 1, \ldots, E$ **do**

  Randomly shuffle: $\mathcal{D}_e = \text{shuffle}(\mathcal{D})$;

  for *batch* $j = 1, \ldots, \lceil n/b \rceil$ **do**

    $B_j \leftarrow$ next b examples from $\mathcal{D}_e$ (or remaining if $< b$);

    Compute $\tilde{\nabla}_j = \frac{1}{|B_j|} \sum_{z \in B_j} \nabla \ell(z; \mathbf{w})$;

    Update: $\mathbf{w} \leftarrow \mathbf{w} - \alpha \tilde{\nabla}_j$;

  end

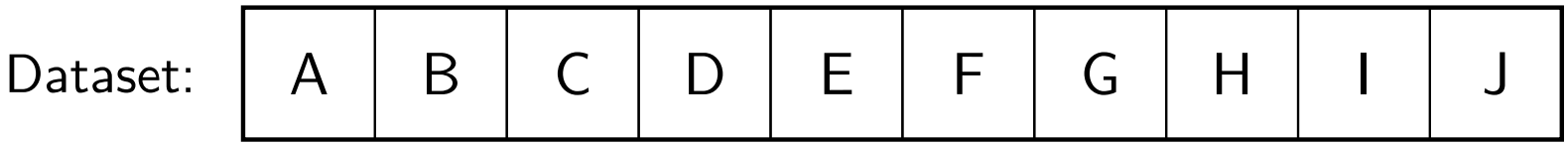
end

Each pass through the shuffled dataset is one **epoch**.

How Shuffling Works: Visualization

How Shuffling Works: Visualization

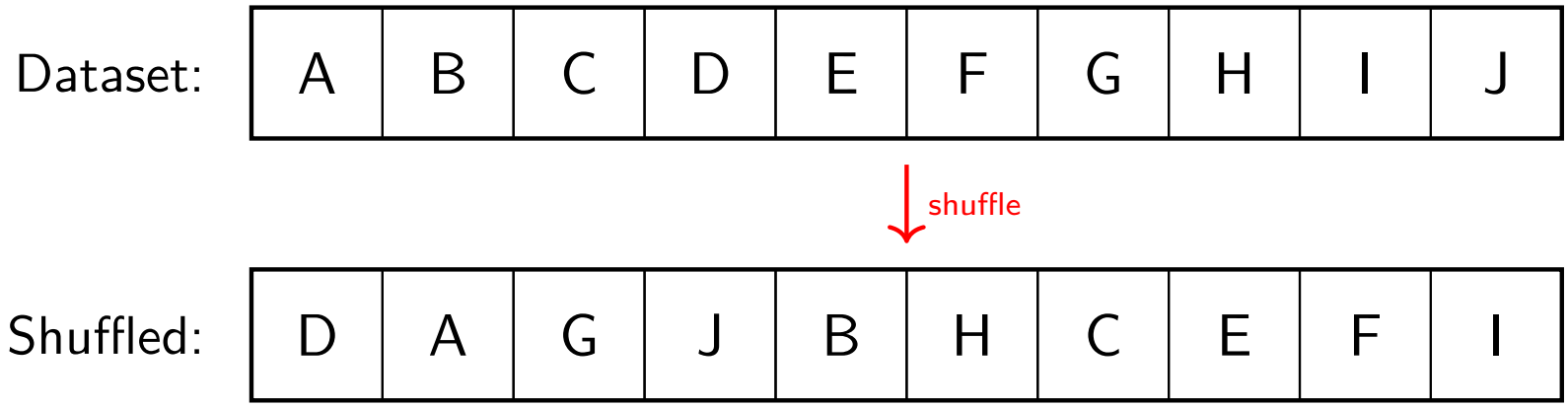
One epoch of shuffling SGD:



Process continues: Batch 3 has 3, Batch 4 has last example.
Then **reshuffle** for next epoch!

How Shuffling Works: Visualization

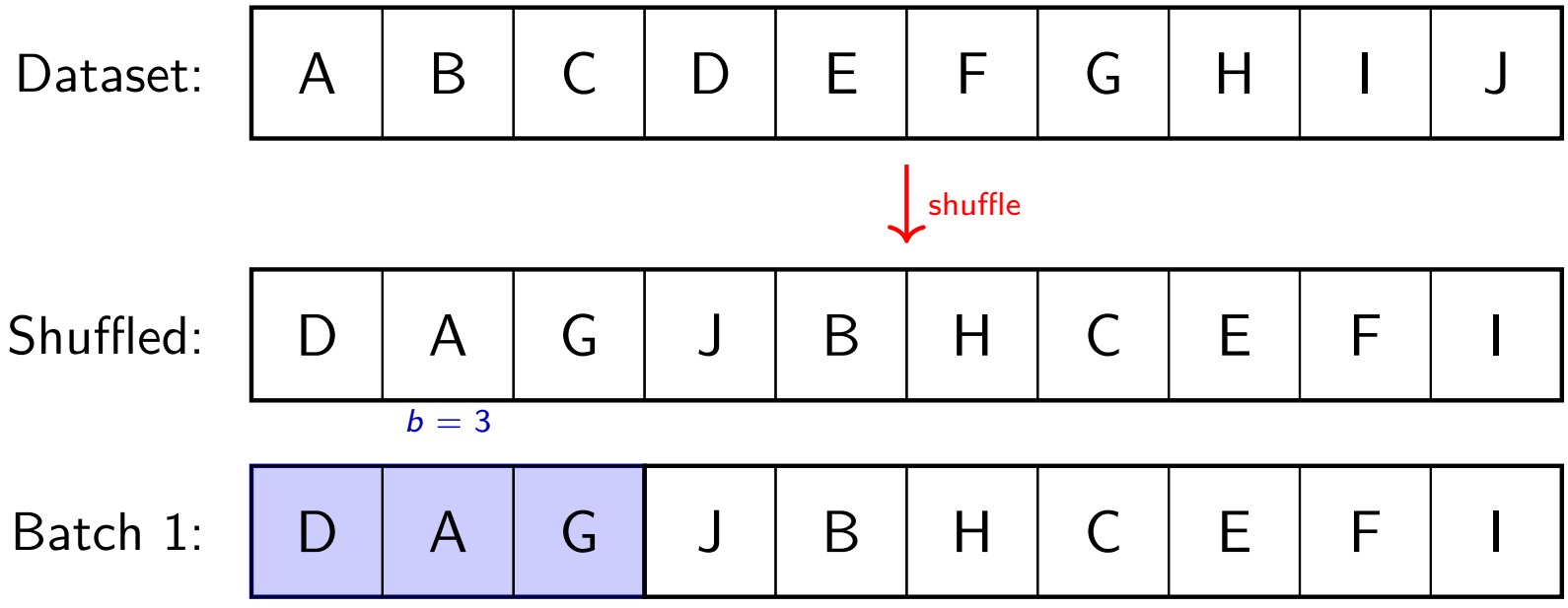
One epoch of shuffling SGD:



Process continues: Batch 3 has 3, Batch 4 has last example.
Then **reshuffle** for next epoch!

How Shuffling Works: Visualization

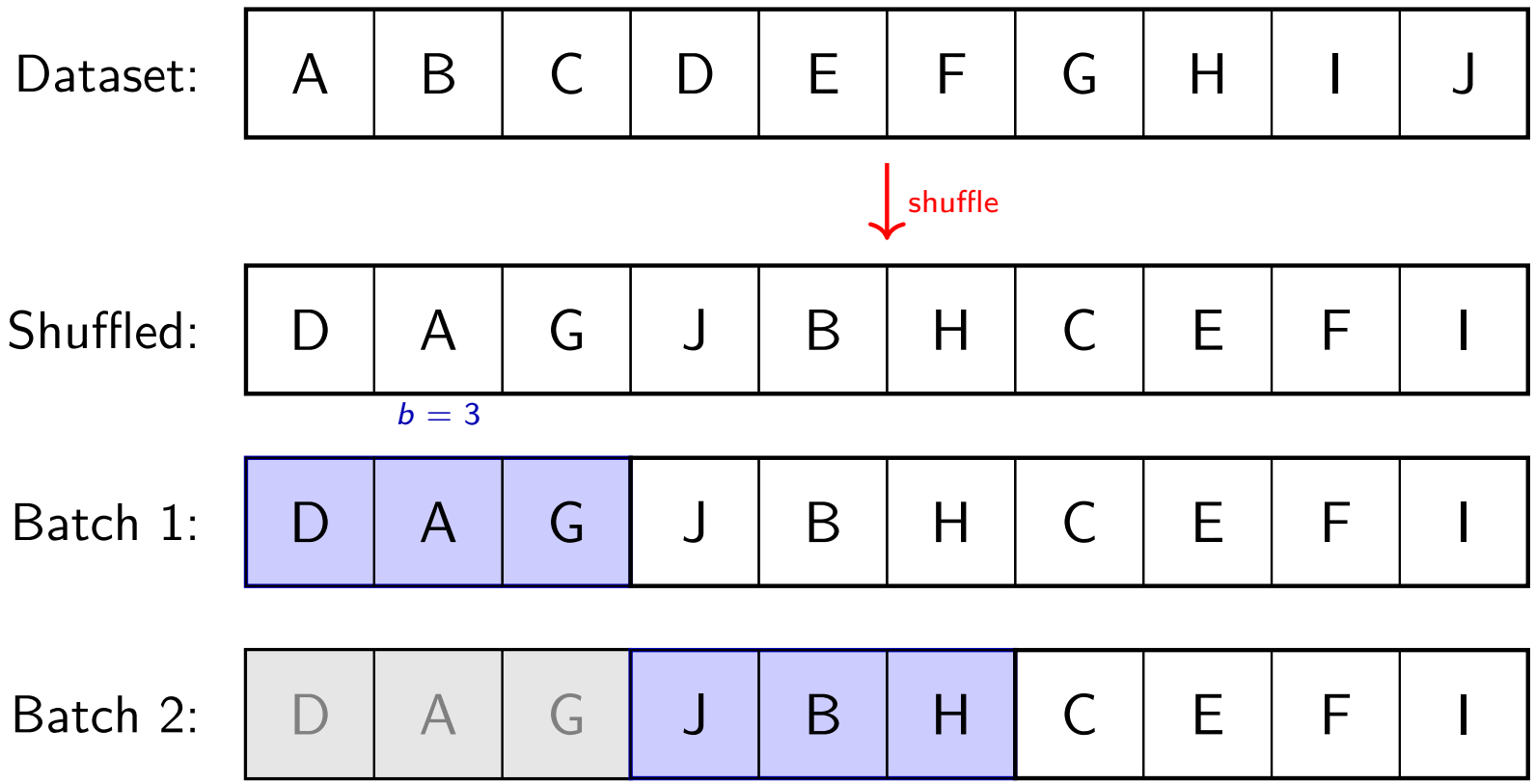
One epoch of shuffling SGD:



Process continues: Batch 3 has 3, Batch 4 has last example.
Then **reshuffle** for next epoch!

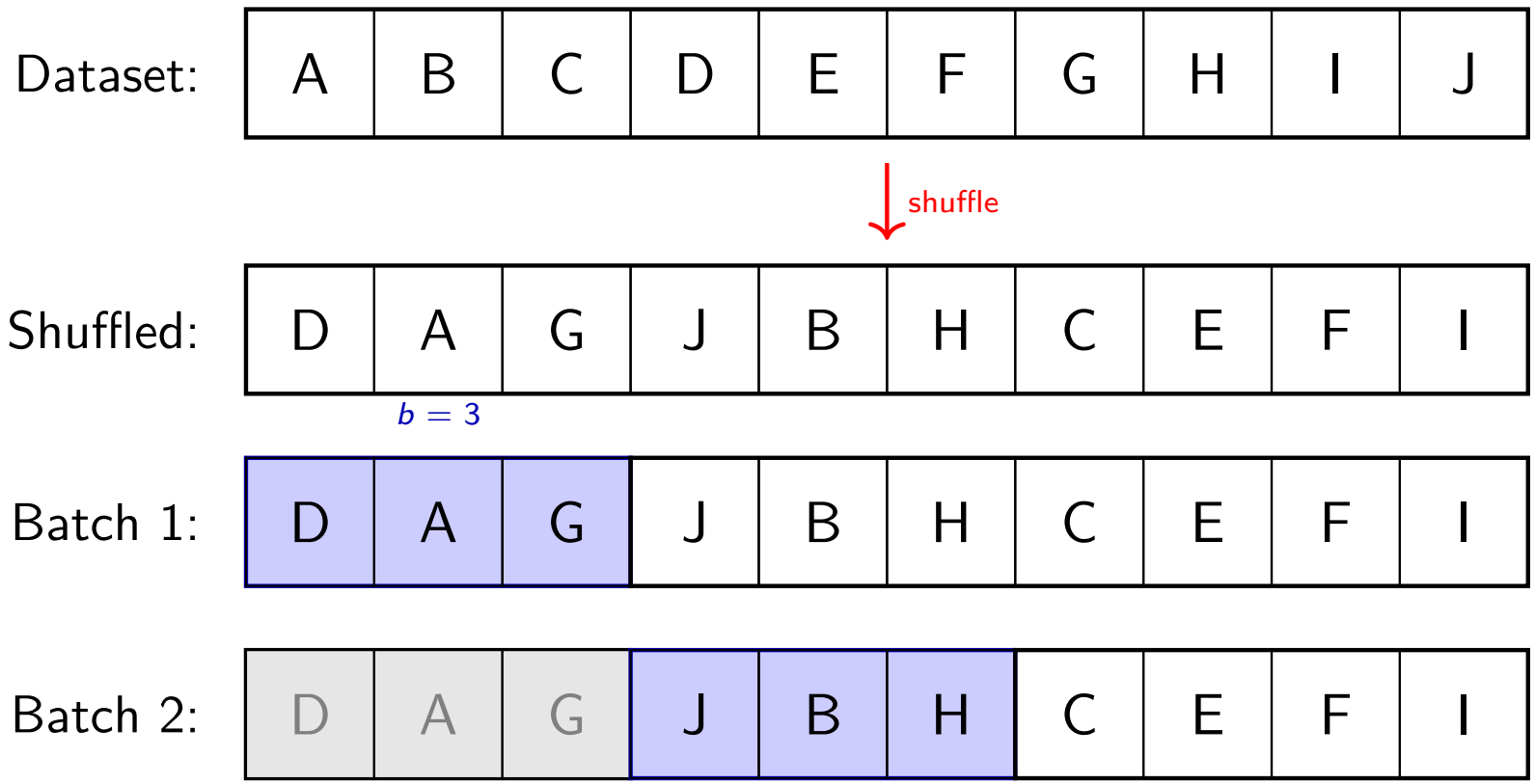
How Shuffling Works: Visualization

One epoch of shuffling SGD:



How Shuffling Works: Visualization

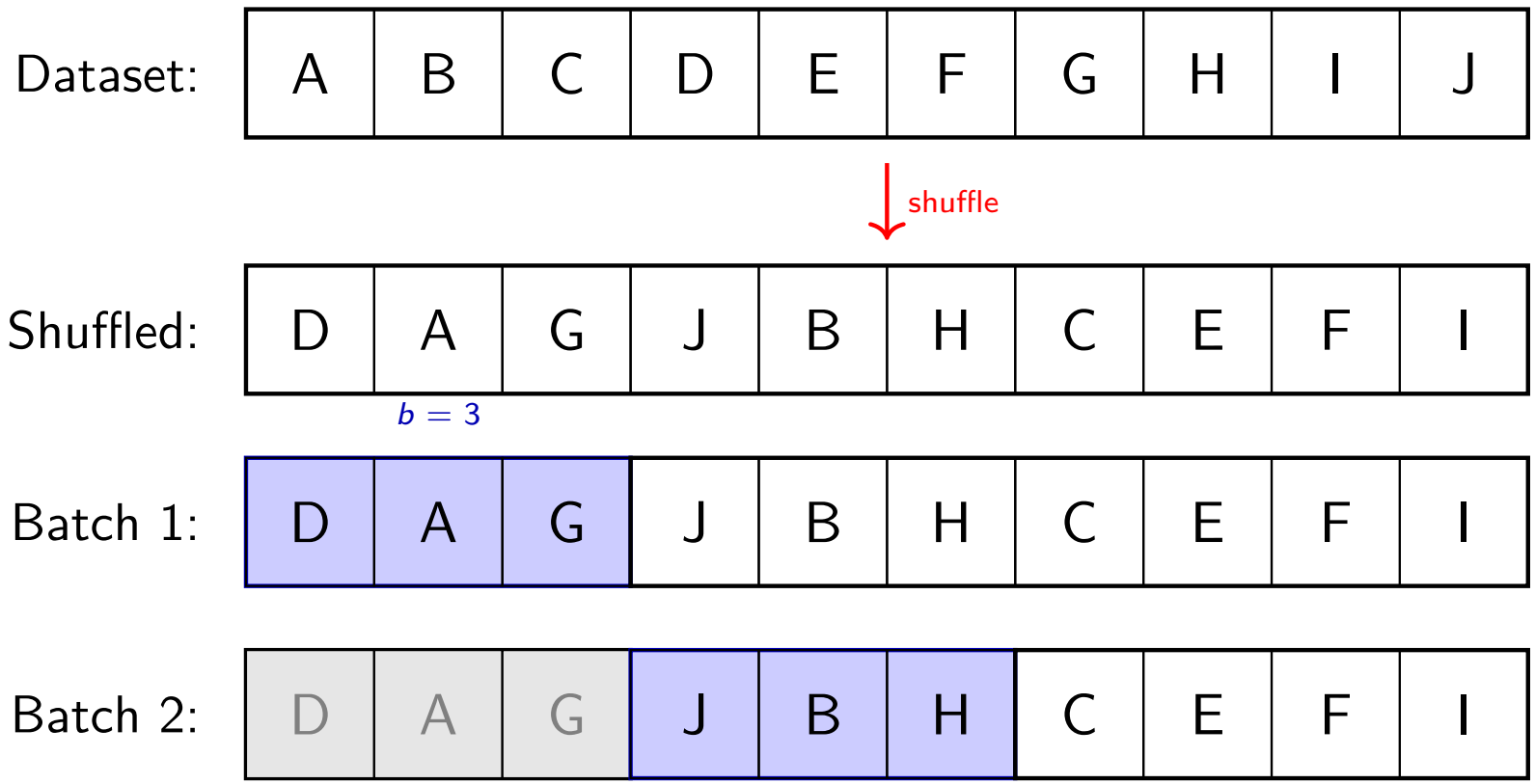
One epoch of shuffling SGD:



Process continues: Batch 3 has 3, Batch 4 has last example.

How Shuffling Works: Visualization

One epoch of shuffling SGD:



Process continues: Batch 3 has 3, Batch 4 has last example.
Then **reshuffle** for next epoch!

Why Shuffling?

Why Shuffling?

Advantages over i.i.d. sampling:

Why Shuffling?

Advantages over i.i.d. sampling:

1. **Lower variance**
   - Each example used exactly once per epoch
   - No repeated examples within epoch
   - Empirically: faster convergence

Why Shuffling?

Advantages over i.i.d. sampling:

1. **Lower variance**
   - Each example used exactly once per epoch
   - No repeated examples within epoch
   - Empirically: faster convergence
2. **More data-efficient**
   - I.i.d. sampling can miss or repeat examples
   - Shuffling ensures uniform coverage

Why Shuffling?

Advantages over i.i.d. sampling:

1. **Lower variance**
   - Each example used exactly once per epoch
   - No repeated examples within epoch
   - Empirically: faster convergence
2. **More data-efficient**
   - I.i.d. sampling can miss or repeat examples
   - Shuffling ensures uniform coverage

Disadvantage:

- Iterates within epoch are *not independent*
- Makes theoretical analysis more complex

Why Shuffling?

Advantages over i.i.d. sampling:

1. **Lower variance**
   - Each example used exactly once per epoch
   - No repeated examples within epoch
   - Empirically: faster convergence
2. **More data-efficient**
   - I.i.d. sampling can miss or repeat examples
   - Shuffling ensures uniform coverage

Disadvantage:

- Iterates within epoch are *not independent*
- Makes theoretical analysis more complex

Good news: Theory exists but is more abstruse (beyond this course). Same intuitions apply!

Polyak Averaging: Variance Reduction in Time

Standard averaging: $\bar{x}_T = \frac{1}{T}\sum_{t=0}^{T-1} x_t$
All iterates weighted equally.

Polyak (Ruppert) averaging: Use a diminishing step size and average iterates, typically weighting later iterates more heavily.
Example for step sizes $\alpha_t = \frac{c}{t+1}$:

$$\bar{x}_T^{\text{Polyak}} = \frac{2}{T(T+1)} \sum_{t=0}^{T-1} (t+1) x_t$$

Intuition: Later iterates are closer to optimum; weight them more.

Result: For strongly convex f, Polyak averaging achieves $O(1/T)$ with **better constants** than uniform averaging.

Why Polyak Averaging Works

Heuristic explanation:

- Early iterates: far from x^*, high suboptimality, low weight
- Late iterates: close to x^*, low suboptimality, high weight
- Effective averaging concentrates on good iterates

Formal analysis: More involved than uniform averaging; relies on tracking iterate variance over time.

Practical use:

- Common in stochastic approximation literature
- Most beneficial when running many iterations

Note: For convex (not strongly convex), standard averaging is often sufficient.

The Gap Between Theory and Practice

What theory explains well:

- Why step sizes must decay: noise accumulation
- Convergence rates: $O(1/T)$ vs $O(1/\sqrt{T})$
- Role of averaging: converts sums to iterate bounds
- Why batching reduces variance

What practice does differently:

- Uses constant step sizes + manual decay schedules
- Often doesn't average iterates (uses last iterate)
- Shuffles data instead of sampling with replacement
- Tunes step sizes heuristically, not based on μ or L
- Uses adaptive methods (Adam, RMSprop) that change step sizes per coordinate

Key point: Theory gives *principles*; practice tunes within those principles.

Constant Step Sizes and Learning Rate Schedules

Theory: Use $\alpha_t = \frac{c}{t+1}$ or $\alpha_t = \frac{c}{(t+1)^\beta}$ for provable convergence.

Practice: Start with constant α, decay manually.
Common schedules:

- **Step decay:** Reduce α by factor (e.g., $\times 0.1$) at fixed epochs
- **Exponential decay:** $\alpha_t = \alpha_0 \cdot \gamma^t$ for $\gamma < 1$
- **Cosine annealing** within the epochs

Why constant initially?

- Faster progress when far from optimum
- Theoretically-prescribed decay too slow early on
- Heuristically decay when validation loss plateaus

Principle preserved: Step size must eventually decrease to eliminate noise.

The Big Picture

Stochastic gradient descent:

- Trades iteration quality for iteration cost
- Enables large-scale optimization
- Requires careful step-size management
- Achieves optimal rates under noise

Design principles:

- **Step size:** Must decay to zero, but not too quickly
- **Averaging:** Smooths noise, converts cumulative to iterate bounds
- **Variance reduction:** Batching (space) and Polyak (time)

Modern extensions:

Control variates, adaptive step sizes, acceleration (momentum), variance reduction, distributed optimization, ...

All build on the same fundamental trade-off: **noise vs progress**.

Questions?

Next time: Nonsmooth "gradient descent"

Appendix Contents

1. Proof of Strongly Convex Convergence Theorem and supporting lemmata
2. Proof of Convex Convergence Theorem
3. Proof of Jensen's Inequality

Theorem: SGD for Strongly Convex Functions

Theorem

Let $f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$ be μ-strongly convex and L-smooth. Assume the stochastic gradients satisfy

$$\mathbb{E}[\nabla f_{i_t}(x_t) \mid x_t] = \nabla f(x_t), \quad \mathbb{E}[\|\nabla f_{i_t}(x_t) - \nabla f(x_t)\|^2 \mid x_t] \le \sigma^2$$

Run SGD with $x_{t+1} = x_t - \alpha_t \nabla f_{i_t}(x_t)$ and step size

$$\alpha_t = \frac{2}{\mu(t + t_0)}, \quad t_0 = \left\lceil \frac{2L}{\mu} \right\rceil$$

Then for all $T \ge 1$:

$$\mathbb{E}[f(x_T) - f(x^*)] \le \frac{2L\, t_0^2}{(T + t_0)^2}\|x_0 - x^*\|^2 + \frac{2L\sigma^2}{\mu^2 T}$$

Proof Strategy

We use a **function-value analysis** (descent lemma approach):

Step 1: Apply smoothness to get descent inequality with stochastic gradients

Step 2: Take expectations and bound using unbiasedness and variance

Step 3: Use PL inequality (from strong convexity) to convert gradient norm to function gap

Step 4: Choose step size $\alpha_t = \frac{2}{\mu(t+t_0)}$ to satisfy $\alpha_t \leq \frac{1}{L}$

Step 5: Solve the resulting recursion to obtain $O(1/T)$ rate

We begin with the supporting lemmas.

Variance Decomposition

Lemma (Variance Decomposition)

For the stochastic gradient $\nabla f_{i_t}(x_t)$, conditioned on x_t:

$$\begin{aligned}\mathbb{E}[\|\nabla f_{i_t}(x_t)\|^2 \mid x_t] &= \|\nabla f(x_t)\|^2 + \mathbb{E}[\|\nabla f_{i_t}(x_t) - \nabla f(x_t)\|^2 \mid x_t] \\ &\leq \|\nabla f(x_t)\|^2 + \sigma^2\end{aligned}$$

Proof:

Expand the squared norm:

$$\begin{aligned}\|\nabla f_{i_t}(x_t)\|^2 &= \|\nabla f(x_t) + (\nabla f_{i_t}(x_t) - \nabla f(x_t))\|^2 \\ &= \|\nabla f(x_t)\|^2 + 2\langle \nabla f(x_t), \nabla f_{i_t}(x_t) - \nabla f(x_t)\rangle \\ &\quad + \|\nabla f_{i_t}(x_t) - \nabla f(x_t)\|^2\end{aligned}$$

Taking conditional expectation, the cross term vanishes by unbiasedness. □

Lemma (Recursion Bound)

Suppose a nonnegative sequence $\{\Delta_t\}$ *satisfies*

$$\Delta_{t+1} \leq \left(1 - \frac{2}{s_t}\right)\Delta_t + \frac{c}{s_t^2}$$

where $s_t = t + t_0$ *with* $t_0 \geq 2$. *Then for all* $t \geq 0$*:*

$$\Delta_t \leq \frac{t_0(t_0-1)}{(t+t_0)(t+t_0-1)}\Delta_0 + \frac{c}{t+t_0-1}$$

In particular, $\Delta_t = O(1/t)$.

Proof Sketch: Define $A_t := s_t(s_t-1)\Delta_t$ and show that the original inequality implies $A_{t+1} \leq A_t + c$. Telescope to obtain $A_t \leq A_0 + ct$. Divide both sides by $s_t(s_t-1)$ and observe that

$$\frac{t}{s_t(s_t-1)} < \frac{1}{s_t - 1}.$$

□

Step 1: Descent Lemma

Apply the smoothness quadratic upper bound at $y = x_{t+1} = x_t - \alpha_t \nabla f_{i_t}(x_t)$:

$$\begin{aligned} f(x_{t+1}) &\leq f(x_t) + \langle \nabla f(x_t), x_{t+1} - x_t \rangle + \frac{L}{2}\|x_{t+1} - x_t\|^2 \\ &= f(x_t) + \langle \nabla f(x_t), -\alpha_t \nabla f_{i_t}(x_t) \rangle + \frac{L}{2}\|\alpha_t \nabla f_{i_t}(x_t)\|^2 \\ &= f(x_t) - \alpha_t \langle \nabla f(x_t), \nabla f_{i_t}(x_t) \rangle + \frac{L\alpha_t^2}{2}\|\nabla f_{i_t}(x_t)\|^2 \end{aligned}$$

Intuition: Smoothness controls how much the function can increase when we take a step. The middle term measures alignment between true and stochastic gradients; the last term is the penalty from step size.

Step 2: Taking Expectations

Take conditional expectation given x_t. By unbiasedness:

$$\mathbb{E}[\langle \nabla f(x_t), \nabla f_{i_t}(x_t) \rangle \mid x_t] = \langle \nabla f(x_t), \nabla f(x_t) \rangle = \|\nabla f(x_t)\|^2$$

By the variance decomposition lemma:

$$\mathbb{E}[\|\nabla f_{i_t}(x_t)\|^2 \mid x_t] \leq \|\nabla f(x_t)\|^2 + \sigma^2$$

Therefore:

$$\begin{aligned}
\mathbb{E}[f(x_{t+1}) \mid x_t] &\leq f(x_t) - \alpha_t \|\nabla f(x_t)\|^2 + \frac{L\alpha_t^2}{2}(\|\nabla f(x_t)\|^2 + \sigma^2) \\
&= f(x_t) - \alpha_t \left(1 - \frac{L\alpha_t}{2}\right) \|\nabla f(x_t)\|^2 + \frac{L\alpha_t^2}{2}\sigma^2
\end{aligned}$$

Step 3: Using the PL Inequality

Note that the step size was chosen so $\alpha_t \leq \frac{1}{L}$, so $1 - \frac{L\alpha_t}{2} \geq \frac{1}{2}$.
Then:

$$\mathbb{E}[f(x_{t+1}) \mid x_t] \leq f(x_t) - \frac{\alpha_t}{2}\|\nabla f(x_t)\|^2 + \frac{L\alpha_t^2}{2}\sigma^2$$

Subtract $f(x^*)$ from both sides and take full expectation. Define $\Delta_t = \mathbb{E}[f(x_t) - f(x^*)]$.
From the PL inequality, $\|\nabla f(x_t)\|^2 \geq 2\mu(f(x_t) - f(x^*))$.
Therefore:

$$\begin{aligned}
\Delta_{t+1} &\leq \Delta_t - \frac{\alpha_t}{2}\mathbb{E}[\|\nabla f(x_t)\|^2] + \frac{L\alpha_t^2}{2}\sigma^2 \\
&\leq \Delta_t - \frac{\alpha_t}{2} \cdot 2\mu\Delta_t + \frac{L\alpha_t^2}{2}\sigma^2 \\
&= (1 - \mu\alpha_t)\Delta_t + \frac{L\alpha_t^2}{2}\sigma^2
\end{aligned}$$

Step 4: Choosing the Step Size

We chose the step size so that $\alpha_t \leq \frac{1}{L}$ and $\alpha_t \to 0$ at a rate satisfying the Robbins-Monro conditions:

$$\alpha_t = \frac{2}{\mu(t + t_0)}, \quad t_0 = \left\lceil \frac{2L}{\mu} \right\rceil$$

The recursion becomes:

$$\begin{aligned} \Delta_{t+1} &\leq \left(1 - \mu \cdot \frac{2}{\mu(t + t_0)}\right) \Delta_t + \frac{L}{2} \cdot \frac{4}{\mu^2(t + t_0)^2} \sigma^2 \\ &= \left(1 - \frac{2}{t + t_0}\right) \Delta_t + \frac{2L\sigma^2}{\mu^2(t + t_0)^2} \end{aligned}$$

Step 5: Solving the Recursion

Apply the induction bound lemma with $s_t = t + t_0$ and $c = \frac{2L\sigma^2}{\mu^2}$:

$$\Delta_t \leq \frac{t_0(t_0-1)}{(t+t_0)(t+t_0-1)}\Delta_0 + \frac{2L\sigma^2}{\mu^2(t+t_0-1)}$$

For the initial gap, by smoothness:

$$\Delta_0 = f(x_0) - f(x^*) \leq \frac{L}{2}\|x_0 - x^*\|^2$$

Substituting and simplifying (using $t_0 \leq \frac{2L}{\mu} + 1$):

$$\mathbb{E}[f(x_T) - f(x^*)] \leq \frac{L\, t_0^2}{2(T+t_0-1)^2}\|x_0 - x^*\|^2 + \frac{2L\sigma^2}{\mu^2 T}$$

□

Summary

Use of strong convexity:

- The PL inequality $\|\nabla f(x)\|^2 \geq 2\mu(f(x) - f(x^*))$ is used to convert the descent lemma (which naturally produces a gradient norm term) into a contraction on the function gap itself

Use of smoothness:

- The descent lemma!
- To choose a step size ensuring $1 - \frac{L\alpha_t}{2} \geq \frac{1}{2}$, so descent isn't canceled by the quadratic error term

NB: this result is different than what is stated in the main slides, because it is less tedious to establish. Both give $O(1/T)$ convergence rates, with the same dependence on $\frac{L\sigma^2}{\mu^2}$.

Theorem: SGD for Convex Functions

Theorem

Let $f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$ be convex with each f_i being L-smooth. Assume the stochastic gradients satisfy

$$\mathbb{E}[\nabla f_{i_t}(x_t) \mid x_t] = \nabla f(x_t), \quad \mathbb{E}[\|\nabla f_{i_t}(x_t) - \nabla f(x_t)\|^2 \mid x_t] \leq \sigma^2$$

and $\|\nabla f(x)\|^2 \leq G^2$ for all x.
Run SGD with $x_{t+1} = x_t - \alpha \nabla f_{i_t}(x_t)$ for T iterations and define the averaged iterate

$$\bar{x}_T = \frac{1}{T}\sum_{t=0}^{T-1} x_t$$

Then for constant step size α:

$$\mathbb{E}[f(\bar{x}_T) - f(x^*)] \leq \frac{\|x_0 - x^*\|^2}{2\alpha T} + \frac{\alpha}{2}(G^2 + \sigma^2)$$

Proof Strategy

We use a **distance-based analysis with averaging**:

Step 1: Track progress via squared distance to optimum

Step 2: Apply convexity to relate function gap to gradient inner products

Step 3: Telescope the distance recursion over T iterations

Step 4: Apply Jensen's inequality to convert average of function values to function value at average

Step 5: Optimize step size to obtain $O(1/\sqrt{T})$ rate

We begin with the supporting lemmas.

Variance Decomposition

Lemma (Second Moment Bound)

For the stochastic gradient $\nabla f_{i_t}(x_t)$, conditioned on x_t:

$$\begin{aligned}\mathbb{E}[\|\nabla f_{i_t}(x_t)\|^2 \mid x_t] &= \|\nabla f(x_t)\|^2 + \mathbb{E}[\|\nabla f_{i_t}(x_t) - \nabla f(x_t)\|^2 \mid x_t] \\ &\leq \|\nabla f(x_t)\|^2 + \sigma^2 \\ &\leq G^2 + \sigma^2\end{aligned}$$

Proof: Identical variance decomposition as in the strongly convex case. The bounded gradient assumption $\|\nabla f(x_t)\|^2 \leq G^2$ then gives the final bound. □

Jensen's Inequality

Lemma (Jensen's Inequality for Convex Functions)

If f is convex, then for any points $x_0, \ldots, x_{T-1}$:

$$f\left(\frac{1}{T}\sum_{t=0}^{T-1} x_t\right) \leq \frac{1}{T}\sum_{t=0}^{T-1} f(x_t)$$

Why this matters: Without strong convexity, we lose contraction and cannot guarantee the last iterate x_T is good. Instead, we prove the *average function value* along the trajectory is small, then use Jensen to transfer this guarantee to the *averaged iterate* $\bar{x}_T$.

Step 1: Distance Recursion

Expand the squared distance after one SGD step:

$$\begin{aligned}\|x_{t+1} - x^*\|^2 &= \|x_t - \alpha \nabla f_{i_t}(x_t) - x^*\|^2 \\ &= \|x_t - x^*\|^2 - 2\alpha \langle \nabla f_{i_t}(x_t), x_t - x^* \rangle + \alpha^2 \|\nabla f_{i_t}(x_t)\|^2\end{aligned}$$

Take conditional expectation given x_t and use unbiasedness:

$$\begin{aligned}\mathbb{E}[\|x_{t+1} - x^*\|^2 \mid x_t] = \|x_t - x^*\|^2 - 2\alpha \langle \nabla f(x_t), x_t - x^* \rangle \\ + \alpha^2 \mathbb{E}[\|\nabla f_{i_t}(x_t)\|^2 \mid x_t]\end{aligned}$$

Intuition: We measure progress by how much closer we get to the optimum in expectation.

Step 2: Applying Convexity

Apply the bounded second moment from Lemma 8:

$$\mathbb{E}[\|x_{t+1} - x^*\|^2 \mid x_t] \leq \|x_t - x^*\|^2 - 2\alpha\langle \nabla f(x_t), x_t - x^* \rangle + \alpha^2(G^2 + \sigma^2)$$

Use the first-order condition for convex functions:

$$f(x_t) - f(x^*) \leq \langle \nabla f(x_t), x_t - x^* \rangle$$

Combining these:

$$2\alpha(f(x_t) - f(x^*)) \leq \|x_t - x^*\|^2 - \mathbb{E}[\|x_{t+1} - x^*\|^2 \mid x_t] + \alpha^2(G^2 + \sigma^2)$$

Intuition: Function progress is bounded by the drop in squared distance, plus noise.

Step 3: Telescoping

Take full expectation and sum from $t = 0$ to $T - 1$:

$$2\alpha \sum_{t=0}^{T-1} \mathbb{E}[f(x_t) - f(x^*)] \leq \sum_{t=0}^{T-1} \left(\mathbb{E}\|x_t - x^*\|^2 - \mathbb{E}\|x_{t+1} - x^*\|^2 \right) + T\alpha^2(G^2 + \sigma^2)$$

The sum telescopes:

$$2\alpha \sum_{t=0}^{T-1} \mathbb{E}[f(x_t) - f(x^*)] \leq \mathbb{E}\|x_0 - x^*\|^2 - \mathbb{E}\|x_T - x^*\|^2 + T\alpha^2(G^2 + \sigma^2)$$

$$\leq \|x_0 - x^*\|^2 + T\alpha^2(G^2 + \sigma^2)$$

Step 4: Applying Jensen's Inequality

Divide by $2\alpha T$ to get the average function value:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[f(x_t)-f(x^*)] \leq \frac{\|x_0-x^*\|^2}{2\alpha T}+\frac{\alpha}{2}(G^2+\sigma^2)$$

Apply Jensen's inequality (Lemma 10):

$$f(\bar{x}_T)=f\left(\frac{1}{T}\sum_{t=0}^{T-1}x_t\right)\leq\frac{1}{T}\sum_{t=0}^{T-1}f(x_t)$$

Taking expectations:

$$\mathbb{E}[f(\bar{x}_T)-f(x^*)]\leq\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[f(x_t)-f(x^*)]$$

Combining gives the main result.

Step 5: Choosing the Step Size

The bound has bias-variance structure:

$$\mathbb{E}[f(\bar{x}_T) - f(x^*)] \leq \underbrace{\frac{\|x_0 - x^*\|^2}{2\alpha T}}_{\text{bias term}} + \underbrace{\frac{\alpha}{2}(G^2 + \sigma^2)}_{\text{variance term}}$$

Minimize over α by setting derivative to zero:

$$\frac{d}{d\alpha}\left[\frac{\|x_0 - x^*\|^2}{2\alpha T} + \frac{\alpha}{2}(G^2 + \sigma^2)\right] = 0$$

This gives:

$$\alpha^* = \frac{\|x_0 - x^*\|}{\sqrt{G^2 + \sigma^2}\sqrt{T}}$$

Convergence Rate

Substituting α^* back into the bound:

$$\mathbb{E}[f(\bar{x}_T) - f(x^*)] \leq \frac{\|x_0 - x^*\|^2}{2T} \cdot \frac{\sqrt{G^2 + \sigma^2}\sqrt{T}}{\|x_0 - x^*\|} + \frac{1}{2} \cdot \frac{\|x_0 - x^*\|}{\sqrt{G^2 + \sigma^2}\sqrt{T}} \cdot (G^2 + \sigma^2)$$

Each term equals $\frac{1}{2}\frac{\sqrt{G^2+\sigma^2}\,\|x_0-x^*\|}{\sqrt{T}}$, so:

$$\mathbb{E}[f(\bar{x}_T) - f(x^*)] \leq \frac{\sqrt{G^2 + \sigma^2}\,\|x_0 - x^*\|}{\sqrt{T}} = O\left(\frac{1}{\sqrt{T}}\right)$$

Contrast: Strongly convex achieves $O(1/T)$; convex achieves $O(1/\sqrt{T})$.

Summary

Why no contraction without strong convexity:

- Without $\mu > 0$, there's no force pulling iterates toward x^*
- The last iterate x_T can wander and is not guaranteed to be good

How averaging saves us:

- Prove the *average function value* $\frac{1}{T}\sum_{t=0}^{T-1} f(x_t)$ is small via telescoping
- Use Jensen's inequality to transfer this to $f(\bar{x}_T)$, the function value at the averaged iterate

Role of bounded gradients:

- Prevents the noise term $\mathbb{E}[\|\nabla f_{i_t}(x_t)\|^2]$ from exploding
- Replaces the stabilizing effect of strong convexity

Jensen's Inequality

Lemma (Jensen's Inequality)

If $f : \mathbb{R}^d \to \mathbb{R}$ is convex and W is a random variable,

$$f(\mathbb{E}[W]) \leq \mathbb{E}[f(W)].$$

Equivalently for finite weighted averages:

$$f\left(\sum_{i=1}^{n} \lambda_i w_i\right) \leq \sum_{i=1}^{n} \lambda_i f(w_i).$$

Intuition: The function value at the average is at most the average of function values. This is the defining property of convex functions.

Proof of Jensen's Inequality (Differentiable Case)

If f is convex and differentiable, then for any w in the domain of f,

$$f(w) \geq f(\mathbb{E}W) + \langle \nabla f(\mathbb{E}W), w - \mathbb{E}W \rangle$$

Since expectation is a linear operator, it preserves inequalities. Therefore let $w \sim W$ and take the expectation of both sides of the inequality to conclude

$$\begin{aligned} \mathbb{E}f(W) &\geq f(\mathbb{E}W) + \mathbb{E}\langle \nabla f(\mathbb{E}W), w - \mathbb{E}W \rangle \\ &= f(\mathbb{E}W) + \langle \nabla f(\mathbb{E}W), \mathbb{E}W - \mathbb{E}W \rangle \\ &= f(\mathbb{E}W). \end{aligned}$$

□