Department of Computer Engineering
BE Computer-B (2025-26 Sem I)
LP-III Machine Learning
Practical Assignment 3: Neural network-based classifier

[CO1, CO3, BT: L3 (Apply)] [Max Marks: 10]

Problem Definition: Given a bank customer, build a neural network-based classifier that can determine whether they will leave or not in the next 6 months.

Dataset Description: The case study is from an open-source dataset from Kaggle. The dataset contains 10,000 sample points with 14 distinct features such as CustomerId, CreditScore, Geography, Gender, Age, Tenure, Balance, etc.

Link to the Kaggle project:

https://www.kaggle.com/barelydedicated/bank-customer-churn-modeling

Perform following steps:

1. Read the dataset.

2. Distinguish the feature and target set and divide the data set into training and test sets.

3. Normalize the train and test data.

4. Initialize and build the model. Identify the points of improvement and implement the same.

5. Print the accuracy score and confusion matrix (5 points).

Learning Outcomes:

By completing this practical, students will be able to:

Learning Outcome	**Bloom's Taxonomy Level**
Understand the concept of customer churn and classification problems	Remember (L1)
Preprocess and normalize numerical and categorical data	Apply (L3)
Build, train, and evaluate a neural network for classification	Apply (L3)
Interpret confusion matrix and accuracy score	Analyze (L4)
Identify model improvement strategies and implement them	Evaluate (L5)
Conclude findings based on results and model performance	Create (L6)

Software / Hardware Requirements

Software:

- Python 3.8+
- Jupyter Notebook / Google Colab
- Libraries: numpy, pandas, matplotlib, seaborn, scikit-learn, tensorflow/keras

Hardware:

- Minimum 4 GB RAM
- Dual-core processor
- Stable internet connection (for data access)
- Recommended: Quad-core CPU, 8GB RAM, GPU support for faster training

Theory:

Customer Churn Prediction:

Churn refers to customers leaving a bank or company. In the banking sector, churn prediction helps in retaining valuable customers by identifying at-risk customers early.

1. Dataset Description:

The case study is from an open-source dataset from Kaggle. The dataset contains 10,000 sample points with 14 distinct features such as CustomerId, CreditScore, Geography, Gender, Age, Tenure, Balance, etc.

Acknowledgement:

The dataset is referred from Kaggle: https://www.kaggle.com/barelydedicated/bank-customer-churn-modeling

2. Artificial Neural Network (ANN)

An **Artificial Neural Network (ANN)** is a machine learning model inspired by the human brain's structure and functioning. It is composed of layers of interconnected **neurons** that process data in multiple stages.

Key Components:

- **Input Layer:** Receives the input features.
- **Hidden Layers:** Perform transformations using **weights**, **biases**, and **activation functions** to learn complex patterns.
- **Output Layer:** Produces the final prediction (in churn prediction, 0 = stays, 1 = leaves).

Neuron Function: $z=W\cdot X + b$

Where:W = weights

- X = inputs
- b = bias

The neuron applies an **activation function** $f(z)$ to introduce non-linearity: $a=f(z)$

For binary classification, the **Sigmoid** function is used: $\sigma(z)=\{1\}/\{1 + e^{-z}\}$

3. Keras

Keras is a high-level **deep learning API** written in Python that allows quick and easy building of neural networks.

It runs on top of **TensorFlow** and supports rapid prototyping.

Advantages:

- Simple and user-friendly syntax.
- Modular design for building networks layer-by-layer.
- Supports both CPU and GPU execution.
- Pre-built layers like Dense, Dropout, Conv2D for different types of networks.

4. TensorFlow

TensorFlow is an open-source deep learning framework developed by Google Brain. It is a **backend engine** for numerical computation and deep learning model training.

Features:

- Supports large-scale machine learning.
- Automatic differentiation for gradient computation.
- Cross-platform execution: CPU, GPU, TPU.
- Works with Keras for high-level modeling.

5. Normalization

Normalization ensures that all numerical features are on a similar **scale** to help the model train faster and perform better. Without normalization, features with larger values dominate those with smaller values, leading to biased learning.

Common Methods:

- **Min-Max Scaling:**

$$x' = \{x - x_{min}\}/\{x_{max} - x_{min}\}$$

- **Standardization (Z-score scaling):**

$$x' = x - \mu / \sigma$$

Where: μ = mean of the feature and σ = standard deviation

6. Confusion Matrix

A **confusion matrix** is a table used to describe the performance of a classification model. It compares **predicted labels** with **actual labels**.

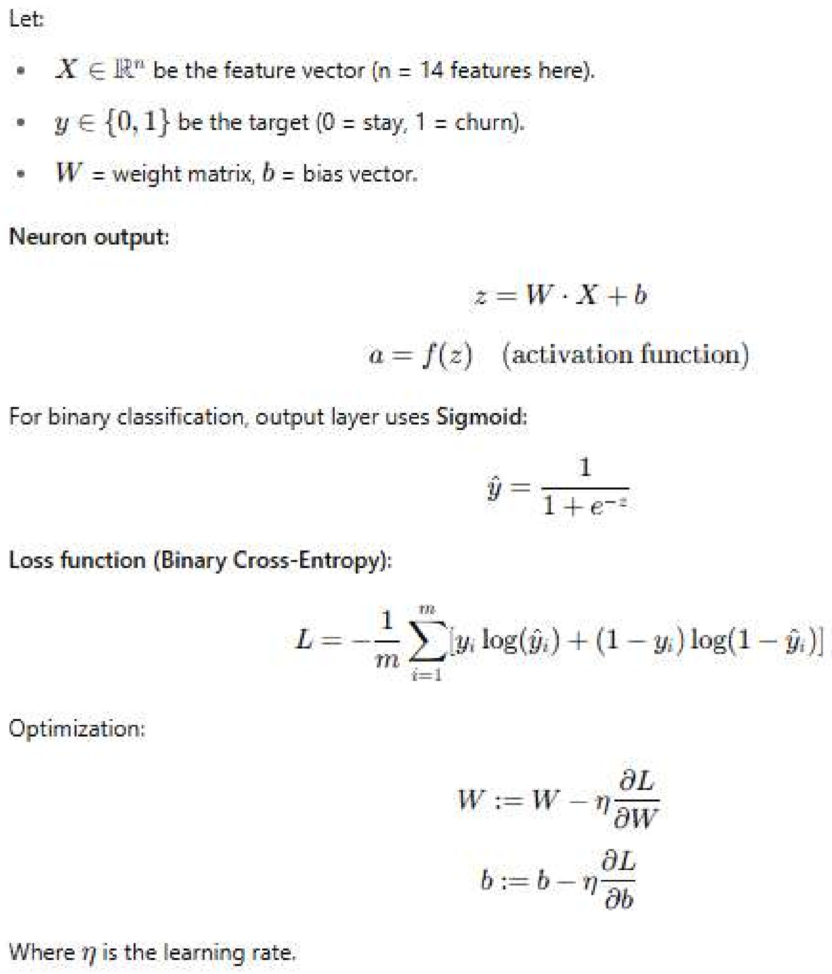
For binary classification (churn prediction):

	Predicted Stay (0)	**Predicted Churn (1)**
Actual Stay (0)	True Negative (TN)	False Positive (FP)
Actual Churn (1)	False Negative (FN)	True Positive (TP)

Interpretation:

- **True Positive (TP):** Customer churned and predicted churn.
- **True Negative (TN):** Customer stayed and predicted stay.
- **False Positive (FP):** Customer stayed but predicted churn (Type I error).
- **False Negative (FN):** Customer churned but predicted stay (Type II error).

Mathematical Model



Let:

- $X \in \mathbb{R}^n$ be the feature vector (n = 14 features here).
- $y \in \{0, 1\}$ be the target (0 = stay, 1 = churn).
- W = weight matrix, b = bias vector.

Neuron output:

$$z = W \cdot X + b$$

$$a = f(z) \quad \text{(activation function)}$$

For binary classification, output layer uses **Sigmoid:**

$$\hat{y} = \frac{1}{1 + e^{-z}}$$

Loss function (Binary Cross-Entropy):

$$L = -\frac{1}{m}\sum_{i=1}^{m}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

Optimization:

$$W := W - \eta\frac{\partial L}{\partial W}$$

$$b := b - \eta\frac{\partial L}{\partial b}$$

Where η is the learning rate.

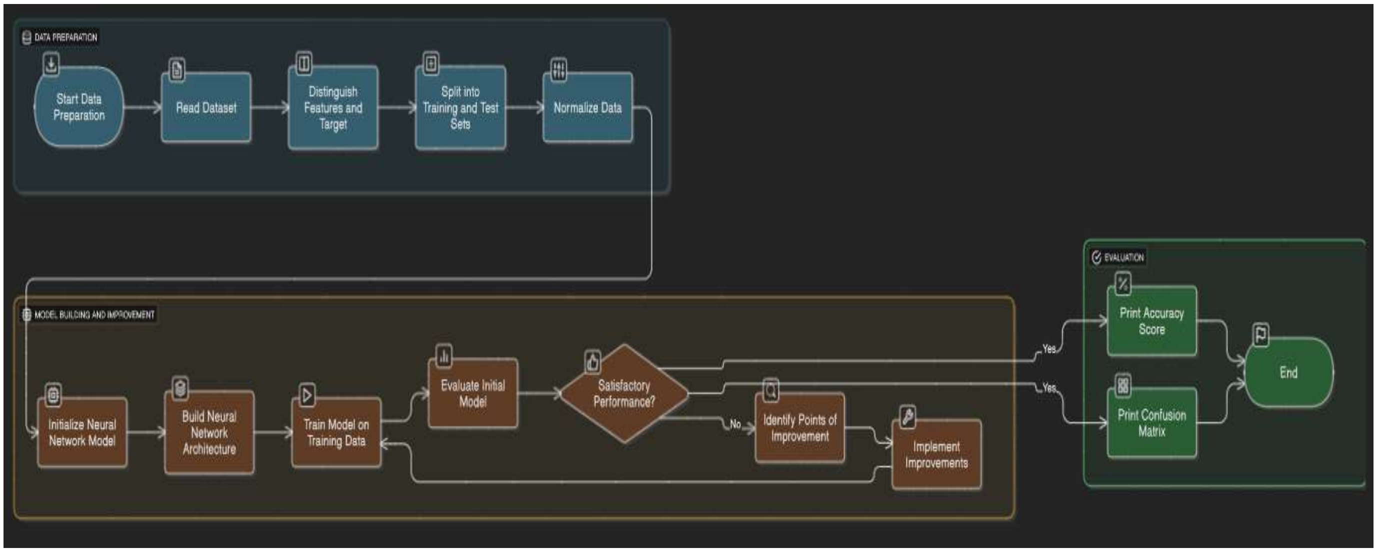
Model Evaluation Metrics:

1. **Accuracy**: {TP + TN}/{TP + TN + FP + FN}
2. **Confusion Matrix**: Shows TP, TN, FP, FN
3. **Precision**: {TP}/{TP + FP}
4. **Recall**: {TP}/{TP + FN}
5. **F1 Score**: Harmonic mean of precision and recall ((2 * P*R) / (P + R))

Algorithm

1. Load dataset using Pandas.
2. Identify features and target variable.
3. Encode categorical variables (Geography, Gender).
4. Split dataset into training (80%) and testing (20%).
5. Normalize numerical features using StandardScaler.
6. Build a Sequential ANN model:
    - Input layer
    - Hidden layers (Dense, ReLU activation)
    - Output layer (Dense, Sigmoid activation)
7. Compile model (loss = binary crossentropy, optimizer = Adam).
8. Train model on training set.
9. Predict on test set.
10. Evaluate accuracy and confusion matrix.

Flow Chart


DATA PREPARATION
Start Data Preparation
Read Dataset
Distinguish Features and Target
Split into Training and Test Sets
Normalize Data
MODEL BUILDING AND IMPROVEMENT
Initialize Neural Network Model
Build Neural Network Architecture
Train Model on Training Data
Evaluate Initial Model
Satisfactory Performance?
No
Identify Points of Improvement
Implement Improvements
Yes
Yes
EVALUATION
Print Accuracy Score
Print Confusion Matrix
End

Result:

(Insert Screenshots: Accuracy, Confusion Matrix, Accuracy/loss curve from Keras training)

Conclusion

The neural network-based model achieved a good accuracy **……… (Write Accuracy)** in predicting bank customer churn. Feature scaling, categorical encoding, and dropout regularization helped improve performance. This model can assist banks in identifying at-risk customers early.

Viva Questions

Question	**BT Level**
What is customer churn?	L1 (Remember)
Why is data normalization important in neural networks?	L2 (Understand)
Explain the working of a neural network with example.	L2 (Understand)
How is binary cross-entropy loss calculated?	L3 (Apply)
How does dropout prevent overfitting?	L4 (Analyze)
Suggest ways to improve model performance in this case.	L5 (Evaluate)
If accuracy is high but recall is low, what does it indicate?	L5 (Evaluate)
How would you modify the network for multi-class classification?	L6 (Create)

Rubrics for Evaluation (Example)

Criteria	**Best**	**Good**	**Average**	**Poor**
Correctness of Program / Demonstration of Program [4]	Perfect [4]	Can be better [3]	Satisfied [2]	Poor [0]
File Submission/Documentation [2]	Perfect [2]	Can be better [1.5]	Satisfied [1]	Poor [0]
Timely Submission [2]	On Time [2]	After 1 week [1.5]	After 1.5 week [1]	At the end of semester [0]
Viva [2]	Perfect [2]	Can be better [1.5]	Satisfied [1]	Poor [0]

Dr. Nikita Singhal

(Subject Incharge)

DR SR Dhore

(HoD)