MLDL Practical 7

Name: Ishaan Khan
Class: D15C
Roll No: 29
Batch: B

Aim: Build an Artificial Neural Network (ANN) using Keras/TensorFlow

Dataset Source

Dataset Name: Heart Disease Dataset
Platform: Kaggle
Dataset Link:
https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

Dataset Description

The Heart Disease dataset is a binary classification dataset used to predict the presence of heart disease based on various medical attributes.

Dataset Characteristics

- Number of instances: 1,025
- Number of features: 13 numerical features
- Target Variable: target
  - 1 → Presence of heart disease
  - 0 → Absence of heart disease

Important Features

- age – Age of patient
- sex – Gender
- cp – Chest pain type
- trestbps – Resting blood pressure
- chol – Serum cholesterol
- thalach – Maximum heart rate achieved
- oldpeak – ST depression induced by exercise

Since the features have different scales, feature scaling is mandatory for proper ANN convergence.

Mathematical Formulation

An Artificial Neural Network consists of:

- Input Layer

- Hidden Layer(s)
- Output Layer

Forward Propagation

For each neuron:

$$\mathbf{z = w^T x + b}$$
$$\mathbf{a = \sigma(z)}$$

Where:

- w = weight vector
- x = input features
- b = bias
- σ = activation function

Hidden layers use **ReLU activation**:

ReLU(z) = max(0, z)

Output layer uses **Sigmoid activation**:

$\sigma(z) = 1 / (1 + e^{-z})$

The sigmoid function converts outputs into probabilities between 0 and 1.

Loss Function – Binary Cross Entropy

Since heart disease prediction is a binary classification problem, Binary Cross-Entropy (BCE) loss is used.

For N samples, the cost function is:

$$\mathbf{L = -(1/N)\ \Sigma\ [\ y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)\ ]}$$

Where:

- y_i = actual label
- $\hat{y}_i$ = predicted probability
- N = total number of samples

This loss function penalizes incorrect predictions and is minimized using backpropagation and gradient descent.

Algorithm Limitations

- Neural networks can overfit small datasets
- Requires proper feature scaling
- Computationally more expensive than traditional algorithms
- Difficult to interpret (Black-box model)

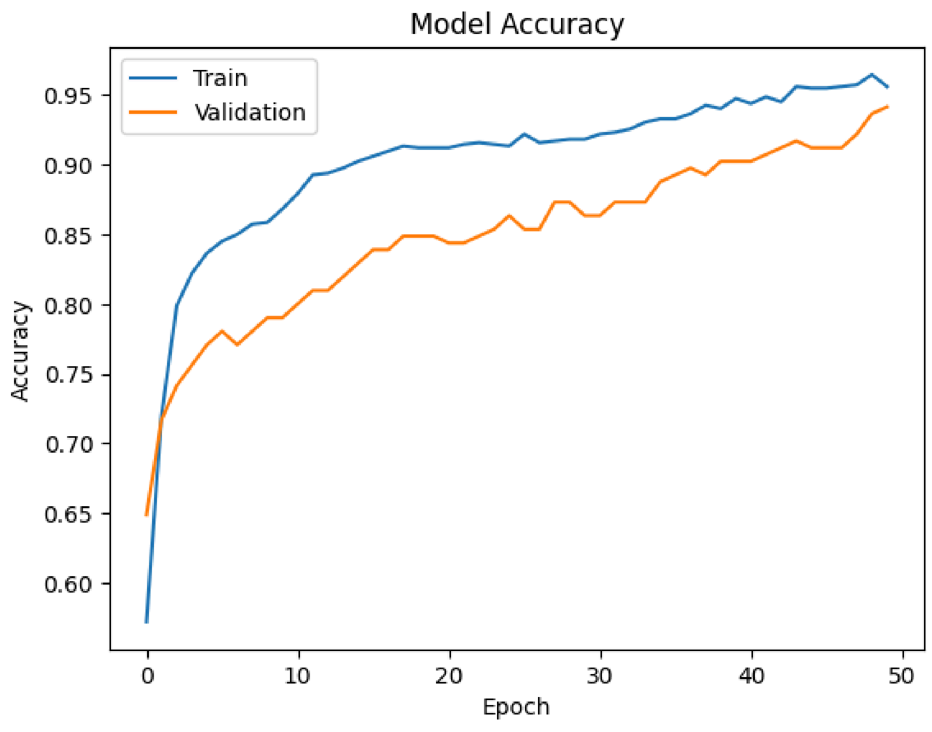
Methodology / Workflow

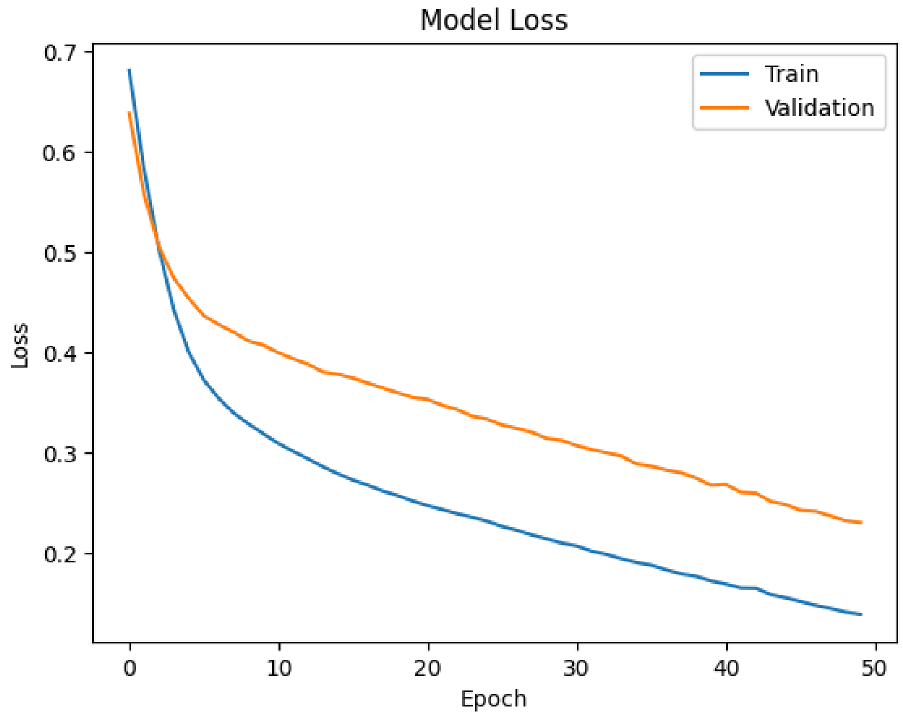
1. Load dataset using KaggleHub
2. Split features and target variable
3. Perform train-test split
4. Apply StandardScaler
5. Build ANN using Sequential model
6. Add Dropout layer for regularization
7. Train model using Adam optimizer
8. Evaluate using Confusion Matrix and ROC-AUC

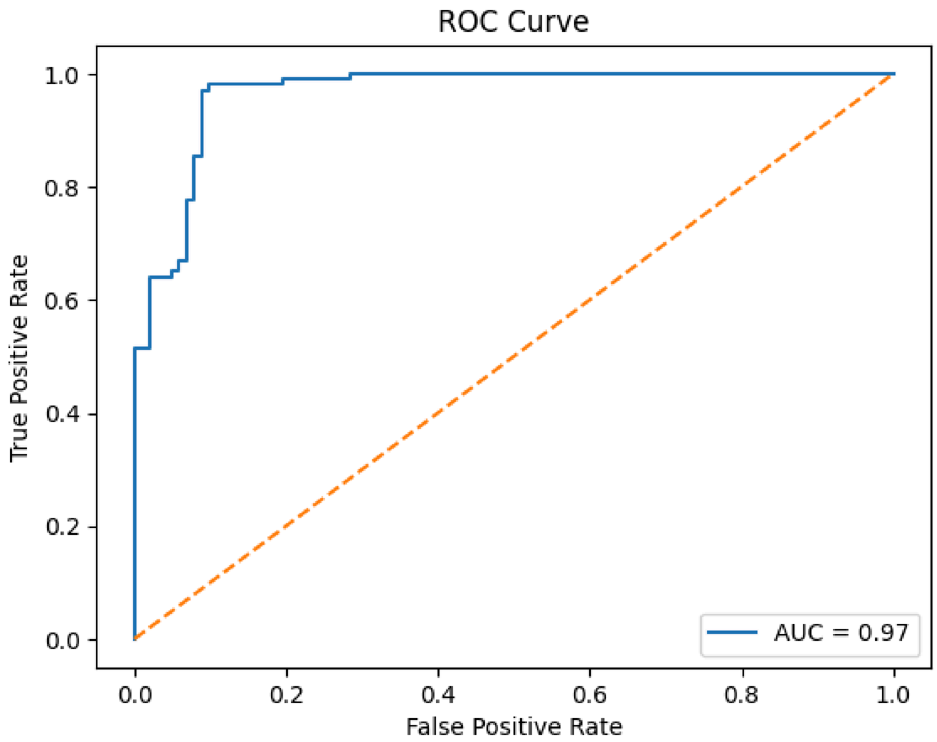
Performance Analysis

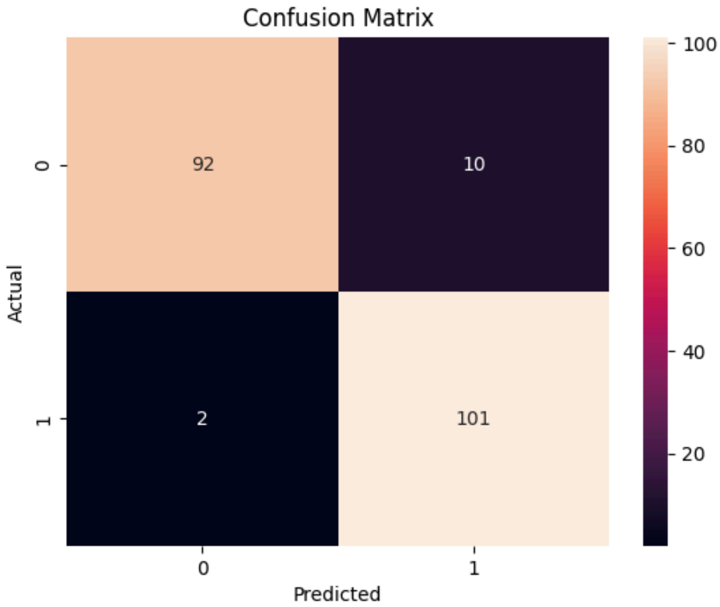
- The ANN achieved high classification accuracy.
- Dropout reduced overfitting by improving validation stability.
- ROC-AUC score demonstrated strong predictive capability.
- The confusion matrix showed low false negatives, which is critical in medical diagnosis.

Output




Model Loss
Train
Validation
0.7
0.6
0.5
0.4
0.3
0.2
Loss
0
10
20
30
40
50
Epoch


ROC Curve
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
AUC = 0.97
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate




```
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.90      0.94       102
           1       0.91      0.98      0.94       103

    accuracy                           0.94       205
   macro avg       0.94      0.94      0.94       205
weighted avg       0.94      0.94      0.94       205
```

Conclusion

In this experiment, an Artificial Neural Network was successfully implemented for heart disease prediction.

Feature scaling, ReLU activation, sigmoid output, and dropout regularization played a crucial role in achieving stable training and strong generalization performance.

This experiment demonstrates how deep learning models can effectively capture complex non-linear relationships in medical datasets and assist in predictive healthcare systems.