CSCI 6751 - Formula Reference Guide

Artificial Intelligence | Fall 2025

1. Gradient Descent

Core Update Rule

> $$\theta_{new} = \theta_{old} - \eta \nabla J(\theta)$$
>
> **Where:**
>
> - η (eta) = Learning rate
> - ∇J = Gradient (derivative of loss function)

Simple Linear Regression (y = ax + b)

> **Loss Function (MSE):**
>
> $$J(a, b) = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$$
>
> **Gradients:**
>
> $$\frac{\partial J}{\partial a} = \frac{2}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i) \cdot x_i$$
>
> $$\frac{\partial J}{\partial b} = \frac{2}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)$$
>
> **Parameter Updates:**
>
> $$a_{new} = a_{old} - \eta \cdot \frac{\partial J}{\partial a}$$
>
> $$b_{new} = b_{old} - \eta \cdot \frac{\partial J}{\partial b}$$

Algorithm Steps

> 1. **Compute predictions:** $\hat{y}_i = a \cdot x_i + b$
> 2. **Calculate errors:** $e_i = \hat{y}_i - y_i$
> 3. **Compute gradient for a:** $\partial J/\partial a = (2/n)\Sigma(e_i \cdot x_i)$

> 4. **Compute gradient for b:** ∂J/∂b = (2/n)Σ(e_i)
> 5. **Update a:** $a_{ne}w$ = $a_{ol}d$ - η·∂J/∂a
> 6. **Update b:** $b_{ne}w$ = $b_{ol}d$ - η·∂J/∂b

Multivariate Linear Regression

> **Model:** $y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_p x_p$
>
> **Gradients:**
>
> $$\frac{\partial J}{\partial \theta_0} = \frac{2}{n} \sum (\hat{y}_i - y_i)$$
>
> $$\frac{\partial J}{\partial \theta_j} = \frac{2}{n} \sum (\hat{y}_i - y_i) \cdot x_{ji} \quad (j = 1, 2, \ldots, p)$$

2. L2 Regularization (Ridge Regression)

Regularized Cost Function

> $$J_{Ridge} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{p}\theta_j^2$$
>
> **Note:** Typically, θ₀ (intercept) is not regularized.

Gradient with L2 Regularization

> $$\frac{\partial J}{\partial \theta_j} = \frac{2}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)\cdot x_{ji} + 2\lambda\theta_j$$

Effect of Lambda (λ)

Lambda Value	Effect
λ = 0	No regularization (standard regression)
Small λ	Weak penalty, potential overfitting
Medium λ	Balanced, optimal performance
Large λ	Strong penalty, potential underfitting

3. Normal Equation (Closed-Form Solution)

Core Formula

> $$\theta = (X^T X)^{-1} X^T y$$
>
> **Where:**
>
> - X = Design matrix (first column is all 1s for intercept)
> - y = Target vector
> - θ = Parameter vector $[\theta_0, \theta_1, \ldots, \theta_p]$

Computation Steps

> 1. Construct design matrix X (add column of 1s)
> 2. Compute $X^T X$
> 3. Compute $(X^T X)^{-1}$
> 4. Compute $X^T y$
> 5. Multiply to obtain $\theta = (X^T X)^{-1} X^T y$

2×2 Matrix Inversion

> **Given:**
>
> $$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$
>
> **Determinant:**
>
> $$\det(A) = ad - bc$$
>
> **Inverse:**
>
> $$A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

> **Memory aid:** Swap diagonal, negate off-diagonal, divide by determinant.

When to Use Normal Equation vs. Gradient Descent

Method	When to Use	Advantages	Disadvantages
Normal Equation	Features < 1000	Direct solution, no iterations	Requires matrix inversion (slow for large p)
Gradient Descent	Features > 1000	No inversion needed, scalable	Requires multiple iterations, tuning η

4. Fuzzy Logic

Triangular Membership Function

Notation: triangular(a, b, c)

```
μ
1 |      /\
  |     /  \
  |    /    \
  |   /      \
0 |__/________\__
    a    b    c
```

Formula:

$$\mu(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a < x \leq b \\ \frac{c-x}{c-b} & b < x < c \\ 0 & x \geq c \end{cases}$$

Key points: a = left boundary, b = peak (μ=1), c = right boundary

Trapezoidal Membership Function

Notation: trapmf(a, b, c, d)

```
μ
1 |    ____
  |   /    \
  |  /      \
  | /        \
0 |/          \
    a  b  c  d
```

Formula:

$$\mu(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a < x < b \\ 1 & b \leq x \leq c \\ \frac{d-x}{d-c} & c < x < d \\ 0 & x \geq d \end{cases}$$

Key points: [a,b] = rising edge, [b,c] = plateau (μ=1), [c,d] = falling edge

Fuzzy Inference System (Mamdani)

Four Steps:

1. **Fuzzification:** Convert crisp inputs to membership degrees
2. **Rule Evaluation:** Compute firing strength for each rule
3. **Aggregation:** Combine outputs from all rules
4. **Defuzzification:** Convert fuzzy output to crisp value

Firing Strength (AND Operation)

Rule form: IF X is A AND Y is B THEN Z is C

Firing Strength (MIN operator):

$$\text{FS} = \min(\mu_A(x), \mu_B(y))$$

Rationale: AND requires both conditions; take the weaker of the two.

Centroid Defuzzification

Weighted average method:

$$\text{Output} = \frac{\sum_{i=1}^{n} (\text{FS}_i \times \text{Output}_i)}{\sum_{i=1}^{n} \text{FS}_i}$$

Where:

> - FS_i = Firing strength of rule i
> - $Output_i$ = Crisp output value of rule i

5. Overfitting and Underfitting

Definitions

Condition	Training Error	Test Error	Cause
Underfitting	High	High	Model too simple
Good Fit	Low	Low (≈ Train)	Optimal complexity
Overfitting	Very low	High (>> Train)	Model too complex

Solutions

> **To reduce overfitting:**
>
> - Increase λ (regularization strength)
> - Decrease polynomial degree
> - Collect more training data
> - Apply early stopping
>
> **To reduce underfitting:**
>
> - Decrease λ
> - Increase polynomial degree
> - Add more features

Hyperparameters

Hyperparameter	Role	Typical Values
Learning Rate (η)	Step size in gradient descent	0.001 to 0.1
Polynomial Degree	Model complexity	1 to 10
Lambda (λ)	Regularization strength	0.001 to 100

> **Note:** Hyperparameters are not learned from data; they must be tuned via cross-validation.

6. Additional Key Concepts

Classification vs. Regression

Task Type	Output	Examples
Regression	Continuous values	House prices, temperature
Classification	Discrete categories	Cat vs. dog, spam detection

Supervised vs. Unsupervised Learning

Learning Type	Characteristics	Examples
Supervised	Labeled data available	Price prediction, image classification
Unsupervised	No labels	Customer segmentation, dimensionality reduction

Fuzzy vs. Classical Logic

Logic Type	Value Range	Example
Classical	Binary (0 or 1)	True / False
Fuzzy	Continuous [0, 1]	0.7 (somewhat true)

7. Common Errors to Avoid

Gradient Descent

- Computing error as $y - \hat{y}$ instead of $\hat{y} - y$
- Forgetting to divide by n (number of samples)
- Using addition instead of subtraction in parameter update
- Forgetting to multiply by learning rate η
- Omitting multiplication by x_i when computing $\partial J/\partial a$

Normal Equation

- Attempting matrix multiplication with incompatible dimensions
- Computing determinant as $ad + bc$ instead of $ad - bc$
- Failing to swap diagonal elements in matrix inversion

Fuzzy Logic

- Misidentifying which region x falls into (rising/plateau/falling)
- Using MAX for AND operations (should use MIN)
- Errors in centroid numerator/denominator calculation
- Forgetting that trapezoidal plateau region has $\mu = 1$

8. Exam Strategy

Time Management (50-minute exam)

- Reading and planning: 3-5 minutes
- Question 1: 20-22 minutes
- Question 2: 20-22 minutes
- Review: 3-5 minutes

Answering Techniques

- Show all steps clearly (partial credit for correct methodology)
- Double-check signs (especially negative signs in gradients)
- Verify dimensions in matrix operations
- If stuck, move on and return later
- Use pencil for easy corrections

CSCI 6751 Artificial Intelligence | Fall 2025

End of Formula Reference Guide