Assignment No 3

Title:

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)

Theory:

Neural Networks

1. Artificial Neuron (Perceptron)

The basic building block of a neural network is an artificial neuron, also known as a perceptron. A perceptron takes a set of input values, applies weights to each input, and combines them to produce an output using an activation function. Mathematically, the output of a perceptron can be represented as:

scssCopy code

output = activation_function(sum(weight_i * input_i) + bias)

The activation function introduces non-linearity to the model and allows the neural network to learn complex patterns in the data.

2. Neural Network Architecture

A neural network consists of multiple layers of interconnected neurons. The three main types of layers in a neural network are:

- **Input Layer**: The first layer that receives the input data.

- **Hidden Layers**: Intermediate layers between the input and output layers. These layers are responsible for learning patterns and representations in the data.
- **Output Layer**: The final layer that produces the network's output, which is typically the prediction for a specific task.

The number of neurons in the input layer is determined by the number of features in the dataset, while the number of neurons in the output layer depends on the nature of the problem (binary classification in this case, so one neuron for churn prediction).

3. Forward Propagation

Forward propagation is the process of passing input data through the neural network to obtain predictions. During forward propagation, the input data travels through each layer, and the neurons in each layer perform computations and pass the information to the next layer until the output layer produces the final predictions.

4. Loss Function

The loss function (also called the cost function or objective function) measures how well the neural network's predictions match the actual target values. For binary classification tasks like customer churn prediction, the commonly used loss function is the binary cross-entropy loss. The goal of training the neural network is to minimize this loss function.

5. Backpropagation

Backpropagation is the core algorithm used to train neural networks. It involves computing the gradients of the loss function with respect to the model's weights and biases. These gradients are then used to update the weights and biases in the direction that minimizes the loss function. The process of iteratively updating the model's parameters using backpropagation is known as optimization.

6. Hyperparameters

Hyperparameters are configuration settings that determine the architecture and behavior of the neural network. Some important hyperparameters include the number of hidden layers, the number of neurons in each hidden layer, the learning rate (controls the step size during optimization), the activation function for each layer, and the number of training epochs (iterations).

Implementation Steps

Introduction

Customer churn refers to the phenomenon where customers discontinue their relationship with a business or service provider. In the banking sector, customer churn is a critical concern as retaining

existing customers is more cost-effective than acquiring new ones. In this lab, we will build a neural network-based classifier to predict whether a bank customer will leave or stay with the bank in the next 6 months.

Prerequisites

Before proceeding with the lab, ensure you have the following:

1. Python installed on your machine.
2. Necessary Python libraries: **pandas**, **numpy**, **scikit-learn**.
3. The "bank_customer_churn.csv" dataset downloaded from Kaggle: Bank Customer Churn Modeling Dataset.

Part 1: Data Preparation

1. Read the dataset: Load the "bank_customer_churn.csv" dataset into a Pandas DataFrame.
2. Explore the dataset: Check the data types, summary statistics, and any missing values in the dataset.
3. Distinguish the feature set and target set: Identify the input features (e.g., **CreditScore**, **Geography**, **Gender**, **Age**, **Tenure**, **Balance**, etc.) and the target variable (i.e., whether the customer churned in the next 6 months).
4. Split the dataset: Divide the dataset into training and test sets to train and evaluate the model.

Part 2: Data Preprocessing

1. Feature Scaling: Normalize the numerical features (e.g., **CreditScore**, **Age**, **Balance**) to bring them within a similar range. We will use Min-Max scaling to scale the features between 0 and 1.

Part 3: Building the Neural Network Model

1. Initialize the model: Create a sequential neural network model sklearn.neural_network
2. Add layers: Design the neural network architecture by adding input, hidden, and output layers. Experiment with different activation functions and the number of hidden layers to identify points of improvement.
3. Compile the model: Configure the loss function, optimizer, and evaluation metric for the model.

Part 4: Model Training and Evaluation

1. Train the model: Fit the neural network model to the training data and set the number of epochs and batch size.

2. Evaluate the model: Use the test set to evaluate the model's performance.
3. Identify points of improvement: Based on the evaluation results, make adjustments to the model architecture, learning rate, or other hyper parameters to improve model performance.
4. Re-train the model: Train the updated model using the adjusted hyper parameters.

Part 5: Model Evaluation Metrics

1. Accuracy Score: Calculate the accuracy score, which represents the percentage of correctly predicted customer churn instances over the total test set.
2. Confusion Matrix: Generate the confusion matrix to analyze the model's true positive, true negative, false positive, and false negative predictions.

Conclusion

In this lab, we explored the task of bank customer churn prediction using a neural network-based classifier. We prepared the data, built the neural network model, trained it on the training set, and evaluated its performance on the test set. We also calculated the accuracy score and generated the confusion matrix to assess the model's performance.