CSCI 145 Problem 12: SGD Is a Monte Carlo Estimator

July 20, 2026

Submission Instructions

Please upload your work to Gradescope by **11:59pm on DATE**.

- Turn in a single PDF of *handwritten* solutions. Typed (including LaTeX) solutions will not be accepted.
- Problems are grouped into three-week **cycles**. Once per cycle, you will meet with the grader and present the solution to *one* problem from that cycle, selected at random when you arrive — so bring your work and be ready to explain any of them.

Problem 12: SGD Is a Monte Carlo Estimator

A lab training a large model doubles its batch size from 512 to 1024. Every gradient step now costs twice the arithmetic. What does the lab get back, and is there a batch size past which it gets back nothing at all?

Lecture proved that the minibatch gradient is unbiased. This problem measures its noise, and finds that every fact Unit 1 proved about sample means is already a fact about training.

Throughout we use the reading's notation. The loss is an average over n examples, the per-example gradient at the current weights $\mathbf{w}$ is $\mathbf{g}^{(i)} = \nabla_{\mathbf{w}} \mathcal{L}_i(\mathbf{w}) \in \mathbb{R}^d$, and the batch gradient over a batch S of size B is

$$\hat{\mathbf{g}}^{\text{batch}} = \frac{1}{B} \sum_{i \in S} \mathbf{g}^{(i)}.$$

Take the batch to be drawn uniformly *with* replacement, so the B per-example gradients in it are independent draws from the n available, each with mean $\nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w})$. Write $\mathbf{e}^{(i)} = \mathbf{g}^{(i)} - \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w})$ for how far one example's gradient sits from the full gradient, so that $\mathbb{E}[\mathbf{e}^{(i)}] = \mathbf{0}$ and $\mathbb{E}\big[\|\mathbf{e}^{(i)}\|_2^2\big] = \sigma^2$, the quantity the reading called the gradient noise.

Part A: The $1/B$ Rate

Expanding the squared norm of a sum of B terms produces B diagonal terms and $B(B-1)$ cross terms. Using independence to kill the cross terms, show that

$$\mathbb{E}\Big[\big\|\hat{\mathbf{g}}^{\text{batch}} - \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w})\big\|_2^2\Big] = \frac{\sigma^2}{B},$$

which is the $\text{Var}(\hat{\mu}_n) = \sigma^2/n$ of the Monte Carlo lecture with B in place of n. Then answer the lab's question: by what factor must B grow to halve the *typical* error $\sigma/\sqrt{B}$ of the estimate, and by what factor does the cost of one step grow?

Part B: When Bigger Batches Stop Helping

Real datasets contain near-duplicates: repeated user actions, augmented copies of one image, boilerplate scraped a thousand times. Duplicate examples ask for the same update, so their gradients no longer disagree independently. For this part only, replace Part A's independence with the equicorrelated model of Problem 1: for every $i \neq j$ in the batch,

$$\mathbb{E}\big[\langle \mathbf{e}^{(i)}, \mathbf{e}^{(j)} \rangle\big] = \rho\, \sigma^2, \qquad \rho \in (0, 1).$$

Redo Part A's expansion, this time keeping the $B(B-1)$ cross terms, to show

$$\mathbb{E}\Big[\big\|\hat{\mathbf{g}}^{\text{batch}} - \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w})\big\|_2^2\Big] = \frac{(1-\rho)\,\sigma^2}{B} + \rho\, \sigma^2,$$

which is exactly Problem 1's equicorrelated variance with B in place of n. Find the batch size at which the two terms are equal, evaluate it at $\rho = 0.1$, and say what happens as $B \to \infty$. Conclude in one sentence why a lab with a heavily duplicated dataset should not expect a batch of one million to beat a batch of one thousand.

Part C: Momentum Trades a Little Bias for Less Variance

Unrolled, momentum's velocity is $\mathbf{v}^{(t+1)} = \sum_{k=0}^{t} \beta^k \, \hat{\mathbf{g}}^{(t-k)}$, where $\hat{\mathbf{g}}^{(t-k)}$ is the batch gradient computed k steps ago. To compare its noise against a single batch gradient's fairly, normalize the weights to sum to one, as Adam's bias correction does:

$$\bar{\mathbf{v}}^{(t)} = (1-\beta) \sum_{k=0}^{t} \beta^k \, \hat{\mathbf{g}}^{(t-k)}.$$

Treat each $\hat{\mathbf{g}}^{(t-k)}$ as an independent estimate with error variance σ^2/B from Part A. Using $\sum_{k=0}^{\infty} \beta^{2k} = \frac{1}{1-\beta^2}$, show that for large t the error variance of $\bar{\mathbf{v}}^{(t)}$ approaches

$$(1-\beta)^2 \cdot \frac{\sigma^2}{B} \cdot \frac{1}{1-\beta^2} = \frac{1-\beta}{1+\beta} \cdot \frac{\sigma^2}{B}.$$

Evaluate the factor $\frac{1-\beta}{1+\beta}$ at the usual $\beta = 0.9$ and state the batch size that would have given the same variance without momentum. Then explain in one sentence why this reduction is not free: the gradients being averaged were computed at *older* weights, so $\bar{\mathbf{v}}^{(t)}$ is a biased estimate of the gradient at the current $\mathbf{w}^{(t)}$. Finally, state in one sentence the lesson of Parts A through C about optimization noise and estimator noise.

Part D*: The $\sqrt{\kappa}$ Rate

On the quadratic $\mathcal{L}(w) = \frac{\lambda}{2} w^2$, eliminating the velocity v from the reading's two momentum updates leaves the single recurrence $w^{(t+1)} = w^{(t)} - \alpha\lambda w^{(t)} + \beta\big(w^{(t)} - w^{(t-1)}\big)$ in w alone, whose characteristic equation is

$$r^2 - (1 + \beta - \alpha\lambda)\, r + \beta = 0.$$

The error after t steps decays like $|r|^t$ for the larger root r. Using that the product of the roots of $r^2 - Sr + P$ equals P, show that whenever the two roots are complex conjugates they satisfy $|r_1| = |r_2| = \sqrt{\beta}$. For curvatures λ ranging over $[\lambda_{\min}, \lambda_{\max}]$ with $\kappa = \lambda_{\max}/\lambda_{\min}$, the classical (Polyak) choice

$$\alpha = \frac{4}{(\sqrt{\lambda_{\max}} + \sqrt{\lambda_{\min}})^2}, \qquad \beta = \left(\frac{\sqrt{\lambda_{\max}} - \sqrt{\lambda_{\min}}}{\sqrt{\lambda_{\max}} + \sqrt{\lambda_{\min}}} \right)^2$$

makes the roots complex conjugate for every λ in that range (you need not verify this). Conclude that every direction contracts at the single rate $\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$, and compare it against plain gradient descent's $\frac{\kappa-1}{\kappa+1}$ from Problem 8 at $\kappa = 15$ (the demo's bowl) and $\kappa = 1000$ (the Optimization demo's badly scaled features), reporting how many steps each needs to shrink the error by a factor of 1000.

Where this goes next. Parts A through C are Unit 1 redeployed: an unbiased average, a variance that shrinks with more samples, a correlation floor no averaging escapes, and a variance reduction that costs a little bias. The same three ideas return when the course reaches reinforcement learning, where a policy gradient is a Monte Carlo estimate with exactly this variance profile and a value-function baseline is the variance reduction, this time with no bias at all.