Total time: 90 mins Total marks: 18 Name and Roll Number: ______________________

Questions

1. (3 points) Compute the ℓ_0, ℓ_1, ℓ_2, and ℓ_∞ norms of the vector

$$\mathbf{x} = [0, 1, 0, 2, 3].$$

Which norm is most suitable for measuring sparsity, and which for measuring small parameter vectors?

Solution: For $\mathbf{x} = [0, 1, 0, 2, 3]$:

$$\|\mathbf{x}\|_0 = \#\{i : x_i \neq 0\} = 3,$$
$$\|\mathbf{x}\|_1 = \sum_i |x_i| = 1 + 2 + 3 = 6,$$
$$\|\mathbf{x}\|_2 = \sqrt{1^2 + 2^2 + 3^2} = \sqrt{14},$$
$$\|\mathbf{x}\|_\infty = \max_i |x_i| = 3.$$

- ℓ_0 measures the count of nonzero entries $\Rightarrow$ best for enforcing **sparsity**.
- ℓ_∞ measures the largest component $\Rightarrow$ controls the worst-case magnitude.
- ℓ_1 promotes sparsity but is convex $\Rightarrow$ easier to optimize than ℓ_0.
- ℓ_2 encourages small values across all coordinates, giving smooth solutions.

2. (4 points) Stochastic gradient descent (SGD) as an unbiased estimator.
   (a) Prove that SGD is an unbiased estimator of the true gradient in supervised learning with squared loss.
   (b) (Empirical check) Consider dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^3$ with

$$(x_1, y_1) = (1, 2), \quad (x_2, y_2) = (2, 3), \quad (x_3, y_3) = (3, 4).$$

We use squared loss $\ell_i(\theta) = \frac{1}{2}(y_i - x_i\theta)^2$ with scalar parameter θ. Compute the full gradient $\nabla J(\theta)$ and the individual gradients $\nabla \ell_i(\theta)$. Show that the expectation of a stochastic gradient step equals the batch gradient.

Solution:

(a) Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, squared loss:

$$\ell_i(\theta) = \tfrac{1}{2}(y_i - x_i^\top \theta)^2.$$

Full empirical risk:

$$J(\theta) = \tfrac{1}{N}\sum_{i=1}^N \ell_i(\theta), \quad \nabla J(\theta) = \tfrac{1}{N}\sum_{i=1}^N \nabla \ell_i(\theta).$$

SGD samples i uniformly and computes $g_i(\theta) = \nabla \ell_i(\theta)$. Expectation:

$$\mathbb{E}[g_i(\theta)] = \tfrac{1}{N}\sum_{i=1}^N \nabla \ell_i(\theta) = \nabla J(\theta).$$

Thus SGD is an unbiased estimator of the true gradient.

(b) For one point:

$$\nabla \ell_i(\theta) = -x_i(y_i - x_i\theta) = x_i^2\theta - x_i y_i.$$

Individual gradients:

$$\nabla \ell_1(\theta) = \theta - 2, \quad \nabla \ell_2(\theta) = 4\theta - 6, \quad \nabla \ell_3(\theta) = 9\theta - 12.$$

Full gradient:

$$\nabla J(\theta) = \tfrac{1}{3}\sum_{i=1}^3 (x_i^2\theta - x_i y_i) = \tfrac{1}{3}(14\theta - 20).$$

Expectation of SGD step:

$$\mathbb{E}[\nabla \ell_i(\theta)] = \tfrac{1}{3}\big((\theta - 2) + (4\theta - 6) + (9\theta - 12)\big) = \tfrac{1}{3}(14\theta - 20).$$

Same as full gradient. Empirical check confirms unbiasedness.

3. (3 points) Constrained gradient descent. We want to optimize a scalar parameter θ under the box constraint $\theta \in [-1, 1]$. One idea is to introduce an unconstrained parameter $\phi \in \mathbb{R}$ and map it to θ using a smooth function so that θ always lies in $[-1, 1]$.
   (a) Propose a suitable mapping from ϕ to θ using the sigmoid function.
   (b) Using this mapping, derive the gradient of the loss $J(\theta)$ w.r.t. ϕ.
   (c) Write down the explicit gradient descent update rule for $\phi^{(k+1)}$ with learning rate η.

Solution:

(a) A valid choice:

$$\theta = 2\sigma(\phi) - 1, \quad \sigma(\phi) = \frac{1}{1 + e^{-\phi}}.$$

This ensures $\theta \in (-1, 1)$.

(b) Chain rule:

$$\frac{\partial J}{\partial \phi} = \frac{\partial J}{\partial \theta} \cdot \frac{\partial \theta}{\partial \phi}.$$

$$\frac{\partial \theta}{\partial \phi} = 2\sigma(\phi)(1 - \sigma(\phi)).$$

(c) Update rule:

$$\phi^{(k+1)} = \phi^{(k)} - \eta \left(\frac{\partial J}{\partial \theta} \cdot 2\sigma(\phi^{(k)})(1 - \sigma(\phi^{(k)})) \right).$$

Then recover

$$\theta^{(k+1)} = 2\sigma(\phi^{(k+1)}) - 1.$$

4. (4 points) Focal Loss vs. Logistic Regression. In binary logistic regression, the cross-entropy (log loss) for a data point with label $y \in \{0, 1\}$ and predicted probability $p \in (0, 1)$ is

$$\ell_{\text{CE}}(p, y) = -\big(y \log p + (1 - y) \log(1 - p)\big).$$

The *focal loss* modifies this as

$$\ell_{\text{FL}}(p, y) = -(1 - p_t)^\gamma \log(p_t), \quad \text{where } p_t = \begin{cases} p & \text{if } y = 1, \\ 1 - p & \text{if } y = 0, \end{cases}$$

with tuning parameter $\gamma \geq 0$.

Definitions:

- An **easy example** is one that the model predicts with high confidence, e.g. $p_t \approx 1$.
- A **hard example** is one with low confidence, e.g. $p_t \ll 1$.

(a) Show that when $\gamma = 0$, focal loss reduces to the usual cross-entropy.
(b) For $\gamma > 0$, explain mathematically how the factor $(1 - p_t)^\gamma$ changes the loss for easy vs. hard examples.
(c) Suppose $y = 1$, $p = 0.99$. Compute CE loss and focal loss with $\gamma = 2$.
(d) Sketch $\ell_{\text{CE}}(p_t)$ and $\ell_{\text{FL}}(p_t)$ for $\gamma = 2$ as a function of $p_t \in [0, 1]$, and briefly explain the difference.

Solution:

(a) If $\gamma = 0$, then $(1 - p_t)^0 = 1$. So

$$\ell_{\text{FL}}(p, y) = -\log(p_t) = \ell_{\text{CE}}(p, y).$$

(b) For $\gamma > 0$: - Easy examples: $p_t \approx 1 \Rightarrow (1 - p_t)^\gamma \approx 0 \Rightarrow$ loss is down-weighted. - Hard examples: $p_t \ll 1 \Rightarrow (1 - p_t)^\gamma \approx 1 \Rightarrow$ loss is nearly unchanged. Thus focal loss emphasizes hard misclassified points.

(c) $y = 1$, $p = 0.99$: - CE: $\ell_{\text{CE}} = -\log(0.99) \approx 0.01005$. - FL with $\gamma = 2$: factor $(1 - 0.99)^2 = 0.0001$, so $\ell_{\text{FL}} \approx 0.0001 \times 0.01005 = 1.0 \times 10^{-6}$.

(d) Sketch: - CE loss $\ell_{\text{CE}}(p_t) = -\log(p_t)$ decreases smoothly from ∞ at $p_t = 0$ to 0 at $p_t = 1$. - FL loss for $\gamma = 2$ has the same shape but is strongly suppressed near $p_t \approx 1$, staying close to zero for easy examples. **Interpretation:** Focal loss reduces the effect of correctly classified points and focuses on misclassified ones.

5. (4 points) Matrix Factorization for Movie Recommendation Assume rank $k = 1$ with

$$U = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad V = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \quad \hat{R} = UV^\top = \begin{bmatrix} u_1 v_1 & u_1 v_2 \\ u_2 v_1 & u_2 v_2 \end{bmatrix},$$

and observed ratings

$$R = \begin{bmatrix} 5 & ? \\ ? & 4 \end{bmatrix}.$$

(a) Write the squared loss over observed entries.

(b) Derive $\frac{\partial J}{\partial u_1}, \frac{\partial J}{\partial v_1}, \frac{\partial J}{\partial u_2}, \frac{\partial J}{\partial v_2}$.

(c) With learning rate $\eta = 0.1$ and initialization

$$u_1 = 0.8,\ v_1 = 1.1, \quad u_2 = 1.3,\ v_2 = 0.9,$$

compute one simultaneous gradient descent update for (u_1, v_1, u_2, v_2).

(d) Compute the loss *before* and *after* the update. Has it decreased?

Solution:

(a) Loss:

$$J(u_1, u_2, v_1, v_2) = \frac{1}{2}\Big[(5 - u_1 v_1)^2 + (4 - u_2 v_2)^2\Big].$$

(b) Let $e_{11} = 5 - u_1 v_1,\ e_{22} = 4 - u_2 v_2$. Then

$$\frac{\partial J}{\partial u_1} = -(5 - u_1 v_1) v_1 = -e_{11} v_1, \quad \frac{\partial J}{\partial v_1} = -(5 - u_1 v_1) u_1 = -e_{11} u_1,$$

$$\frac{\partial J}{\partial u_2} = -(4 - u_2 v_2) v_2 = -e_{22} v_2, \quad \frac{\partial J}{\partial v_2} = -(4 - u_2 v_2) u_2 = -e_{22} u_2.$$

(c) Numerical step (simultaneous GD): first

$$e_{11} = 5 - (0.8)(1.1) = 4.12, \qquad e_{22} = 4 - (1.3)(0.9) = 2.83.$$

Gradients:

$$\partial_{u_1} J = -4.12 \cdot 1.1 = -4.532, \quad \partial_{v_1} J = -4.12 \cdot 0.8 = -3.296,$$

$$\partial_{u_2} J = -2.83 \cdot 0.9 = -2.547, \quad \partial_{v_2} J = -2.83 \cdot 1.3 = -3.679.$$

Updates ($x^+ = x - \eta\, \partial_x J$):

$$u_1^+ = 0.8 - 0.1(-4.532) = 1.2532, \quad v_1^+ = 1.1 - 0.1(-3.296) = 1.4296,$$

$$u_2^+ = 1.3 - 0.1(-2.547) = 1.5547, \quad v_2^+ = 0.9 - 0.1(-3.679) = 1.2679.$$

(d) Loss before:

$$J_{\text{before}} = \tfrac{1}{2}\big[(5 - 0.8 \cdot 1.1)^2 + (4 - 1.3 \cdot 0.9)^2\big] = \tfrac{1}{2}\big[4.12^2 + 2.83^2\big] \approx 12.49165.$$

Loss after (using updated values):

$$J_{\text{after}} = \tfrac{1}{2}\big[(5 - 1.2532 \cdot 1.4296)^2 + (4 - 1.5547 \cdot 1.2679)^2\big] \approx 7.2050.$$

Yes, the loss decreased (from ≈ 12.49 to ≈ 7.21).