Deep Learning: Principles and Practices

Lab 4

Supervised Learning – Regression (ML + Deep Learning)

Dataset: insurance.csv

Objectives

• Understand various **machine learning and deep learning approaches** for regression problems.
• Implement and compare **traditional ML regression models** with **Artificial Neural Networks (ANN)**.
• Evaluate model performance using **MAE, MSE, RMSE, and R^2**.
• Analyze **bias–variance tradeoff** across ML and DL models.
• Understand how regression problems can be solved using **both ML and DL paradigms**.

Theory to Be Read Before Lab

Machine Learning (Revision)

• Regression types: Simple, Multiple, Polynomial
• Evaluation Metrics: MAE, MSE, RMSE, R^2
• Train–test split rationale
• Bias and Variance concepts
• Ridge and Lasso regularization
• Decision Tree regression overview

Deep Learning (New)

• Artificial Neural Network (ANN) architecture
• Feed-forward computation
• Loss functions for regression (MSE)
• Backpropagation and Gradient Descent (conceptual)
• Role of optimizers (SGD, Adam – overview)
• Effect of epochs, batch size, and learning rate

Exercise 1: Dataset Understanding and Preprocessing

1. Load the dataset and display .head(), .info(), and .describe().
2. Identify categorical and numerical features.
3. Handle missing values if any.
4. Encode categorical features using:
   • Label Encoding
   • One-Hot Encoding
5. Normalize/standardize numerical features.
6. Split the dataset into training and testing sets (80:20).

Exercise 2: Apply Machine Learning Regression Models

Apply the following regression models to predict **charges**:
• Simple Linear Regression (using bmi only)
• Multiple Linear Regression (all features)
• Polynomial Regression (degree 2 and 3)
• Ridge Regression (with tuning alpha)
• Lasso Regression (with tuning alpha)
• Decision Tree Regressor

For **each model**:
• Train the model
• Predict on test data
• Evaluate and record:
o MAE
o MSE
o RMSE
o R^2 Score
• Create scatter plot of **Actual vs Predicted values**

Exercise 3: Artificial Neural Network (ANN) for Regression

Step 1: ANN Model Design

Design a **Feed-Forward Neural Network** for regression with:

- Input layer: number of neurons = number of features
- Hidden layer(s): 1 or 2 hidden layers
- Activation function:
    - Hidden layer: ReLU
    - Output layer: Linear activation
- Loss function: Mean Squared Error (MSE)
- Optimizer: Adam (default parameters)

Sample ANN Code (Reference Only)

(This is only a skeleton to help you get started. Students must complete preprocessing, training, and evaluation.)

```
from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense


model = Sequential()

model.add(Dense(16, activation='relu', input_shape=(n_features,)))

model.add(Dense(8, activation='relu'))

model.add(Dense(1, activation='linear'))


model.compile(optimizer='adam', loss='mse')
```

Step 2: Model Training

- Compile the ANN model
- Train using training data

• Use validation split (e.g., 10–20%)
• Train for suitable epochs (e.g., 100)

Step 3: Model Evaluation

• Predict on test data
• Compute:
o MAE
o MSE
o RMSE
o R^2 Score
• Plot:
o Training vs Validation loss curve
o Actual vs Predicted scatter plot

Exercise 4: Model Comparison – ML vs DL

Step 1: Create Evaluation Tables

Table A: Training Data

Model	**MAE**	**MSE**	**RMSE**	**R^2**
Simple Linear Regression				
Multiple Linear Regression				
Polynomial Regression				
Ridge Regression				
Lasso Regression				
Decision Tree Regressor				
ANN Regressor				

Table B: Testing Data

Model	MAE	MSE	RMSE	R^2
Simple Linear Regression				
Multiple Linear Regression				
Polynomial Regression				
Ridge Regression				
Lasso Regression				
Decision Tree Regressor				
ANN Regressor				

Exercise 5: Analysis and Interpretation

• Compare ML models vs ANN performance
• Identify overfitting and underfitting
• Discuss impact of:
    o Model complexity
    o Regularization
    o Number of epochs (ANN)
• Explain why ANN can model **non-linear relationships** better
• Comment on computational cost vs performance

Learning Outcome (Very Important)

At the end of this lab, students should understand that:
• Regression problems can be solved using **both ML and DL models**
• ANN is a **generalized non-linear regression model**
• Model selection depends on:
    o Data size
    o Complexity
    o Interpretability
    o Performance requirements

Submission Guidelines

• Submit Jupyter notebook as PDF
• Each exercise must include:
  o Problem statement
  o Output
  o Brief comments
• Header on each page:
  ECSCI24302 Machine Learning Essentials
• Footer:
  [Lab 3] [Page Number] [Enrollment Number]

Knowledge Check Questions (Answer any 5)

1. Why is linear activation used in the output layer of ANN for regression?
2. How does ANN differ from Polynomial Regression?
3. What role does optimizer play in ANN training?
4. Why does ANN require feature scaling?
5. Compare overfitting behavior of Decision Tree and ANN.
6. When would ML regression models be preferred over ANN?
7. How does batch size affect ANN training?