Deep Neural Networks

Programming Assignment

Course: Deep Neural Networks **Assignment**
Type: Group Programming

Maximum Marks: 10

Format: Jupyter Notebook (.ipynb)

1. Learning Objectives

By completing this assignment, you will:

□ Understand regression and classification from first principles
□ Implement gradient descent optimization
□ Build neural networks without high-level libraries
□ Evaluate and compare machine learning models
□ Analyze computational trade-offs in model selection

2. Dataset
Use the following links to collect the datasets

Regression

- https://www.kaggle.com/c/house-prices-advanced-regression-techniques
- https://www.kaggle.com/datasets/mirichoi0218/insurance
- https://www.kaggle.com/datasets/whenamancodes/student-performance

Classification

- https://www.kaggle.com/datasets/ronitf/heart-disease-uci
- https://www.kaggle.com/datasets/mathchi/diabetes-data-set
- https://www.kaggle.com/datasets/altruistdelhite04/loan-prediction-problem-dataset

What You MUST Implement from Scratch

For Regression:

– Linear Regression with batch gradient descent

For Binary Classification:

– Logistic Regression with SGD gradient descent

Allowed Libraries: You MAY use:

NumPy – Array operations

Pandas – Data loading and manipulation

Matplotlib/Seaborn – Visualization

sklearn – ONLY for:

- train test split
- StandardScaler / MinMaxScaler
- LabelEncoder / OneHotEncoder

Prohibited Libraries: You CANNOT use:

sklearn.linear model (any models)

sklearn.neural network (MLPClassifier, MLPRegressor)

tensorflow, keras, pytorch, jax

Any high-level ML library for models

WARNING

Penalty for prohibited libraries: -5 marks

Assignment Structure

A. Linear Neural Network for Regression (5 Marks)

Problem Statement

- As a group, design and implement a linear neural network with a single output neuron to predict a continuous target variable.
- Each group must choose one Kaggle regression dataset different from other groups.

Model Description

- Output equation: $\hat{y} = wx + b$
- Loss function: Mean Squared Error (MSE) $L = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
- Optimization method: Batch Gradient Descent

Group Tasks

1. Initialize weight $w$and bias b.
2. Perform forward pass to compute predictions.
3. Calculate MSE loss.
4. Derive gradients of loss with respect to $w$and b.
5. Update parameters using gradient descent.
6. Train the model for a fixed number of epochs.
7. Display final parameters and prediction results.

B. Linear Neural Network for Classification (5 Marks)

Problem Statement

- Implement a linear neural network with a single output neuron to perform binary classification**.**
- Each group must select one Kaggle binary classification dataset**.**

Model Description

- Net input: $z = wx + b$
- Activation function:
  - Sigmoid**:** $\sigma(z) = \frac{1}{1+e^{-z}}$
- Decision rule: $\hat{y} = 1$ if $\sigma(z) \geq 0.5$, 0 otherwise
- Loss function: Binary Cross-Entropy

Group Tasks

1. Initialize weights and bias.
2. Compute net input and activated output.
3. Calculate loss for classification.
4. Update parameters using gradient descent.
5. Apply threshold to obtain class labels.
6. Report predicted vs actual classes.

Submission Requirement

Each group must submit

1. Source code file or Jupyter Notebook
2. Output screenshots
3. Short group report (2–3 pages) containing:
   - Problem understanding
   - Algorithm explanation
   - Results and observations
4. Contribution of each member

Clean Notebook

Before submitting:

1. Kernel Restart & Clear Output
2. Kernel Restart & Run All
3. Verify all cells execute without errors
4. File Save and Checkpoint

File Naming **Format:** YourGroupID_assignment.ipynb

Submit to LMS

5. Upload .ipynb and PDF file to assignment portal
6. Verify upload successful

Academic Integrity Policy

- Code sharing between groups is not permitted.
- All submissions will be reviewed for similarity.
- Equal marks awarded to all group members unless contribution issues are identified.