[Fall 2025] ECE 5290/7290 and ORIE 5290
Distributed Optimization for Machine Learning and AI

Practice Problems

How to use this practice set effectively:

- Start by reviewing your lecture notes and homework solutions to identify weak areas.
- Attempt each problem in this set *without* referring to the solution first. Treat each as an exam question. I know it is tough (in fact, it is harder than the actual exam), so view this set as additional reading material.
- After solving, compare your reasoning with the provided solutions—focus not only on the final answer but also on the step-by-step logic and the role of assumptions.
- For computational questions, practice deriving results by hand; for conceptual or multiple-choice questions, justify why each incorrect option fails.
- Revisit problems involving eigenvalue geometry, stochastic gradients, or network mixing several days later—spaced repetition helps retain intuition.

Question 1: Federated Learning: Personalization vs. Global Model

A mobile keyboard app trains next-word prediction via federated learning. Different users type in different styles (non-IID). You can deploy either a single global model or a global model with lightweight on-device personalization (e.g., last layer fine-tuning).

Which statement is most accurate?

(a) Personalization helps most when data are IID and harms when data are non-IID.
(b) Personalization helps when data are non-IID by adapting to user-specific patterns with little extra communication.
(c) Personalization mainly reduces communication cost but usually worsens accuracy on non-IID data.
(d) Personalization is only useful if devices have identical compute and battery profiles.

Correct: (b). Non-IID data benefit from small local adaptations on top of a shared global model, with little extra bandwidth.

Question 2: Local SGD in a Bandwidth-Limited Setting

In federated training with *Local SGD*, each device does τ local updates between communications. You have a tight uplink budget.

Which statement best describes the trade-off as τ increases?

(a) Communication cost rises and client drift shrinks.

(b) Communication cost drops, but client drift grows, which hurts final accuracy under data heterogeneity.

(c) Both communication cost and drift grow linearly.

(d) Neither communication cost nor drift is affected by τ.

Correct: (b). Fewer syncs save bandwidth but allow local models to drift further apart on non-IID data.

Question 3: Gradient Compression vs. Convergence in Distributed Training

To save bandwidth, you apply unbiased stochastic quantization with variance parameter ω (larger ω = fewer bits). With a fixed wall-clock budget, when is *more* compression (larger ω) most advantageous?

(a) Early training, when gradients are large and faster iteration throughput dominates.

(b) Late training, when gradients are tiny and we aim for the lowest possible variance floor.

(c) Equally beneficial early and late.

(d) Never beneficial; unbiased compression always slows convergence.

Correct: (a). Early on, extra iterations from faster communication outweigh the added quantization variance; later you typically reduce compression to reach a lower error floor.

Question 4: Learning-Rate Schedules Under a Time Budget

You can run only 3 wall-clock hours of training. Which schedule typically gives better accuracy?

(a) Fixed learning rate throughout.

(b) Linear warm-up for a short initial phase, then step decay (reduce the LR by a constant factor at a few planned times) matched to the allotted time.

(c) Start with an extremely large learning rate and keep it constant.

(d) Decrease the learning rate by a random amount each epoch.

Correct: (b). A brief warm-up avoids early instability (especially with larger batches/initialization), and step decay lets you take larger steps early and smaller steps later to settle to a lower error—within the same time budget.

Question 5: Robustness of Gradient Descent to Noise Geometry

This question explores how the curvature of a quadratic function affects the steady-state error of Gradient Descent (GD) when gradients are noisy.

We consider

$$f(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^\top A\mathbf{x}, \qquad A \succ 0, \quad \widehat{\nabla} f(\mathbf{x}) = A\mathbf{x} + \boldsymbol{\varepsilon},$$

where $\mathbb{E}[\boldsymbol{\varepsilon}] = 0$ and $\mathrm{Cov}(\boldsymbol{\varepsilon}) = \sigma^2 A$. The GD update is

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta\,\widehat{\nabla} f(\mathbf{x}_t).$$

Linear algebra hint. Any symmetric positive definite matrix A admits an *eigendecomposition*

$$A = U\Lambda U^\top,$$

where U is orthonormal ($U^\top U = I$) and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$ collects the eigenvalues ($\lambda_i > 0$). In these coordinates, the dynamics in $\mathbf{x}$ decouple into d one-dimensional scalar "modes."

(a) Which of the following best explains the purpose of the change of coordinates $\mathbf{z}_t = U^\top \mathbf{x}_t$?

(a) It rescales the learning rate to accelerate convergence.
(b) It transforms the multi-dimensional update into independent one-dimensional recursions.
(c) It eliminates gradient noise entirely.
(d) It forces all eigenvalues to become equal.

Answer: (b) *Explanation:* In the eigenbasis of A, each coordinate evolves independently as a scalar recursion $z_{t+1}^{(i)} = (1 - \eta\lambda_i)z_t^{(i)} - \eta\varepsilon_t^{(i)}$, so the dynamics "decouple" across eigenvectors.

(b) **(6 points)** In each eigen-direction, the mean update follows $z_{t+1}^{(i)} = (1 - \eta\lambda_i)z_t^{(i)}$. What is the stability condition on the stepsize η so that GD converges in mean?

(a) $0 < \eta < 1/L$
(b) $0 < \eta < 1/\mu$
(c) $0 < \eta < 2/L$
(d) $0 < \eta < 2/\mu$

Answer: (c) *Explanation:* The recursion is stable when $|1 - \eta\lambda_i| < 1$ for all eigenvalues, giving the condition $0 < \eta < 2/\lambda_{\max} = 2/L$. If η exceeds this range, iterates diverge.

(c) **(5 points)** Suppose the gradient noise covariance aligns with A (i.e., $\mathrm{Cov}(\varepsilon) = \sigma^2 A$). Which statement best describes the geometry of the steady-state error covariance in $\mathbf{x}$-space?

(a) The error covariance is isotropic (same in all directions).
(b) The error ellipse is elongated along directions of large λ_i (high curvature).
(c) The error ellipse is elongated along directions of small λ_i (flat curvature).
(d) The noise geometry has no influence on the steady-state error.

Answer: (b) *Explanation:* Noise variance in each mode scales with curvature ($\propto \lambda_i$), so directions with higher λ_i exhibit stronger fluctuations, forming an elongated "error ellipse" along steep eigenvectors.

(d) **(5 points)** Two matrices have identical eigenvalues but different eigenvectors:

$$A_1 = \begin{pmatrix} 10 & 0 \\ 0 & 1 \end{pmatrix}, \qquad A_2 = \begin{pmatrix} 5.5 & 4.5 \\ 4.5 & 5.5 \end{pmatrix}.$$

If GD starts from $\mathbf{x}_0 = [1, 1]^\top$, which statement best describes the trajectories?

(a) A_1: Zig-zag along coordinate axes; A_2: smooth diagonal path.
(b) A_1: Circular path; A_2: chaotic path.
(c) A_1: Straight diagonal; A_2: divergent.
(d) Both produce identical trajectories.

Answer: (a)

Explanation: For $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top A\mathbf{x}$, the *level sets* $f(\mathbf{x}) = c$ are ellipses given by $\mathbf{x}^\top A\mathbf{x} = 2c$. Their principal axes are the eigenvectors of A, and the gradient $\nabla f(\mathbf{x}) = A\mathbf{x}$ is orthogonal to these ellipses—hence GD, which moves along $-\nabla f(\mathbf{x})$, always steps perpendicular to the current contour.

For $A_1 = \operatorname{diag}(10, 1)$, the eigenvectors coincide with the coordinate axes, so the ellipses are axis-aligned. The gradient components along x_1 and x_2 differ greatly: the steep x_1 direction dominates at first, pulling the iterate quickly toward the x_2 axis. Once near the axis, the x_2 component dominates and reverses direction, leading to alternating updates that appear as a *zig-zag trajectory.*

For A_2, the same eigenvalues produce ellipses of the same shape, but rotated by 45°. Its eigenvectors are $[1, 1]^\top/\sqrt{2}$ and $[1, -1]^\top/\sqrt{2}$, and the starting point $\mathbf{x}_0 = [1, 1]^\top$ lies exactly along one eigenvector. Therefore, each gradient step points along that same diagonal direction, producing a smooth, nearly straight path to the minimum. The two systems share the same condition number (and thus the same convergence rate) but exhibit very different trajectory geometries.

Question 6: GD with Model Mis-specification

Consider $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top A\mathbf{x} - \mathbf{b}^\top\mathbf{x}$, but the update uses $\tilde{A} = A + \Delta$, i.e.,

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta(\tilde{A}\mathbf{x}_t - \mathbf{b}). \qquad (\star)$$

Hint. ***Neumann series.*** *For small* $\|\Delta\|$, $(A + \Delta)^{-1} = A^{-1} - A^{-1}\Delta A^{-1} + O(\|\Delta\|^2)$.

(a) Express the fixed point $\tilde{\mathbf{x}}^\star$ of the perturbed iteration $(\star)$ and its bias $\tilde{\mathbf{x}}^\star - \mathbf{x}^\star$ relative to $\mathbf{x}^\star = A^{-1}\mathbf{b}$.

The iteration is linear and converges (for small η) to the fixed point of $\mathbf{x} = \mathbf{x} - \eta(\tilde{A}\mathbf{x} - \mathbf{b})$, i.e.

$$\tilde{A}\tilde{\mathbf{x}}^\star = \mathbf{b} \;\Rightarrow\; \tilde{\mathbf{x}}^\star = (A + \Delta)^{-1}\mathbf{b}.$$

Using the Neumann expansion for small $\|\Delta\|$: $(A + \Delta)^{-1} = A^{-1} - A^{-1}\Delta A^{-1} + O(\|\Delta\|^2)$. Hence bias:

$$\tilde{\mathbf{x}}^\star - \mathbf{x}^\star = \big[(A + \Delta)^{-1} - A^{-1}\big]\mathbf{b} \approx -A^{-1}\Delta A^{-1}\mathbf{b}.$$

(b) Give a sufficient condition on $(\eta, \|\Delta\|)$ for convergence of the iteration and bounded bias.

Convergence requires the spectral radius of $I - \eta\tilde{A}$ to be

$$\rho(I - \eta\tilde{A}) < 1$$

a sufficient condition is $0 < \eta < 2/\|\tilde{A}\|$. If $A \succ 0$ and $\|\Delta\| < \lambda_{\min}(A)$, then

$$\tilde{A} \succ 0 \text{ and } \|\tilde{A}\| \leq \|A\| + \|\Delta\|.$$

Thus choose

$$\eta < 2/(\|A\| + \|\Delta\|).$$

Bounded bias follows from $(A+\Delta)^{-1}$ existing and the first-order estimate above:

$$\|\tilde{\mathbf{x}}^\star - \mathbf{x}^\star\| \lesssim \|A^{-1}\|^2 \|\Delta\| \cdot \|\mathbf{b}\|.$$

(c) Discuss why ill-conditioning (e.g., $\lambda_{\min}(A)$ is very small) magnifies solution bias even for small $\|\Delta\|$.

Because $\|A^{-1}\| = 1/\lambda_{\min}(A)$ is large for ill-conditioned A, the first-order bias $\|A^{-1}\Delta A^{-1}\mathbf{b}\|$ scales like $\|A^{-1}\|^2\|\Delta\|\|\mathbf{b}\|$, hence small model error causes large solution bias.

Question 7: GD with Random Step Perturbations

This problem explores how small randomness in the stepsize affects Gradient Descent (GD) on a quadratic. Let $\eta_t = \eta(1+\epsilon_t)$ with i.i.d. ϵ_t satisfying $\mathbb{E}[\epsilon_t] = 0$ and $\mathrm{Var}(\epsilon_t) = \tau^2 \ll 1$. Consider $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top A\mathbf{x}$ with $A \succ 0$, and the update $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t A\mathbf{x}_t$.

(a) Derive how stepsize randomness changes the expected squared distance compared with deterministic GD. Express your result using the eigenvalues λ_i of A.

In the eigen-basis, $\mathbf{z}_t = U^\top \mathbf{x}_t$ and each scalar mode evolves as

$$z_{t+1}^{(i)} = \big(1 - \eta(1+\epsilon_t)\lambda_i\big)\, z_t^{(i)}.$$

Then

$$\mathbb{E}\big[(z_{t+1}^{(i)})^2\big] = \mathbb{E}\big[(1-\eta\lambda_i - \eta\lambda_i\epsilon_t)^2\big]\,(z_t^{(i)})^2 = \Big[(1-\eta\lambda_i)^2 + \eta^2\lambda_i^2\tau^2\Big]\,(z_t^{(i)})^2.$$

Versus deterministic GD (factor $(1-\eta\lambda_i)^2$), random steps add the term $\eta^2\lambda_i^2\tau^2$, so the expected squared distance decreases more slowly.

(b) Qualitatively, when does stepsize randomness have a significant effect, and when can it be ignored?

The extra term $\eta^2\lambda_i^2\tau^2$ is largest for steep directions (large λ_i) and for larger η, so randomness matters when η is near its usual upper range or the problem is ill-conditioned. If τ^2 is tiny and $\eta\lambda_i \ll 1$ for all i, the effect is negligible and behavior is close to fixed-step GD.

(c) For $A = \mathrm{diag}(1, 100)$ and $\eta = 0.01$, compare qualitatively the effect of $\tau = 0,\ 0.05,\ 0.1$ on progress along each coordinate.

Mode $\lambda_1 = 1$: $(1-\eta\lambda_1)^2 + \eta^2\lambda_1^2\tau^2 \approx (0.99)^2 + \eta^2\tau^2$, so the impact is small even when τ grows. Mode $\lambda_2 = 100$: the factor $(1-1)^2 + \eta^2 \cdot 100^2\tau^2 = \eta^2 10^4\tau^2$ grows quickly with τ, causing noisy, slower progress in the steep direction. Thus $\tau = 0$ behaves cleanly; $\tau = 0.05$ shows visible slow-down in the λ_2 mode; $\tau = 0.1$ makes that mode markedly noisier.

Question 8: Optimal Mini-batch under a Fixed Time Budget

In mini-batch stochastic gradient descent (SGD), each update uses a random batch of B samples. Larger batches produce more accurate gradient estimates (smaller variance) but are slower to compute. Smaller batches allow more updates per unit time but introduce higher gradient noise. This question examines how to choose B when total computation time is limited.

Setup. Assume that one mini-batch SGD step with batch size B takes

$$c_{\text{iter}}(B) = c_0 + c_1 B \;\text{ units of time,}$$

where c_0 is fixed overhead (e.g., setup and synchronization) and $c_1 B$ accounts for the cost of processing B samples. The total available training time is T, so the number of updates that can be performed is approximately

$$N \approx \frac{T}{c_{\text{iter}}(B)} = \frac{T}{c_0 + c_1 B}.$$

For sufficiently small learning rate η, the expected suboptimality after N steps can be approximated by

$$E(N,B) \approx \underbrace{\alpha(1-\gamma)^N}_{\text{optimization term}} + \underbrace{\beta\frac{\eta\sigma^2}{B}}_{\text{noise floor}},$$

where: - α and γ describe the convergence speed of the noiseless dynamics; - σ^2 measures gradient variance; - $\beta\frac{\eta\sigma^2}{B}$ represents the steady-state noise floor, which decreases as B increases.

This captures the trade-off: smaller $B \Rightarrow$ more iterations (larger N) but noisier updates; larger $B \Rightarrow$ fewer updates but lower variance.

(a) Formulate the optimization problem for selecting B that minimizes $E(N,B)$ given T, c_0, c_1, and σ^2.

Substitute $N = T/(c_0 + c_1 B)$ into $E(N,B)$ to get

$$\min_{B\in\mathbb{N}} \alpha(1-\gamma)^{T/(c_0+c_1 B)} + \beta\,\frac{\eta\sigma^2}{B}.$$

The first term decreases with smaller B (more steps), while the second decreases with larger B (less variance), illustrating the fundamental computation–variance trade-off.

(b) How does the optimal batch size $B^\star$ qualitatively vary with σ^2, total time T, and the ratio c_1/c_0?

$B^\star$ increases with higher σ^2 (noisier gradients make larger batches worthwhile), with longer total time T (more time allows the benefit of variance reduction to accumulate), and decreases when c_1/c_0 is large (high per-sample cost discourages large batches).

(c) Under what conditions is $B = 1$ (single-sample updates) close to optimal? Explain intuitively.

When the fixed overhead c_1 dominates the computation cost (so each update is relatively cheap) and the optimization term $\alpha(1-\gamma)^N$ dominates the noise floor (early in training or when the variance is modest), small batches are efficient. In this regime, $B = 1$ provides many quick, noisy updates that still yield fast initial progress.

Question 9: Importance Sampling for SGD

In standard stochastic gradient descent (SGD), at each iteration, we randomly select one data sample i (or a small batch) and compute its gradient $\nabla f_i(\mathbf{x})$ as an unbiased estimate of the full gradient

$$\nabla f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\mathbf{x}).$$

Uniform sampling ($p_i = 1/n$) treats all samples equally, but not all samples contribute equally to the variance of the stochastic gradient. If some gradients are much larger than others, it can be more efficient to sample them more often.

To formalize this, consider the *importance-sampling* variant of SGD: at each step, select index $i \in \{1, \dots, n\}$ with probability $p_i > 0$ (where $\sum_i p_i = 1$) and compute

$$g = \frac{1}{np_i}\nabla f_i(\mathbf{x}).$$

This weighting ensures the estimator remains unbiased even when sampling is non-uniform.

(a) Show that g is an unbiased estimator of the true gradient $\nabla f(\mathbf{x})$, and write an explicit expression for its variance.

Unbiasedness:

$$\mathbb{E}[g] = \sum_{i=1}^{n} p_i \frac{1}{np_i}\nabla f_i(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{x}) = \nabla f(\mathbf{x}).$$

Variance:

$$\operatorname{Var}(g) = \mathbb{E}\big[\|g - \nabla f(\mathbf{x})\|^2\big] = \sum_{i=1}^{n} p_i \left\| \frac{1}{np_i}\nabla f_i(\mathbf{x}) - \frac{1}{n}\sum_{j=1}^{n}\nabla f_j(\mathbf{x}) \right\|^2 .$$

(b) Explain how choosing probabilities p_i proportional to $\|\nabla f_i(\mathbf{x})\|$ can reduce variance compared with uniform sampling. (You may assume $\|\nabla f_i(\mathbf{x})\|$ are known.)

A convenient upper bound on the variance is

$$\sum_{i=1}^{n} \frac{1}{n^2 p_i}\|\nabla f_i(\mathbf{x})\|^2 - \|\nabla f(\mathbf{x})\|^2.$$

Minimizing the first term subject to $\sum_i p_i = 1$ gives (by Cauchy–Schwarz)

$$p_i^\star \propto \|\nabla f_i(\mathbf{x})\|.$$

This choice samples larger gradients more frequently, balancing their contribution to total variance. Compared with uniform $p_i = 1/n$, it never increases variance and typically decreases it significantly when gradient magnitudes vary widely.

(c) In large-scale or deep-learning settings, it is often expensive to compute $\|\nabla f_i(\mathbf{x})\|$ for all samples. Discuss practical ways to approximate or implement importance sampling in such systems.

Common strategies include:

- Using *stale gradient norms* from recent iterations instead of recomputing them each step;
- Using the per-example loss $f_i(\mathbf{x})$ as a proxy for $\|\nabla f_i(\mathbf{x})\|$;
- Performing *stratified sampling* by class or cluster to ensure balanced coverage;
- Approximating at coarser granularity, e.g., layer-wise or block-wise gradient norms in deep models;
- Periodically refreshing the sampling distribution p_i to reduce overhead.

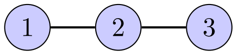
These heuristics preserve most of the variance-reduction benefits while keeping computation manageable.

Question 10: Optimal Mixing Step on a 3-Node Path

In distributed averaging or consensus algorithms, each node updates its value by mixing with its neighbors. Let W denote the *mixing matrix*, defined as

$$W = I - \alpha L,$$

where L is the graph Laplacian and $\alpha > 0$ is the mixing stepsize controlling how strongly nodes average with their neighbors.



P$_3$: 3-node path graph

The convergence rate of the consensus process

$$\mathbf{x}_{t+1} = W\mathbf{x}_t$$

depends on the spectral properties of W: the largest eigenvalue is 1 (corresponding to the consensus subspace), and the second-largest eigenvalue magnitude $\rho_2(W)$ determines the asymptotic convergence speed:

$$\|\mathbf{x}_t - \bar{\mathbf{x}}\| \approx \rho_2(W)^t.$$

Smaller $\rho_2(W)$ means faster averaging (shorter mixing time).

We consider a simple 3-node path graph (nodes 1–2–3) and aim to choose α to minimize $\rho_2(W)$.

(a) Write down the Laplacian L, compute the eigenvalues of the Laplacian L and then the eigenvalues of $W = I - \alpha L$ in terms of α.

For the 3-node path, the Laplacian is

$$L = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix},$$

which has eigenvalues $\{0, 1, 3\}$. Therefore $W = I - \alpha L$ has eigenvalues

$$\{\, 1,\ 1-\alpha,\ 1-3\alpha \,\}.$$

(b) Find the value $\alpha^\star$ that minimizes the second-largest eigenvalue magnitude $\max\{|1-\alpha|, |1-3\alpha|\}$, and compute the corresponding convergence rate.

To minimize the maximum magnitude, set $|1-\alpha| = |1-3\alpha|$, giving $\alpha = \frac{1}{2}$. Then the nontrivial eigenvalues are $1 - \frac{1}{2} = \frac{1}{2}$ and $1 - 3\frac{1}{2} = -\frac{1}{2}$. Hence the second-largest eigenvalue magnitude (SLEM) is $\rho_2(W) = \frac{1}{2}$. The asymptotic contraction per step is therefore 0.5.

Question 11: Heterogeneity and Slow Mixing in Decentralized SGD

Decentralized SGD replaces global averaging with local message passing among neighbors:

$$x_i^{k+1} = \sum_{j \in \mathcal{N}_i} W_{ij}\, x_j^k - \eta\, \nabla f_i(x_i^k),$$

where W is a mixing matrix satisfying $W\mathbf{1} = \mathbf{1}$, η is the stepsize, and each node i holds a local objective $f_i(x)$. When data are heterogeneous (local optima differ) and communication is infrequent (poorly connected network), the nodes can drift apart, leading to oscillation or even divergence if η is too large.

This problem asks you to construct a minimal counterexample illustrating this effect.

(a) Construct a minimal example (network topology, local objectives, and stepsize choice) where decentralized SGD fails to converge because of strong heterogeneity and weak mixing. Clearly state your setup and explain the mechanism behind divergence.

Setup: Consider two nodes (1 and 2) connected by a single edge with a small mixing weight $\alpha \ll 1$:

$$W = \begin{pmatrix} 1-\alpha & \alpha \\ \alpha & 1-\alpha \end{pmatrix}.$$

Let the local objectives be strongly conflicting:

$$f_1(x) = \tfrac{1}{2}(x-1)^2, \qquad f_2(x) = \tfrac{1}{2}(x+1)^2.$$

Then the local minimizers differ by 2. With a large stepsize η, each node moves aggressively toward its own minimizer before sufficient mixing occurs. Because W exchanges information slowly, the two nodes repeatedly "overshoot" in opposite directions, producing oscillations or even divergence in their disagreement $x_1 - x_2$.

(b) Suggest one or more strategies to restore convergence and justify why they work.

Possible remedies include:

- **Reduce the stepsize η:** smaller updates limit local overshoot, giving mixing more time to reconcile nodes before they drift apart.
- **Improve connectivity:** increase the spectral gap of W (e.g., larger α or adding links) to accelerate information averaging.
- **Periodic global averaging:** occasionally synchronize all nodes (hybrid decentralized/federated approach) to reset accumulated disagreement.

Each mitigation strengthens the coupling between nodes relative to their local drift, stabilizing the dynamics.

(c) In practice, how can one detect that such instability is occurring during training? What empirical signals would indicate a poorly tuned decentralized system?

Typical warning signs include:

- Rapidly growing disagreement $\sum_i \|x_i - \bar{x}\|^2$ between nodes;
- Oscillating or divergent local losses despite bounded gradients;
- Sensitivity: a small increase in η causes training to diverge;
- Network-wide quantities (e.g., averaged loss) fluctuate without settling.

Such behaviors indicate that the combination of large η and small spectral gap (slow mixing) makes the decentralized updates unstable.

Question 12: Choosing Local Steps τ under Time and Heterogeneity

In federated / local-SGD style training, each communication round performs τ *local* gradient steps before synchronizing across clients. Let the per-round wall time be

$$T(\tau) \;=\; \underbrace{\tau\, c_{\text{comp}}}_{\text{local compute}} \;+\; \underbrace{c_{\text{comm}}}_{\text{communication}} \;,$$

so the number of rounds in time budget T is approximately $N(\tau) \approx T/T(\tau)$. A stylized expected error model that separates *optimization progress*, *SGD noise*, and *client-drift* effects is

$$E(\tau) \;\approx\; \underbrace{\frac{A}{\eta\, N(\tau)}}_{\text{optimization term}} \;+\; \underbrace{B\,\eta\,\sigma^2}_{\text{SGD noise floor}} \;+\; \underbrace{\frac{C}{\tau}}_{\text{mini-batch variance reduction}} \;+\; \underbrace{D\,\tau}_{\text{heterogeneity (drift) penalty}} \;,$$

where $A, B, C, D > 0$ are problem-dependent constants and η is the stepsize (assumed fixed here). This captures the trade-off: increasing τ *reduces* communication and the C/τ term, but *increases* round time and the drift term $D\tau$.

(a) Give qualitative guidelines for the choice of τ in the two regimes: (i) $c_{\text{comm}} \gg c_{\text{comp}}$ (communication-dominated), and (ii) $c_{\text{comm}} \ll c_{\text{comp}}$ (compute-dominated).

(i) Communication-dominated $c_{\text{comm}} \gg c_{\text{comp}}$. Since each round is expensive to communicate, using a *larger* τ amortizes c_{comm} over more local work: $T(\tau) \approx \tau c_{\text{comp}} + c_{\text{comm}}$ decreases relative round-overhead as τ increases. Increasing τ also shrinks C/τ. However, too large τ inflates the drift penalty $D\tau$. *Guideline:* increase τ until the marginal drift cost $D\tau$ begins to outweigh the communication savings; keep τ where C/τ and $D\tau$ are balanced.

(ii) Compute-dominated $c_{\text{comm}} \ll c_{\text{comp}}$. Here $T(\tau) \approx \tau c_{\text{comp}}$, so increasing τ reduces the number of rounds nearly inversely and slows optimization progress. Moreover, large τ increases drift ($D\tau$) while providing diminishing gains in C/τ. *Guideline:* prefer *smaller* τ to keep rounds frequent and drift small; only increase τ if C/τ clearly dominates other terms.

(b) Treating τ as continuous, differentiate the smooth surrogate $E(\tau) \approx \dfrac{A}{\eta}\dfrac{T(\tau)}{T} + B\eta\sigma^2 + \dfrac{C}{\tau} + D\tau$ (using $N(\tau) = T/T(\tau)$) and give the interior optimality condition when it exists.

Using $T(\tau) = \tau c_{\text{comp}} + c_{\text{comm}}$, the surrogate becomes

$$E(\tau) \approx \frac{A}{\eta}\,\frac{\tau c_{\text{comp}} + c_{\text{comm}}}{T} \;+\; B\eta\sigma^2 \;+\; \frac{C}{\tau} \;+\; D\tau.$$

Differentiating (ignoring constants independent of τ):

$$\frac{dE}{d\tau} \;=\; \frac{A\,c_{\text{comp}}}{\eta T} \;-\; \frac{C}{\tau^2} \;+\; D.$$

An interior minimizer satisfies

$$\boxed{\frac{C}{\tau^2} \;=\; \frac{A\,c_{\text{comp}}}{\eta T} \;+\; D} \quad\Longrightarrow\quad \boxed{\tau^\star \;=\; \sqrt{\frac{C}{Ac_{\text{comp}}/(\eta T) \;+\; D}}}$$

provided the right-hand side is positive. This shows $\tau^\star$ *increases* when communication/optimization pressure is high (small $Ac_{\text{comp}}/(\eta T)$ and small D), and *decreases* as heterogeneity D grows.

(c) How does stronger client heterogeneity (non-IID data) affect the optimal τ?

Heterogeneity increases the drift penalty D (local models diverge more during unsynchronized steps). From $\tau^\star = \sqrt{C\,/\,\big(Ac_{\text{comp}}/(\eta T) + D\big)}$, increasing D *decreases* $\tau^\star$. *Interpretation:* with more heterogeneity, synchronize *more frequently* (smaller τ) to limit drift.

Question 13: (ECE 7290 only) Unbiased Quantization in SGD

Suppose gradients are compressed by an unbiased operator $\mathcal{Q}$ with

$$\mathbb{E}[\mathcal{Q}(g)] = g, \qquad \mathbb{E}\big[\|\mathcal{Q}(g) - g\|^2\big] \le \omega\,\|g\|^2,$$

for some $\omega \ge 0$ that increases as fewer bits are used (larger compression). Assume f is L-smooth and (optionally) μ-strongly convex; we use mini-batch size B so the baseline gradient noise is σ^2/B.

(a) Modify a canonical SGD recursion to include quantization and identify the additional variance term in a standard expected-suboptimality bound.

A single SGD step with compressed stochastic gradient g_t is

$$x_{t+1} \;=\; x_t \;-\; \eta\,\mathcal{Q}(g_t), \qquad \mathbb{E}[g_t \,|\, x_t] = \nabla f(x_t), \quad \mathbb{E}\|g_t - \nabla f(x_t)\|^2 \le \sigma^2/B.$$

Decompose $\mathcal{Q}(g_t) = g_t + \delta_t$ with $\mathbb{E}[\delta_t \,|\, g_t] = 0$ and $\mathbb{E}\|\delta_t\|^2 \le \omega\|g_t\|^2$. In the usual smooth/strongly-convex analysis, the steady-state (noise-floor) term is proportional to the total variance entering the update. Thus, compared to baseline variance σ^2/B, quantization adds an *extra* component

$$\boxed{\text{extra variance} \;\approx\; \omega\,\mathbb{E}\|g_t\|^2 \;\approx\; \omega\,\mathbb{E}\|\nabla f(x_t)\|^2}$$

(the last approximation uses that mini-batch noise vanishes near the optimum). A stylized bound for

μ-strongly convex f is

$$\mathbb{E}\big[f(x_t) - f^\star\big] \;\lesssim\; (1-\eta\mu)^t\, C_0 \;+\; \frac{\eta L}{2\mu}\left(\frac{\sigma^2}{B} \;+\; \omega\,\mathbb{E}\|\nabla f(x_t)\|^2\right),$$

showing the new variance contribution scales with ω and the local gradient magnitude.

(b) Discuss the trade-off between fewer bits (larger ω) and more iterations within a fixed time budget.

Fewer bits reduce communication time per iteration, allowing *more* iterations in a fixed wall time, improving the *optimization* term. However, larger ω increases the *variance floor* via $\omega\,\mathbb{E}\|\nabla f(x_t)\|^2$. *Trade-off:* when far from the optimum (large gradients), the added quantization noise is relatively small in *relative* terms and faster iterations can dominate; near the optimum (small gradients), the quantization noise can dominate and limit accuracy. Hence, aggressive compression is best early; later, one should reduce ω (more bits) to reach a lower error floor.

(c) Which regime tolerates quantization best?

(i) Near the optimum (small gradients),

(ii) Far from the optimum (large gradients).

Answer: (ii) Far from the optimum.

When gradients are large, the relative impact of the quantization error (bounded by $\omega\|g\|^2$) is smaller compared to the signal scale, and the speedup from reduced communication is most useful. Near the optimum, $\|\nabla f(x_t)\|$ is small but the added quantization term can dominate the variance floor, preventing further progress.

Question 14: (ECE 7290 only) Heavy-Ball Method and Its Stability Region

Momentum methods accelerate gradient descent by adding an extra term that reuses previous steps to gain inertia. For a quadratic objective

$$f(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^\top A\mathbf{x}, \quad A \succ 0, \quad \text{spectrum } [\mu, L],$$

consider the *heavy-ball* update

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta\, A\mathbf{x}_t + \beta(\mathbf{x}_t - \mathbf{x}_{t-1}),$$

where $\eta > 0$ is the stepsize and $\beta \in [0,1)$ is the momentum parameter.

Intuitively, the term $\beta(\mathbf{x}_t - \mathbf{x}_{t-1})$ pushes the iterate further along its previous motion direction. This can speed up convergence if tuned well, but can also cause oscillation or divergence if η or β are too large. The goal of this problem is to analyze when the iterates $\mathbf{x}_t$ converge for quadratic functions—i.e., to find the *stability region* in the (η, β) plane.

(a) Because A is symmetric positive definite, it can be diagonalized as $A = U\Lambda U^\top$. Explain why this transformation allows us to study the update one coordinate (eigen-direction) at a time, and write the resulting 1D recursion.

In the eigenbasis of A, $\mathbf{z}_t = U^\top \mathbf{x}_t$ evolves independently along each eigen-direction. For eigenvalue λ_i, the scalar variable $z_t^{(i)}$ satisfies

$$z_{t+1}^{(i)} = (1 + \beta - \eta\lambda_i)\, z_t^{(i)} - \beta\, z_{t-1}^{(i)}.$$

Thus, instead of analyzing a d-dimensional system, we can study d separate 1D recursions.

(b) The stability of this recursion depends on how errors evolve over time. The **characteristic polynomial** captures this evolution by describing how z_t depends on its past values: for a linear recurrence $z_{t+1} = az_t + bz_{t-1}$, the characteristic polynomial is $r^2 - ar - b = 0$. Its roots r_1, r_2 describe how errors decay or oscillate (stable if $|r_1|, |r_2| < 1$).

Write the characteristic polynomial for the heavy-ball update above and explain qualitatively what determines stability.

Substituting $a = 1 + \beta - \eta\lambda_i$ and $b = -\beta$ gives

$$r^2 - (1 + \beta - \eta\lambda_i)r + \beta = 0.$$

The two roots $r_{1,2}$ describe how the error along that eigen-direction evolves. The method is stable if both $|r_{1,2}| < 1$ (errors shrink each step). Large η or β make $|r|$ approach or exceed 1, causing oscillation or divergence.

(c) Using your scalar recursion, determine roughly how the stability region depends on η, β, and the largest eigenvalue L of A. In particular, describe the range of η that keeps the updates stable.

For a single eigenvalue λ, the heavy-ball update

$$x_{t+1} = (1 + \beta - \eta\lambda)x_t - \beta x_{t-1}$$

is stable if both characteristic roots of

$$r^2 - (1 + \beta - \eta\lambda)r + \beta = 0$$

lie inside the unit circle ($|r| < 1$). Applying a standard discrete-time stability test gives

$$|\beta| < 1, \quad 1 - (1 + \beta - \eta\lambda) + \beta > 0, \quad 1 + (1 + \beta - \eta\lambda) + \beta > 0.$$

Simplifying these inequalities yields

$$0 < \eta\lambda < 2(1 + \beta).$$

To ensure convergence for all eigenvalues $\lambda \in [\mu, L]$, this must hold for the largest one, giving the rough stability rule:

$$\boxed{0 < \eta < \frac{2(1 + \beta)}{L}, \quad 0 < \beta < 1.}$$

When $\beta = 0$, this reduces to the standard condition for gradient descent $\eta < 2/L$.

Question 15: (ECE 7290 only) Heavy-Ball Method under Gradient Noise

In stochastic optimization, gradient noise prevents exact convergence even for convex quadratic problems. Momentum can speed up optimization in noiseless settings but may also amplify stochastic fluctuations. This question compares how plain GD and the heavy-ball momentum method behave under additive noise.

Setup. Consider the 1D quadratic model

$$f(x) = \tfrac{1}{2}\lambda x^2,$$

and assume each gradient evaluation is corrupted by additive noise:

$$g_t = \lambda x_t + \varepsilon_t, \quad \mathbb{E}[\varepsilon_t] = 0, \quad \mathrm{Var}(\varepsilon_t) = \sigma^2.$$

The GD and momentum updates are

$$\text{GD: } x_{t+1} = x_t - \eta g_t, \qquad \text{Momentum: } x_{t+1} = x_t - \eta g_t + \beta(x_t - x_{t-1}),$$

where η is the stepsize and $\beta \in [0, 1)$ is the momentum parameter.

We are interested in how noise accumulates in the steady state—that is, the long-run expected value of $\mathbb{E}[x_t^2]$ once transient effects vanish.

(a) For the scalar model above, derive how momentum changes the mapping from gradient noise ε_t to parameter fluctuations x_t. Express qualitatively (or quantitatively, if possible) how the steady-state variance scales with β.

Starting from

$$x_{t+1} = (1 + \beta - \eta\lambda)x_t - \beta x_{t-1} - \eta\varepsilon_t, \qquad \mathbb{E}[\varepsilon_t] = 0,\ \mathrm{Var}(\varepsilon_t) = \sigma^2,$$

each update depends on both x_t and x_{t-1}.

To analyze it, define the 2D state vector

$$s_t = \begin{bmatrix} x_t \\ x_{t-1} \end{bmatrix}$$

the second-order recursion (because it involves two past iterates) can be written as

$$s_{t+1} = As_t + B\varepsilon_t, \quad A = \begin{bmatrix} 1 + \beta - \eta\lambda & -\beta \\ 1 & 0 \end{bmatrix}, \ B = \begin{bmatrix} -\eta \\ 0 \end{bmatrix}.$$

Let $\Sigma = \lim_{t\to\infty} \mathbb{E}[s_t s_t^\top]$ denote this steady-state covariance. From the recursion $s_{t+1} = As_t + B\varepsilon_t$, taking the outer product and expectation gives

$$\mathbb{E}[s_{t+1}s_{t+1}^\top] = \mathbb{E}\big[(As_t + B\varepsilon_t)(As_t + B\varepsilon_t)^\top\big].$$

Because the gradient noise ε_t is zero-mean and independent of s_t, the cross terms vanish, leaving

$$\mathbb{E}[s_{t+1}s_{t+1}^\top] = A\,\mathbb{E}[s_t s_t^\top]\,A^\top + \mathbb{E}[\varepsilon_t^2]\,BB^\top.$$

At steady state, $\Sigma = \lim_{t\to\infty} \mathbb{E}[s_t s_t^\top]$ satisfies $\Sigma = A\Sigma A^\top + \sigma^2 BB^\top$. Solving yields

$$\boxed{\mathrm{Var}(x_t) = \frac{\eta\sigma^2(1+\beta)}{\lambda(1-\beta)(2+2\beta-\eta\lambda)}}.$$

When the learning rate η is small, the term $\eta\lambda$ in the denominator of the exact formula

$$\mathrm{Var}(x_t) = \frac{\eta\sigma^2(1+\beta)}{\lambda(1-\beta)(2+2\beta-\eta\lambda)}$$

is much smaller than the constants $2+2\beta$. Neglecting it (a standard first-order approximation) simplifies the expression to

$$\mathrm{Var}(x_t) \approx \frac{\eta\sigma^2}{2\lambda(1-\beta^2)}.$$

This approximation reveals two important behaviors:

- The variance grows *linearly* with the learning rate η and the noise level σ^2, as in ordinary stochastic gradient descent.
- The extra factor $\frac{1}{1-\beta^2}$ shows how momentum amplifies noise: as β increases toward 1, this factor becomes large, meaning the iterates fluctuate more due to accumulated noise.

For example, if $\beta = 0.9$, then $\frac{1}{1-\beta^2} \approx 5.3$, so the steady-state variance is over five times larger than that of plain gradient descent ($\beta = 0$). In short, higher momentum accelerates convergence in noiseless problems but can substantially increase variability when gradients are noisy.

(b) Explain qualitatively when momentum improves and when it degrades performance.

Momentum helps when deterministic convergence speed is the bottleneck—i.e., in ill-conditioned, low-noise problems where curvature varies strongly across directions. It accelerates slow modes without significantly increasing variance. However, in high-noise regimes the $\frac{1}{(1-\beta)^2}$ amplification dominates, causing noisy oscillations and higher steady-state error. Thus momentum trades faster transient decay for greater steady-state variance.

(c) The impact of noise on the optimization dynamics depends on the *noise-to-signal ratio*, roughly measured by $\sigma^2/(\lambda^2 x_t^2)$ or by the steady-state noise level $\eta\sigma^2/\lambda$. Use this idea to reason about when momentum is helpful or harmful in the following cases:

(i) Ill-conditioned, low-noise problems (large condition number, small σ^2);

(ii) Well-conditioned, high-noise problems (small condition number, large σ^2);

(iii) Ill-conditioned, moderate-noise problems.

Option (iii) typically offers the best trade-off, while (i) also benefits. In (ii), noise dominates, so momentum's amplification is harmful. In (iii), momentum accelerates slow curvature directions (helping ill-conditioned problems) while noise remains moderate enough that the amplification factor $\frac{1}{(1-\beta)^2}$ does not overwhelm the gains.

Question 16: (ECE 7290 only) Noisy Consensus Steady State

Consider the linear consensus iteration with additive noise

$$\mathbf{x}^{k+1} = W\mathbf{x}^k + \boldsymbol{\xi}^k, \qquad \mathbb{E}[\boldsymbol{\xi}^k] = \mathbf{0}, \quad \mathrm{Cov}(\boldsymbol{\xi}^k) = \sigma^2 I,$$

where $W \in \mathbb{R}^{n\times n}$ is a symmetric, doubly-stochastic mixing matrix: $W\mathbf{1} = \mathbf{1}$ and $W = W^\top$. Let $J := \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ be the projector onto the consensus subspace and $P := I - J$ the projector onto the *disagreement* (mean-zero) subspace. Write the average $\bar{x}^k := \frac{1}{n}\mathbf{1}^\top\mathbf{x}^k$ and the disagreement vector $\mathbf{y}^k := P\mathbf{x}^k$.

Remark. Because $\lambda_1(W) = 1$, the network average follows a random walk under additive noise; a *steady state* exists only for the disagreement part $\mathbf{y}^k$.

(a) Show that the network average is preserved in *expectation* but the disagreement exhibits a nonzero variance floor. Derive the recursion for the disagreement covariance $\Sigma^k := \mathbb{E}[\mathbf{y}^k(\mathbf{y}^k)^\top]$ and the equation that the covariance $\Sigma := \lim_{k\to\infty}\mathbb{E}[\mathbf{y}^k(\mathbf{y}^k)^\top]$ should satisfy at the steady state.

Taking expectations,

$$\mathbb{E}[\mathbf{x}^{k+1}] = W\,\mathbb{E}[\mathbf{x}^k] \;\Rightarrow\; \mathbb{E}[\bar{x}^{k+1}] = \tfrac{1}{n}\mathbf{1}^\top\mathbb{E}[\mathbf{x}^{k+1}] = \tfrac{1}{n}\mathbf{1}^\top W\mathbb{E}[\mathbf{x}^k] = \mathbb{E}[\bar{x}^k],$$

so the average is preserved in expectation.

For disagreement $\mathbf{y}^k = P\mathbf{x}^k$, note $PW = WP$ (since $W\mathbf{1} = \mathbf{1}$ and W is symmetric), hence

$$\mathbf{y}^{k+1} = P\mathbf{x}^{k+1} = PW\mathbf{x}^k + P\boldsymbol{\xi}^k = W\mathbf{y}^k + P\boldsymbol{\xi}^k.$$

With $\mathbb{E}[\boldsymbol{\xi}^k] = 0$ and independence across time, the disagreement covariance $\Sigma^k := \mathbb{E}[\mathbf{y}^k(\mathbf{y}^k)^\top]$ obeys

$$\Sigma^{k+1} = W\Sigma^k W^\top + \mathbb{E}[P\boldsymbol{\xi}^k(\boldsymbol{\xi}^k)^\top P] = W\Sigma^k W + \sigma^2 P.$$

If $|\lambda_i(W)| < 1$ for $i \geq 2$, then Σ^k converges to the unique solution of the following equation

$$\boxed{\Sigma = W\Sigma W + \sigma^2 P}$$

which yields a nonzero variance floor on the disagreement subspace.

(b) In the noisy consensus iteration

$$\mathbf{x}^{k+1} = W\mathbf{x}^k + \boldsymbol{\xi}^k, \qquad \mathbb{E}[\boldsymbol{\xi}^k] = 0,\ \mathrm{Cov}(\boldsymbol{\xi}^k) = \sigma^2 I,$$

suppose W is symmetric and doubly stochastic with eigenvalues $1 = \lambda_1 > \lambda_2 \geq \cdots \geq \lambda_n > -1$. At steady state, how does the disagreement variance in each eigenmode $i \geq 2$ depend on λ_i?

Hint: Since $\mathbf{y}^{k+1} = W\mathbf{y}^k + P\boldsymbol{\xi}^k$, in the eigenbasis of W, each mode i evolves as $y_i^{k+1} = \lambda_i y_i^k + \xi_i^k$.

(a) $\widetilde{\Sigma}_{ii} = \dfrac{\sigma^2}{1-\lambda_i}$

(b) $\widetilde{\Sigma}_{ii} = \dfrac{\sigma^2}{1-\lambda_i^2}$

(c) $\widetilde{\Sigma}_{ii} = \sigma^2(1-\lambda_i^2)$

(d) $\widetilde{\Sigma}_{ii} = \sigma^2\lambda_i^2$

Correct answer: (b). In the eigenbasis of W, each mode evolves as $y_i^{k+1} = \lambda_i y_i^k + \xi_i^k$, so the steady-state variance satisfies $\mathbb{E}[y_i^2] = \lambda_i^2\mathbb{E}[y_i^2] + \sigma^2$. Solving gives $\widetilde{\Sigma}_{ii} = \sigma^2/(1-\lambda_i^2)$ for $i \geq 2$.

(c) How does the *spectral gap* $\delta := 1 - \max_{i \geq 2} |\lambda_i(W)|$ affect the steady-state level of disagreement across the network?

(a) A larger spectral gap leads to *more* steady-state disagreement.
(b) A smaller spectral gap leads to *less* steady-state disagreement.
(c) The steady-state disagreement variance is roughly proportional to $1/\delta$, so smaller gaps (weaker connectivity) yield *larger* disagreement.
(d) The spectral gap does not affect disagreement in steady state.

Correct answer: (c). Because the per-mode variance grows as $\sigma^2/(1 - \lambda_i^2) \approx \sigma^2/[2(1 - \lambda_i)]$ for λ_i near 1, the overall disagreement is inversely proportional to the spectral gap δ. A smaller gap (poorly connected network) slows information mixing and leads to a higher noise floor, while a larger gap (better connected) keeps nodes more closely synchronized.