Homework — Linear Regression by hand (Week 2)

Course: Intro to Machine Learning
Format: Hand calculations only (you may use a basic calculator for arithmetic). Show **ALL** steps.
Deliverable: A single PDF with legible, step-by-step work (scan or photos).

Learning objectives

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

1. Set up the linear model with an intercept and write the MSE cost.
2. Derive and solve the normal equation for 1-feature regression without explicitly inverting a matrix.
3. Compute predictions, residuals, and MSE.
4. Execute one step of gradient descent (full-batch) and one LMS/SGD update (single example) using the gradient from the normal-equation derivation.
5. Explain why feature scaling helps gradient-based optimization.
6. Recognize when the normal-equation system is singular/ill-conditioned and why naive inversion is risky.
7. (*Optional*) Connect maximum likelihood (Gaussian noise) to least squares.

Notation. Include an intercept. Let $\boldsymbol{\theta} = [\theta_0,\ \theta_1]^\top$ with model $\hat{y} = \theta_0 + \theta_1 x$.

Dataset (shared for all parts)

Learning the trend of reported car accidents for a new motorcycle model. Use the following 5 data points:

i	x (years)	y (accidents)
1	0	1
2	1	3
3	2	2
4	3	5
5	4	4

Sanity checks you may use: $\sum x = 10$, $\sum x^2 = 30$, $\sum y = 15$, $\sum(x\,y) = 38$, $\bar{x} = 2$, $\bar{y} = 3$.

Part A — Normal equation (by hand) (30 points)

1) Form the design matrix $\mathbf{X} = [\mathbf{1}\ \ \mathbf{x}] \in \mathbb{R}^{5\times 2}$ and vector $\mathbf{y} \in \mathbb{R}^5$.

2) Write the MSE:

$$J(\boldsymbol{\theta}) = \frac{1}{n} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2.$$

Take the derivative and set to zero to obtain the normal equation:

$$(\mathbf{X}^\top \mathbf{X})\, \boldsymbol{\theta} = \mathbf{X}^\top \mathbf{y}.$$

(Show the key steps — component-wise or using matrix calculus.)

3) Compute $\mathbf{X}^\top \mathbf{X}$ and $\mathbf{X}^\top \mathbf{y}$ numerically from the dataset.

4) Solve the 2×2 linear system for $\boldsymbol{\theta} = [\theta_0, \theta_1]^\top$ by hand using Gaussian elimination (preferred) or the 2×2 inverse.

5) Write the fitted line $\hat{y} = \theta_0 + \theta_1 x$.

Part B — Predictions & MSE (20 points)

1) Compute fitted values $\hat{y}_i$ for each i and fill the table:

i	x	y	$\hat{y}_i$	residual $r_i = y_i - \hat{y}_i$
1	0	1		
2	1	3		
3	2	2		
4	3	5		
5	4	4		

2) Compute MSE $= \frac{1}{n} \sum_i r_i^2$.

3) In 2–3 sentences, interpret the slope and intercept in plain language.

4) Identify which point has the largest $|r_i|$ and give one possible explanation (outlier? model misspecification? small n?).

Part C — Gradient Descent & LMS updates (30 points)

1) **Use the result from Part A.2.** Without re-deriving, use the gradient obtained in the normal-equation derivation:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \frac{2}{n} \mathbf{X}^\top (\mathbf{X}\boldsymbol{\theta} - \mathbf{y}).$$

Proceed with the following computations.

2) **Full-batch GD step.** With learning rate $\alpha = 0.1$ and initial $\boldsymbol{\theta}^{(0)} = [0, 0]^\top$, compute *one* gradient descent update by hand:

$$\boldsymbol{\theta}^{(1)} = \boldsymbol{\theta}^{(0)} - \alpha \nabla J(\boldsymbol{\theta}^{(0)}).$$

3) **LMS/SGD step (single example).** Let $\tilde{\mathbf{x}}_i = [1,\ x_i]^\top$ and $\hat{y}_i = \boldsymbol{\theta}^\top \tilde{\mathbf{x}}_i$. Use the per-example gradient

$$\nabla_{\boldsymbol{\theta}} \ell_i = 2(\hat{y}_i - y_i)\, \tilde{\mathbf{x}}_i.$$

Using example $i = 3$ ($x = 2$, $y = 2$) and $\boldsymbol{\theta} = [0, 0]^\top$, perform *one* LMS update with $\alpha = 0.1$:

$$\boldsymbol{\theta} \ \leftarrow\ \boldsymbol{\theta} - \alpha\, \nabla \ell_3.$$

4) Do a **second** update — you may choose GD **or** LMS (or compute both) — and briefly compare your result with the closed-form solution from Part A.
(hint: compare at least one of the following: parameter error $\|\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}\|_2$, the objective $J(\boldsymbol{\theta}^{(k)})$, or the gradient norm $\|\nabla J(\boldsymbol{\theta}^{(k)})\|_2$. Comment on trade-offs: GD tends to be smooth/monotonic on this quadratic with a suitable α; LMS has noisier steps but cheaper per-update and supports streaming data; the closed-form gives the exact solution but requires solving a linear system.)

Part D — Feature scaling & GD stability (10 points)

1) Compute the standardized feature $z = (x - \bar{x})/s_x$, where s_x is the sample standard deviation of x.

2) Redo *one* full-batch GD step from $\boldsymbol{\theta} = [0, 0]^\top$ using $[1,\ z]$ as the feature vector. Compare the gradient magnitudes with the unscaled case.

3) **Why Scale Features?** In 3–5 sentences, explain why scaling features makes gradient descent faster and more stable. Define *normalization* (rescale to $[0, 1]$) and *standardization* (mean 0, variance 1). Describe what happens to GD updates when features have very different ranges (e.g., the zig-zag effect) and how scaling changes the shape of cost contours. You may use a brief analogy (e.g., house area in m^2 vs. number of bedrooms).
(hint: normalization → values in $[0, 1]$; standardization → mean 0, variance 1; scaling prevents gradients from zig-zagging across narrow valleys; makes contours more circular/balanced; greatly improves convergence speed and stability.)

Part E — Concept: When to use closed-form vs. gradient methods (10 points)

1) For the given dataset, compute $\det(\mathbf{X}^\top \mathbf{X})$. Is the system safely invertible?

2) Consider a variant dataset where all x's are the same (e.g., all $x = 2$). Write $\mathbf{X}^\top \mathbf{X}$ and explain why the normal equation cannot be solved uniquely.

3) In practice, when would you **prefer** the normal-equation (closed-form) solution and when would you **prefer** GD/LMS? Give two concrete considerations (e.g., dataset/feature size, need for online updates, memory limits, ease of tuning).
(hint: closed-form is exact and fast for small problems but involves solving a linear system; GD/LMS scales to large/streaming data with low per-step cost but needs learning-rate/tolerance choices and multiple iterations.)

(Optional) Part F — Probabilistic interpretation (MLE $\leftrightarrow$ least squares) (+10 points)

Assume $y_i \mid x_i \sim \mathcal{N}(\theta_0 + \theta_1 x_i,\ \sigma^2)$ i.i.d.

1) Write the log-likelihood $\log p(\mathbf{y} \mid \boldsymbol{\theta}, \sigma^2)$.
2) Show that the negative log-likelihood equals a constant plus $\frac{1}{2\sigma^2}\sum_i (y_i - \hat{y}_i)^2$.
3) Conclude that maximizing the likelihood over $\boldsymbol{\theta}$ is equivalent to minimizing SSE (least squares).

What to submit & formatting

- One PDF with: each part on a separate page, steps clearly labeled.
- Show intermediate sums/products (e.g., $\sum(x\,y)$, $\sum x^2$) and linear system steps.
- If you used a calculator, round only at the **last** step (≥ 4 decimal places).

Grading rubric (100 pts + optional 10)

- **A. Normal equation (by hand)** — 30 pts: correct derivation and solution of $\boldsymbol{\theta}$ with clear steps (no hand-wavy inversion).
- **B. Predictions & MSE** — 20 pts: correct $\hat{y}$, residuals, MSE; coherent interpretation.
- **C. GD & LMS** — 30 pts: correct use of the gradient from Part A.2; correct one GD step and one LMS step.
- **D. Feature scaling** — 10 pts: correct z-score, GD step comparison, and explanation.
- **E. Closed-form vs. GD/LMS** — 10 pts: clear trade-offs with two concrete considerations.
- **Optional: F. MLE connection** — +10 pts.

Academic integrity & submission

This assignment is **not** a group project; each student must complete and submit their **own** work. Upload a single PDF to the **E3** system. **Deadline: 9/28 (Sun) at 11:59 AM** (Taipei time, UTC+8). **Late submissions receive no credit.**