Data Science for Mathematicians
Exercises 5: Gradient Descent

Instructions

Answer all exercises completely. Show all working, justify your answers, and state any assumptions you make. For computational exercises, carry out all intermediate steps explicitly. For proof exercises, clearly identify which definitions and theorems you are applying.

Exercises

Exercise 1. Gradient, Hessian, and Critical-Point Classification
Consider the loss function $L(\theta_1, \theta_2) = 3\theta_1^2 - 4\theta_1\theta_2 + 2\theta_2^2 + \theta_1 - 2\theta_2$.

(a) Compute the gradient $\nabla L(\theta_1, \theta_2)$.

(b) Compute the Hessian matrix $\mathbf{H}_L$ and verify that it is symmetric.

(c) Find the critical point by solving $\nabla L = \mathbf{0}$. Show all steps.

(d) Compute the eigenvalues of $\mathbf{H}_L$ and classify the critical point.

Exercise 2. Two Manual Batch Gradient Descent Steps
We model $y = \beta x$ (regression through the origin) on four samples: $(1, 2), (2, 4), (3, 5), (4, 8)$.

(a) Write the design matrix $\mathbf{X} \in \mathbb{R}^{4\times 1}$ and response vector $\mathbf{y} \in \mathbb{R}^4$.

(b) State the gradient formula $\nabla L(\beta) = \frac{2}{n}\mathbf{X}^T(\mathbf{X}\beta - \mathbf{y})$ and evaluate it at $\beta = 0$.

(c) Starting from $\beta_0 = 0$ with learning rate $\eta = 0.05$, compute β_1 and β_2.

(d) Compute $L(\beta_0)$, $L(\beta_1)$, and $L(\beta_2)$, where $L(\beta) = \frac{1}{n}\|\mathbf{X}\beta - \mathbf{y}\|_2^2$. Verify that the loss decreases at each step.

Exercise 3. Convexity via Hessian Definiteness
For each function below, compute the Hessian, determine its definiteness, and state whether the function is convex, concave, or neither.

(a) $f(x_1, x_2) = x_1^2 + 4x_1x_2 + 4x_2^2$.

(b) $g(x_1, x_2) = -x_1^2 - x_2^2$.

(c) $h(x_1, x_2) = x_1^2 - x_2^2$.

Exercise 4. Learning Rate Sensitivity
Consider the loss $L(\theta) = 2\theta^2$ with $\theta_0 = 5$. The gradient descent update is $\theta_{k+1} = \theta_k - \eta\nabla L(\theta_k)$.

(a) Show that the update rule simplifies to $\theta_{k+1} = (1 - 4\eta)\,\theta_k$.

(b) With $\eta = 0.1$: compute θ_1, θ_2, θ_3. Does the sequence converge?

(c) With $\eta = 0.3$: compute θ_1, θ_2, θ_3. Does the sequence converge?

(d) With $\eta = 0.5$: show analytically that $\theta_k = 5(-1)^k$ for all $k \geq 0$. Does gradient descent converge in this case?

Exercise 5. Deriving the OLS Gradient

Let $L(\boldsymbol{\beta}) = \frac{1}{n}\|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2$, where $\mathbf{X} \in \mathbb{R}^{n \times p}$ and $\mathbf{y} \in \mathbb{R}^n$.

(a) Expand $\|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2$ as a scalar quadratic form in $\boldsymbol{\beta}$. Your result should contain three terms.

(b) Using the identities $\nabla_{\boldsymbol{\beta}}(\mathbf{c}^T\boldsymbol{\beta}) = \mathbf{c}$ and $\nabla_{\boldsymbol{\beta}}(\boldsymbol{\beta}^T\mathbf{A}\boldsymbol{\beta}) = 2\mathbf{A}\boldsymbol{\beta}$ (for symmetric $\mathbf{A}$), differentiate term by term.

(c) Combine the results to obtain $\nabla L(\boldsymbol{\beta}) = \frac{2}{n}\mathbf{X}^T(\mathbf{X}\boldsymbol{\beta} - \mathbf{y})$.

Exercise 6. Convexity is Preserved Under Non-Negative Linear Combinations

Let f and g be convex functions on a convex set $C \subseteq \mathbb{R}^p$, and let $\alpha, \beta \geq 0$.

(a) Using only the definition of convexity, prove that $h = \alpha f + \beta g$ is convex on C.

(b) As a consequence, explain why any finite sum $\frac{1}{n}\sum_{i=1}^n L_i(\boldsymbol{\theta})$ of convex per-sample losses is a convex function of $\boldsymbol{\theta}$.

(c) Give an example of two convex functions whose pointwise *product* is not convex. (A simple one-variable example suffices.)

Exercise 7. Unbiasedness of the Stochastic Gradient

Let $L(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^n L_i(\boldsymbol{\theta})$ be the full loss, and let index i be drawn uniformly at random from $\{1, \ldots, n\}$.

(a) Prove that $\mathbb{E}_i[\nabla L_i(\boldsymbol{\theta})] = \nabla L(\boldsymbol{\theta})$. State which property of expectation you use.

(b) Explain why unbiasedness alone is insufficient for convergence of stochastic gradient descent. What additional conditions on the step-size schedule $\{\eta_k\}$ are required? State the Robbins–Monro conditions.

(c) Suppose a practitioner uses a fixed step size $\eta_k = \eta$ for all k. Describe qualitatively how the iterates behave as $k \to \infty$, and contrast this with the behavior under a decaying schedule.

Exercise 8. Per-Step Loss Decrease for Smooth Convex Functions

Suppose $L\colon \mathbb{R}^p \to \mathbb{R}$ is convex and has an M-Lipschitz gradient, meaning

$$L(\boldsymbol{\theta}') \leq L(\boldsymbol{\theta}) + \nabla L(\boldsymbol{\theta})^T(\boldsymbol{\theta}' - \boldsymbol{\theta}) + \frac{M}{2}\|\boldsymbol{\theta}' - \boldsymbol{\theta}\|_2^2$$

for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^p$.

(a) Substitute one batch gradient descent step $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta\,\nabla L(\boldsymbol{\theta}_k)$ into the above inequality to obtain an upper bound on $L(\boldsymbol{\theta}_{k+1})$.

(b) Show that for $\eta \leq 1/M$, the upper bound implies

$$L(\boldsymbol{\theta}_{k+1}) \leq L(\boldsymbol{\theta}_k) - \frac{\eta}{2}\|\nabla L(\boldsymbol{\theta}_k)\|_2^2.$$

(c) Interpret this result: for what values of $\boldsymbol{\theta}_k$ does gradient descent make no progress? What does this say about the fixed points of the update?

Exercise 9. Saddle Points and Non-Convex Loss Landscapes
Consider the function $f(\theta) = \theta^4 - 4\theta^2 + \theta$.

(a) Compute $f'(\theta)$ and find all critical points by solving $f'(\theta) = 0$. (You may solve numerically to two decimal places; the equation has three real roots near $\theta \approx -1.77$, $\theta \approx 0.13$, and $\theta \approx 1.64$.) Classify each critical point using $f''(\theta)$.

(b) Starting gradient descent from $\theta_0 = -1$, which critical point does the algorithm converge to? Is it the global minimum? Explain why.

(c) Contrast this behavior with that of gradient descent on a convex loss. State the theorem from the lecture that guarantees gradient descent on the OLS loss finds the global minimum.

Exercise 10. Tracing One Epoch: BGD, SGD, and Mini-Batch GD
We fit the model $y = \beta x$ to the dataset below using the loss $L_i(\beta) = (x_i\beta - y_i)^2$ for each sample.

i	1	2	3	4	5	6
x_i	1	2	3	4	5	6
y_i	1	3	4	6	8	9

Use $\beta_0 = 0$ and $\eta = 0.02$ throughout.

(a) For batch gradient descent, compute $\nabla L(\beta_0) = \frac{2}{n}\sum_{i=1}^{6} x_i(x_i\beta_0 - y_i)$ and report β_1 after one update.

(b) For SGD (process samples in order $i = 1, 2, \ldots, 6$), the per-sample update is $\beta \leftarrow \beta - \eta \cdot 2x_i(x_i\beta - y_i)$. Starting from $\beta_0 = 0$, carry out all six updates and report β after each one.

(c) For mini-batch gradient descent with $b = 2$ and batches $\mathcal{B}_1 = \{1, 2\}$, $\mathcal{B}_2 = \{3, 4\}$, $\mathcal{B}_3 = \{5, 6\}$, carry out three updates and report β after each one.

(d) The exact OLS solution is $\beta^* = \mathbf{X}^T\mathbf{y}/(\mathbf{X}^T\mathbf{X})$. Compute β^* and determine which variant (BGD, SGD, or mini-batch) ended the epoch closest to β^*.

Exercise 11. Mini-Batch Variance Reduction
Assume that per-sample gradients $\nabla L_i(\boldsymbol{\theta})$ are independent with $\mathbb{E}[\nabla L_i] = \mathbf{g}$ and $\mathrm{Var}(\nabla L_i) = \sigma^2$.

(a) Define the mini-batch gradient $\mathbf{g}_k = \frac{1}{b}\sum_{i\in\mathcal{B}_k} \nabla L_i(\boldsymbol{\theta})$ and prove that $\mathbb{E}[\mathbf{g}_k] = \mathbf{g}$.

(b) Using independence of the per-sample gradients, prove that $\mathrm{Var}(\mathbf{g}_k) = \sigma^2/b$.

(c) Suppose $\sigma^2 = 8$ and a practitioner requires $\mathrm{Var}(\mathbf{g}_k) \leq 0.5$. What is the minimum batch size b needed?

(d) For $n = 1000$, compute the number of gradient evaluations per epoch for $b = 1, 10, 100, 1000$. Comment on the trade-off between update frequency and gradient variance.

Exercise 12. Diagnosing and Fixing a Diverging Run

A practitioner runs batch gradient descent on a regression problem with $n = 500$, $p = 10$ features. The loss *increases* at every iteration.

(a) Using the per-step decrease bound $L(\boldsymbol{\theta}_{k+1}) \leq L(\boldsymbol{\theta}_k) - \frac{\eta}{2}\|\nabla L(\boldsymbol{\theta}_k)\|_2^2 + \text{error}$, explain under what condition on η the bound fails and divergence occurs. What is the role of the Lipschitz constant M?

(b) The Hessian of the OLS loss is $\frac{2}{n}\mathbf{X}^T\mathbf{X}$. Suppose the largest eigenvalue of $\mathbf{X}^T\mathbf{X}$ is $\lambda_{\max} = 250$. Compute M (the largest eigenvalue of the Hessian) and derive the safe upper bound on η.

(c) If the practitioner switches to mini-batch gradient descent with batch size $b = 50$, does the safe upper bound on η change? Justify your answer carefully.

Solutions

Solution 1. Gradient, Hessian, and Critical-Point Classification

(a) Computing partial derivatives:

$$\frac{\partial L}{\partial \theta_1} = 6\theta_1 - 4\theta_2 + 1, \qquad \frac{\partial L}{\partial \theta_2} = -4\theta_1 + 4\theta_2 - 2.$$

Therefore $\nabla L(\theta_1, \theta_2) = \begin{bmatrix} 6\theta_1 - 4\theta_2 + 1 \\ -4\theta_1 + 4\theta_2 - 2 \end{bmatrix}$.

(b) The Hessian is the matrix of second-order partial derivatives:

$$\mathbf{H}_L = \begin{bmatrix} \partial^2 L/\partial\theta_1^2 & \partial^2 L/\partial\theta_1\partial\theta_2 \\ \partial^2 L/\partial\theta_2\partial\theta_1 & \partial^2 L/\partial\theta_2^2 \end{bmatrix} = \begin{bmatrix} 6 & -4 \\ -4 & 4 \end{bmatrix}.$$

Symmetry is immediate since $\partial^2 L/\partial\theta_1\partial\theta_2 = -4 = \partial^2 L/\partial\theta_2\partial\theta_1$. By Schwarz's theorem, this holds for any twice continuously differentiable function.

(c) We solve $\nabla L = \mathbf{0}$:

$$6\theta_1 - 4\theta_2 + 1 = 0, \qquad -4\theta_1 + 4\theta_2 - 2 = 0.$$

Adding the two equations: $2\theta_1 - 1 = 0$, so $\theta_1 = \frac{1}{2}$. Substituting into the second equation: $-2 + 4\theta_2 - 2 = 0$, giving $\theta_2 = 1$. The unique critical point is $(\theta_1^*, \theta_2^*) = \left(\frac{1}{2}, 1\right)$.

(d) The characteristic polynomial of $\mathbf{H}_L$ is

$$\det(\mathbf{H}_L - \lambda\mathbf{I}) = (6-\lambda)(4-\lambda) - (-4)^2 = \lambda^2 - 10\lambda + 24 - 16 = \lambda^2 - 10\lambda + 8.$$

The eigenvalues are $\lambda = \frac{10 \pm \sqrt{100-32}}{2} = 5 \pm \sqrt{17}$. Numerically, $\lambda_1 \approx 9.12 > 0$ and $\lambda_2 \approx 0.88 > 0$. Since both eigenvalues are strictly positive, $\mathbf{H}_L$ is positive definite. The critical point $\left(\frac{1}{2}, 1\right)$ is a strict global minimum.

Solution 2. Two Manual Batch Gradient Descent Steps

(a) $\mathbf{X} = [1, 2, 3, 4]^T \in \mathbb{R}^4$ and $\mathbf{y} = [2, 4, 5, 8]^T \in \mathbb{R}^4$, with $n = 4$.

(b) At $\beta = 0$, we have $\mathbf{X}\beta - \mathbf{y} = -\mathbf{y} = [-2, -4, -5, -8]^T$. Therefore:

$$\mathbf{X}^T(\mathbf{X}\beta_0 - \mathbf{y}) = 1\cdot(-2) + 2\cdot(-4) + 3\cdot(-5) + 4\cdot(-8) = -2 - 8 - 15 - 32 = -57.$$

$$\nabla L(\beta_0) = \tfrac{2}{4} \cdot (-57) = -28.5.$$

(c) Applying the update rule $\beta_{k+1} = \beta_k - \eta\,\nabla L(\beta_k)$:

$$\beta_1 = 0 - 0.05 \times (-28.5) = 1.425.$$

At $\beta_1 = 1.425$, the residuals are $\mathbf{X}\beta_1 - \mathbf{y} = [1.425 - 2,\ 2.85 - 4,\ 4.275 - 5,\ 5.7 - 8]^T = [-0.575, -1.15, -0.725, -2.3]^T$.

$$\mathbf{X}^T(\mathbf{X}\beta_1 - \mathbf{y}) = 1(-0.575) + 2(-1.15) + 3(-0.725) + 4(-2.3) = -14.25.$$

$$\nabla L(\beta_1) = \tfrac{2}{4}\cdot(-14.25) = -7.125, \qquad \beta_2 = 1.425 - 0.05 \times (-7.125) = 1.78125.$$

(d) We compute $L(\beta) = \frac{1}{4}\sum_{i=1}^{4}(ix \cdot \beta - y_i)^2$ at each iterate:

$$\begin{aligned}
L(\beta_0) &= \tfrac{1}{4}(4 + 16 + 25 + 64) = \tfrac{109}{4} = 27.25, \\
L(\beta_1) &= \tfrac{1}{4}\big(0.575^2 + 1.15^2 + 0.725^2 + 2.3^2\big) \\
&= \tfrac{1}{4}(0.3306 + 1.3225 + 0.5256 + 5.29) \approx 1.867, \\
L(\beta_2) &= \tfrac{1}{4}\big(0.21875^2 + 0.4375^2 + 0.34375^2 + 0.875^2\big) \\
&= \tfrac{1}{4}(0.0479 + 0.1914 + 0.1182 + 0.7656) \approx 0.281.
\end{aligned}$$

The sequence $27.25 > 1.867 > 0.281$ confirms monotone decrease of the loss.

Solution 3. Convexity via Hessian Definiteness

(a) $f(x_1, x_2) = x_1^2 + 4x_1x_2 + 4x_2^2 = (x_1 + 2x_2)^2$. The Hessian is

$$\mathbf{H}_f = \begin{bmatrix} 2 & 4 \\ 4 & 8 \end{bmatrix}.$$

The eigenvalues satisfy $\lambda^2 - 10\lambda + 16 - 16 = \lambda(\lambda - 10) = 0$, giving $\lambda_1 = 10$ and $\lambda_2 = 0$. Since both eigenvalues are non-negative, $\mathbf{H}_f \succeq \mathbf{0}$ (positive semidefinite). Therefore f is **convex** (but not strictly convex).

(b) $g(x_1, x_2) = -x_1^2 - x_2^2$. The Hessian is

$$\mathbf{H}_g = \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix} = -2\mathbf{I}.$$

Both eigenvalues equal $-2 < 0$, so $\mathbf{H}_g \preceq \mathbf{0}$ (negative definite). Therefore g is **concave**.

(c) $h(x_1, x_2) = x_1^2 - x_2^2$. The Hessian is

$$\mathbf{H}_h = \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix}.$$

The eigenvalues are $\lambda_1 = 2 > 0$ and $\lambda_2 = -2 < 0$. Since the Hessian is indefinite, h is **neither convex nor concave**.

Solution 4. Learning Rate Sensitivity

(a) We have $\nabla L(\theta) = \frac{\mathrm{d}}{\mathrm{d}\theta}(2\theta^2) = 4\theta$. Substituting into the update:

$$\theta_{k+1} = \theta_k - \eta \cdot 4\theta_k = (1 - 4\eta)\,\theta_k.$$

(b) With $\eta = 0.1$, the multiplier is $1 - 0.4 = 0.6$:

$$\theta_1 = 0.6 \times 5 = 3, \quad \theta_2 = 0.6 \times 3 = 1.8, \quad \theta_3 = 0.6 \times 1.8 = 1.08.$$

Since $|0.6| < 1$, the iterates satisfy $\theta_k = 5 \cdot (0.6)^k \to 0$ as $k \to \infty$. The sequence converges to the minimum $\theta^* = 0$.

(c) With $\eta = 0.3$, the multiplier is $1 - 1.2 = -0.2$:

$$\theta_1 = -0.2 \times 5 = -1, \quad \theta_2 = -0.2 \times (-1) = 0.2, \quad \theta_3 = -0.2 \times 0.2 = -0.04.$$

Since $|-0.2| < 1$, the iterates satisfy $\theta_k = 5 \cdot (-0.2)^k \to 0$. The sequence converges to $\theta^* = 0$, though with sign oscillations.

(d) With $\eta = 0.5$, the multiplier is $1-2.0 = -1$. We prove the claim by induction. Base case: $\theta_0 = 5 = 5(-1)^0$. Inductive step: if $\theta_k = 5(-1)^k$, then $\theta_{k+1} = (-1)\theta_k = 5(-1)^{k+1}$. Hence $\theta_k = 5(-1)^k$ for all $k \geq 0$.
Since $|\theta_k| = 5$ for all k, the sequence oscillates permanently between 5 and -5 and does not converge. Gradient descent fails at this learning rate.

Solution 5. Deriving the OLS Gradient

(a) Expanding the squared norm:

$$\begin{aligned}\|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 &= (\mathbf{X}\boldsymbol{\beta} - \mathbf{y})^T(\mathbf{X}\boldsymbol{\beta} - \mathbf{y}) \\ &= \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} - \mathbf{y}^T\mathbf{X}\boldsymbol{\beta} - \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{y} + \mathbf{y}^T\mathbf{y}.\end{aligned}$$

Since $\mathbf{y}^T\mathbf{X}\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{y})^T\boldsymbol{\beta}$ is a scalar and equals $\boldsymbol{\beta}^T\mathbf{X}^T\mathbf{y}$, we obtain

$$L(\boldsymbol{\beta}) = \frac{1}{n}\big(\boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} - 2\mathbf{y}^T\mathbf{X}\boldsymbol{\beta} + \mathbf{y}^T\mathbf{y}\big).$$

(b) We differentiate term by term. The matrix $\mathbf{A} = \mathbf{X}^T\mathbf{X}$ is symmetric, so:

$$\begin{aligned}\nabla_{\boldsymbol{\beta}}\big(\boldsymbol{\beta}^T\mathbf{A}\boldsymbol{\beta}\big) &= 2\mathbf{A}\boldsymbol{\beta} = 2\mathbf{X}^T\mathbf{X}\boldsymbol{\beta}, \\ \nabla_{\boldsymbol{\beta}}\big(-2\mathbf{y}^T\mathbf{X}\boldsymbol{\beta}\big) &= -2\mathbf{X}^T\mathbf{y}, \\ \nabla_{\boldsymbol{\beta}}\big(\mathbf{y}^T\mathbf{y}\big) &= \mathbf{0}.\end{aligned}$$

(c) Combining and multiplying by $\frac{1}{n}$:

$$\nabla L(\boldsymbol{\beta}) = \frac{1}{n}\big(2\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} - 2\mathbf{X}^T\mathbf{y}\big) = \frac{2}{n}\mathbf{X}^T(\mathbf{X}\boldsymbol{\beta} - \mathbf{y}). \qquad \square$$

Solution 6. Convexity is Preserved Under Non-Negative Linear Combinations

(a) Let $\mathbf{x}, \mathbf{y} \in C$ and $t \in [0,1]$. Since C is convex, $(1-t)\mathbf{x}+t\mathbf{y} \in C$. We compute

$$\begin{aligned}h\big((1-t)\mathbf{x} + t\mathbf{y}\big) &= \alpha f\big((1-t)\mathbf{x} + t\mathbf{y}\big) + \beta g\big((1-t)\mathbf{x} + t\mathbf{y}\big) \\ &\leq \alpha\big[(1-t)f(\mathbf{x}) + tf(\mathbf{y})\big] + \beta\big[(1-t)g(\mathbf{x}) + tg(\mathbf{y})\big] \\ &= (1-t)\big[\alpha f(\mathbf{x}) + \beta g(\mathbf{x})\big] + t\big[\alpha f(\mathbf{y}) + \beta g(\mathbf{y})\big] \\ &= (1-t)h(\mathbf{x}) + th(\mathbf{y}).\end{aligned}$$

The inequality uses convexity of f and g together with $\alpha, \beta \geq 0$. Hence h is convex. $\square$

(b) The empirical loss $L(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^n L_i(\boldsymbol{\theta})$ is a non-negative linear combination of n functions (each with coefficient $1/n > 0$). By repeated application of part (a), if each L_i is convex, then L is convex.

(c) Let $f(x) = x$ and $g(x) = x$ on $\mathbb{R}$, both convex. Their product $h(x) = x^2$ is convex, which is not a counterexample. Instead, consider $f(x) = x$ and $g(x) = -x+1$ on $[0,1]$: both are convex (affine), but $h(x) = x(1-x) = x-x^2$ has $h''(x) = -2 < 0$, so h is concave and not convex.

Solution 7. Unbiasedness of the Stochastic Gradient

(a) Since i is drawn uniformly from $\{1, \ldots, n\}$, each index is chosen with probability $1/n$. By linearity of expectation:

$$\mathbb{E}_i[\nabla L_i(\boldsymbol{\theta})] = \sum_{i=1}^{n} \mathbb{P}(i) \cdot \nabla L_i(\boldsymbol{\theta}) = \sum_{i=1}^{n} \frac{1}{n} \nabla L_i(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \nabla L_i(\boldsymbol{\theta}) = \nabla L(\boldsymbol{\theta}).$$

We use the linearity of expectation and the fact that expectation of a discrete random variable is the probability-weighted sum. □

(b) Unbiasedness guarantees that on average the stochastic gradient points in the correct direction, but individual updates are noisy. With a fixed learning rate η, the noise in the gradient prevents exact convergence to $\boldsymbol{\theta}^*$; the iterates fluctuate around the optimum indefinitely.

The step-size schedule $\{\eta_k\}$ must satisfy the following Robbins–Monro conditions:

$$\sum_{k=1}^{\infty} \eta_k = \infty \quad \text{(steps are large enough to reach } \boldsymbol{\theta}^*\text{)},$$

$$\sum_{k=1}^{\infty} \eta_k^2 < \infty \quad \text{(steps are small enough for variance to vanish)}.$$

A common choice satisfying both conditions is $\eta_k = c/k$ for some constant $c > 0$.

(c) With a fixed $\eta_k = \eta$, the Robbins–Monro conditions are violated (since $\sum \eta^2 = \infty$). The iterates do not converge to a single point; instead, they enter a neighborhood of $\boldsymbol{\theta}^*$ whose size is controlled by $\eta\sigma$, where σ^2 is the gradient variance. They continue to fluctuate within this neighborhood as $k \to \infty$.

Under a decaying schedule (e.g. $\eta_k = c/k$), the gradient variance contribution vanishes as $\eta_k \to 0$, and the iterates converge to $\boldsymbol{\theta}^*$.

Solution 8. Per-Step Loss Decrease for Smooth Convex Functions

(a) We substitute $\boldsymbol{\theta}' = \boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta \nabla L(\boldsymbol{\theta}_k)$ and $\boldsymbol{\theta} = \boldsymbol{\theta}_k$ into the descent lemma. Then $\boldsymbol{\theta}' - \boldsymbol{\theta} = -\eta \nabla L(\boldsymbol{\theta}_k)$, so:

$$\begin{aligned} L(\boldsymbol{\theta}_{k+1}) &\leq L(\boldsymbol{\theta}_k) + \nabla L(\boldsymbol{\theta}_k)^T(-\eta \nabla L(\boldsymbol{\theta}_k)) + \frac{M}{2} \| - \eta \nabla L(\boldsymbol{\theta}_k)\|_2^2 \\ &= L(\boldsymbol{\theta}_k) - \eta \|\nabla L(\boldsymbol{\theta}_k)\|_2^2 + \frac{M\eta^2}{2} \|\nabla L(\boldsymbol{\theta}_k)\|_2^2. \end{aligned}$$

(b) Factoring the right-hand side:

$$L(\boldsymbol{\theta}_{k+1}) \leq L(\boldsymbol{\theta}_k) - \eta \left(1 - \frac{M\eta}{2}\right) \|\nabla L(\boldsymbol{\theta}_k)\|_2^2.$$

For $\eta \leq 1/M$, we have $M\eta \leq 1$, so $1 - \frac{M\eta}{2} \geq 1 - \frac{1}{2} = \frac{1}{2}$. Therefore:

$$L(\boldsymbol{\theta}_{k+1}) \leq L(\boldsymbol{\theta}_k) - \frac{\eta}{2} \|\nabla L(\boldsymbol{\theta}_k)\|_2^2. \qquad \square$$

(c) The bound guarantees $L(\boldsymbol{\theta}_{k+1}) \leq L(\boldsymbol{\theta}_k)$ with strict decrease proportional to $\|\nabla L(\boldsymbol{\theta}_k)\|_2^2$. No progress occurs only when $\|\nabla L(\boldsymbol{\theta}_k)\|_2^2 = 0$, i.e., when $\nabla L(\boldsymbol{\theta}_k) = \mathbf{0}$. For a convex function, $\nabla L(\boldsymbol{\theta}) = \mathbf{0}$ if and only if $\boldsymbol{\theta}$ is a global minimizer. The fixed points of the gradient descent update are precisely the global minima of L.

Solution 9. Saddle Points and Non-Convex Loss Landscapes

(a) We have $f'(\theta) = 4\theta^3 - 8\theta + 1$ and $f''(\theta) = 12\theta^2 - 8$.
Numerically, the three real roots of $f'(\theta) = 0$ are approximately:

Critical point θ^*	$f''(\theta^*)$	Classification
$\theta_1 \approx -1.77$	$f''(-1.77) = 12(3.13) - 8 \approx 29.6 > 0$	Local minimum
$\theta_2 \approx 0.13$	$f''(0.13) = 12(0.017) - 8 \approx -7.8 < 0$	Local maximum
$\theta_3 \approx 1.64$	$f''(1.64) = 12(2.69) - 8 \approx 24.3 > 0$	Local minimum

Evaluating: $f(-1.77) \approx (-1.77)^4 - 4(-1.77)^2 + (-1.77) \approx 9.82 - 12.53 - 1.77 \approx -4.48$ and $f(1.64) \approx 7.24 - 10.76 + 1.64 \approx -1.88$. Thus $\theta_1 \approx -1.77$ is the global minimum.

(b) From $\theta_0 = -1$, we compute $f'(-1) = 4(-1)^3 - 8(-1) + 1 = -4 + 8 + 1 = 5 > 0$. The gradient points to the right, so gradient descent moves to the left, toward $\theta_1 \approx -1.77$. The algorithm converges to the local minimum at θ_1, which is also the global minimum in this case.
However, if we had started at $\theta_0 = 1$, the gradient $f'(1) = 4 - 8 + 1 = -3 < 0$ would push the iterates to the right, toward the local (but not global) minimum at $\theta_3 \approx 1.64$. This illustrates how non-convex loss landscapes can trap gradient descent in suboptimal local minima depending on the initialization.

(c) For a convex loss, every local minimum is a global minimum. The lecture establishes this via the Global Optimality Theorem: if L is convex and $\nabla L(\boldsymbol{\theta}^*) = \mathbf{0}$, then $\boldsymbol{\theta}^*$ is a global minimizer of L. For the OLS loss $L(\boldsymbol{\beta}) = \frac{1}{n}\|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2$, the Hessian $\frac{2}{n}\mathbf{X}^T\mathbf{X}$ is positive semidefinite, so L is convex, and gradient descent converges to the global minimum regardless of initialization (provided $\eta \leq 1/M$).

Solution 10. Tracing One Epoch: BGD, SGD, and Mini-Batch GD

(a) We have $n = 6$. At $\beta_0 = 0$, the full gradient is

$$\begin{aligned}\nabla L(\beta_0) &= \frac{2}{6}\sum_{i=1}^{6} x_i(x_i \cdot 0 - y_i) \\ &= \frac{2}{6}\sum_{i=1}^{6}(-x_i y_i) \\ &= -\frac{1}{3}(1 + 6 + 12 + 24 + 40 + 54) = -\frac{137}{3} \approx -45.67.\end{aligned}$$

$$\beta_1 = 0 - 0.02 \times (-45.67) \approx 0.9133.$$

(b) We apply $\beta \leftarrow \beta - 0.02 \cdot 2x_i(x_i\beta - y_i)$ for $i = 1, \ldots, 6$ in sequence (values rounded to 4 decimal places):

Step i	Update gradient $2x_i(x_i\beta - y_i)$	β after update
1	$2(1)(0 - 1) = -2$	$0 + 0.04 = 0.0400$
2	$2(2)(2 \times 0.04 - 3) = 4(-2.92) = -11.68$	$0.04 + 0.2336 = 0.2736$
3	$2(3)(3 \times 0.2736 - 4) = 6(-3.1792) = -19.075$	$0.2736 + 0.3815 = 0.6551$
4	$2(4)(4 \times 0.6551 - 6) = 8(-3.3796) = -27.037$	$0.6551 + 0.5407 = 1.1958$
5	$2(5)(5 \times 1.1958 - 8) = 10(-2.021) = -20.21$	$1.1958 + 0.4042 = 1.6000$
6	$2(6)(6 \times 1.6 - 9) = 12(0.6) = 7.20$	$1.6000 - 0.1440 = 1.4560$

After one SGD epoch: $\beta \approx 1.4560$.

(c) The mini-batch gradient for batch $\mathcal{B} = \{i, j\}$ at parameter β is $\frac{1}{2}[2x_i(x_i\beta - y_i) + 2x_j(x_j\beta - y_j)]$.
Batch $\mathcal{B}_1 = \{1, 2\}$ at $\beta = 0$:

$g = \frac{1}{2}[2(1)(0-1)+2(2)(0-3)] = \frac{1}{2}[-2-12] = -7. \quad \beta_1 = 0-0.02(-7) = 0.14.$

Batch $\mathcal{B}_2 = \{3, 4\}$ at $\beta = 0.14$:

$g = \frac{1}{2}[6(0.42-4)+8(0.56-6)] = \frac{1}{2}[-21.48-43.52] = -32.5. \quad \beta_2 = 0.14+0.65 = 0.79.$

Batch $\mathcal{B}_3 = \{5, 6\}$ at $\beta = 0.79$:

$g = \frac{1}{2}[10(3.95-8)+12(4.74-9)] = \frac{1}{2}[-40.5-51.12] = -45.81. \quad \beta_3 = 0.79+0.9162 = 1.7062.$

After one mini-batch epoch: $\beta \approx 1.7062$.

(d) We compute $\beta^* = \mathbf{X}^T\mathbf{y}/(\mathbf{X}^T\mathbf{X})$:

$\mathbf{X}^T\mathbf{y} = 1(1)+2(3)+3(4)+4(6)+5(8)+6(9) = 137, \quad \mathbf{X}^T\mathbf{X} = 1+4+9+16+25+36 = 91.$

$$\beta^* = \frac{137}{91} \approx 1.5055.$$

The distances to β^* after one epoch are as follows:

$$|\beta_1^{\text{BGD}} - \beta^*| \approx |0.9133 - 1.5055| = 0.592,$$
$$|\beta^{\text{SGD}} - \beta^*| \approx |1.456 - 1.5055| = 0.050,$$
$$|\beta^{\text{MB}} - \beta^*| \approx |1.7062 - 1.5055| = 0.201.$$

SGD ended the epoch closest to β^*. Note that BGD made only one update (the fewest), while SGD made six per-sample updates (and thus more total progress) within a single epoch.

Solution 11. Mini-Batch Variance Reduction

(a) Since the samples in batch $\mathcal{B}_k$ are drawn from $\{1, \ldots, n\}$ and each ∇L_i has expectation $\mathbf{g}$, linearity of expectation gives

$$\mathbb{E}[\mathbf{g}_k] = \mathbb{E}\left[\frac{1}{b}\sum_{i\in\mathcal{B}_k} \nabla L_i(\boldsymbol{\theta})\right] = \frac{1}{b}\sum_{i\in\mathcal{B}_k} \mathbb{E}[\nabla L_i(\boldsymbol{\theta})] = \frac{1}{b}\cdot b\,\mathbf{g} = \mathbf{g}.$$

The mini-batch gradient is an unbiased estimator of the full gradient. □

(b) Since the per-sample gradients are independent with variance σ^2, and the variance of a sum of independent random variables is the sum of variances:

$$\text{Var}(\mathbf{g}_k) = \text{Var}\left(\frac{1}{b}\sum_{i\in\mathcal{B}_k} \nabla L_i\right) = \frac{1}{b^2}\sum_{i\in\mathcal{B}_k} \text{Var}(\nabla L_i) = \frac{1}{b^2}\cdot b\sigma^2 = \frac{\sigma^2}{b}. \qquad \square$$

(c) We require $\sigma^2/b \le 0.5$, i.e., $b \ge \sigma^2/0.5 = 8/0.5 = 16$. The minimum batch size is $b = 16$.

(d) For $n = 1000$, each epoch consists of n/b mini-batches, and each mini-batch requires b gradient evaluations, giving a total of $n = 1000$ gradient evaluations per epoch regardless of b. The number of *parameter updates* per epoch is n/b:

Batch size b	Updates per epoch	Gradient evals per epoch	$\mathrm{Var}(\mathbf{g}_k) = \sigma^2/b$
1	1000	1000	σ^2
10	100	1000	$\sigma^2/10$
100	10	1000	$\sigma^2/100$
1000	1	1000	$\sigma^2/1000$

The trade-off is as follows. Smaller batches yield more frequent parameter updates per epoch (exploring more of the loss landscape) but with higher gradient variance (noisier updates). Larger batches reduce gradient variance and enable more stable updates, but at the cost of fewer updates per epoch. In practice, batch sizes of 32–256 are common, balancing this trade-off while also exploiting hardware parallelism (GPUs).

Solution 12. Diagnosing and Fixing a Diverging Run

(a) From Exercise 8, the per-step decrease bound is

$$L(\boldsymbol{\theta}_{k+1}) \leq L(\boldsymbol{\theta}_k) - \eta\left(1 - \frac{M\eta}{2}\right) \|\nabla L(\boldsymbol{\theta}_k)\|_2^2.$$

The bound guarantees a decrease only when the coefficient of $\|\nabla L(\boldsymbol{\theta}_k)\|_2^2$ is non-negative, i.e., when $\eta \leq 2/M$. For the tighter convergence guarantee (the $\frac{\eta}{2}$ bound), we require $\eta \leq 1/M$. When $\eta > 2/M$, the quadratic term $\frac{M\eta^2}{2}\|\nabla L\|^2$ dominates $\eta\|\nabla L\|^2$, so the loss can increase. The Lipschitz constant M of the gradient captures the maximum curvature of the loss landscape; a large M means the gradient changes rapidly, requiring a small step size.

(b) The Hessian of the OLS loss is $\mathbf{H} = \frac{2}{n}\mathbf{X}^T\mathbf{X}$. Its largest eigenvalue is

$$M = \frac{2}{n}\lambda_{\max}(\mathbf{X}^T\mathbf{X}) = \frac{2}{500} \times 250 = 1.0.$$

The safe upper bound on the learning rate is $\eta \leq 1/M = 1$. Any $\eta > 2/M = 2$ guarantees divergence; any $\eta > 1$ may diverge or oscillate.

(c) The safe learning rate bound does **not** change. The bound $\eta \leq 1/M$ is derived from the Lipschitz constant of the gradient of the *full* loss function L, not of the mini-batch loss. The full loss landscape has the same curvature regardless of which subset of samples we use to estimate the gradient. Reducing the batch size increases gradient variance (noisy updates) but does not affect the curvature of L. Therefore, to prevent divergence, the practitioner must still satisfy $\eta \leq 1$.

In practice, a common heuristic is to scale the learning rate proportionally to the batch size ($\eta \propto b$) when increasing b, but this is a practical tuning strategy, not a change to the theoretical safe bound derived from M.