Multi-Layer perceptron Neutral Network For bank's marketing campaign

Punsisi Kosgahakumbura

February 2024

Table of Contents

01. A Brief Introduction to the Dataset

The dataset provides a comprehensive insight into a bank's marketing campaign, offering a detailed analysis of over 11,000 customers who were contacted regarding a term deposit offer. It encompasses 16 distinct features, ranging from demographic information such as age and occupation to campaign-specific details like contact type and prior interactions with the bank. The dataset culminates in a binary target variable indicating whether a customer ultimately subscribed to the term deposit or not.

By delving into this dataset, financial institutions gain invaluable insights into customer behavior and preferences, enabling them to tailor future marketing endeavors more effectively. Understanding the intricate interplay between customer demographics, campaign tactics, and subscription outcomes empowers banks to optimize their outreach strategies, ensuring they target the right audience with the most suitable approach. Through data-driven analysis of this rich tapestry of customer information, financial institutions can maximize the efficacy of their marketing campaigns, ultimately enhancing customer engagement and satisfaction while optimizing resource allocation.

02. Neural Network Architecture

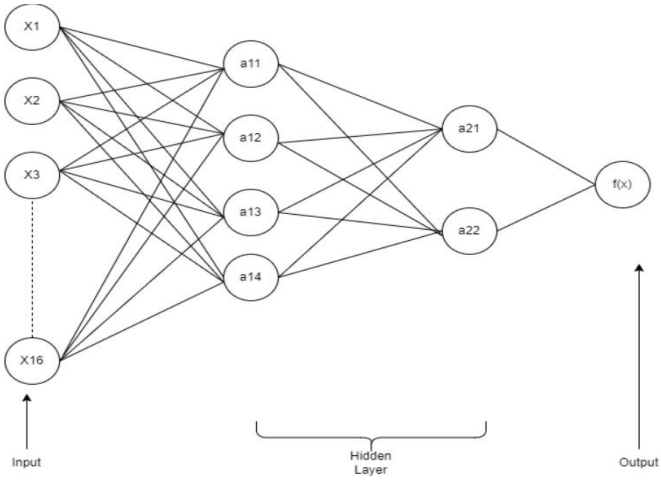

Figure 01

Given the dataset's dimensionality of 16 input features, the decision was made to correspondingly utilize 16 neurons in the input layer. Acknowledging the relatively manageable complexity of the dataset and its modest dimensionality, a choice was made to incorporate two hidden layers into the neural network architecture.

The selection of the number of neurons in the hidden layers is crucial for effective learning. To determine an appropriate number, a guideline was followed, which suggests that the number of hidden neurons should ideally fall between the size of the input layer and the output layer. Additionally, considering the square root of the product of the input layer nodes and output layer nodes, a preliminary estimation was made, resulting in four neurons allocated to the first hidden layer.

The subsequent layer's neuron count was deliberately reduced to promote progressively refined pattern and feature extraction, facilitating the identification of the target class. Hence, a decision was made to employ two neurons in the second hidden layer, aiming to further enhance the network's ability to discern intricate patterns and relationships within the data.

03. Used Parameters

3.1 Activation Function

The neural network architecture for this scenario comprises an input layer and two hidden layers, each utilizing the hyperbolic tangent (tanh) activation function. Tanh is chosen for its ability to squash input values to the range [-1, 1], aiding in data normalization and stability during training. For the output layer, the sigmoid activation function is employed due to its suitability for binary classification tasks, such as predicting whether a customer subscribes to a term deposit. This setup allows the network to effectively capture nonlinear relationships within the data and accurately predict customer responses to the term deposit offer.

3.2 Loss function

The output of this problem is whether a customer deposits or not, making it a binary classification problem. Therefore, Binary Cross Entropy is chosen as the loss function due to its compatibility with the sigmoid activation function commonly used in the output layer of neural networks for binary outcomes. The sigmoid function maps raw model outputs to probabilities between 0 and 1, aligning well with the nature of binary classification tasks. Binary Cross Entropy measures the discrepancy between predicted probabilities and true binary labels, making it particularly suitable for problems where the model's output is interpreted as the probability of belonging to the positive class.

3.3 Optimizer

Stochastic Gradient Descent (SGD) is employed as the optimizer for the neural network. SGD, a widely utilized optimization algorithm, iteratively adjusts the model parameters to minimize the Binary Cross Entropy loss function. This iterative process involves computing the gradient of the loss with respect to the model parameters using randomly selected mini-batches from the training data.

3.4 Learning Rate and Momentum

The momentum is set to 0.8, indicating that 80% of the past gradient's influence persists in the current weight update. A momentum of 0.8 strikes a balance between responsiveness to recent gradients and maintaining stability. Additionally, the learning rate (lr) used is 0.01, a moderate value controlling the step size in weight space during each iteration.

3.4 Epochs

For the initial run, 100 epochs were executed with a batch size of 64, resulting in a run summary accuracy of 0.84615 and a loss of 0.32359.

Subsequently, another trial involved 100 epochs with a reduced batch size of 32, producing a runtime accuracy of 0.80769 and a loss of 0.35455.

Further experimentation included 50 epochs with a batch size of 32, where the run summary displayed an accuracy of 0.80769 and a loss of 0.39228.

In the last approach, 50 epochs were run with a batch size of 64, achieving the highest accuracy at 0.88462 and a minimum loss of 0.27056.

04. The Result Graphs Generated by Weights & Biases

The results graphs of the finalized parameters using;

Epochs: 50 Learning rate: 0.01

Batch size: 64 Momentum: 0.8

Figure 02

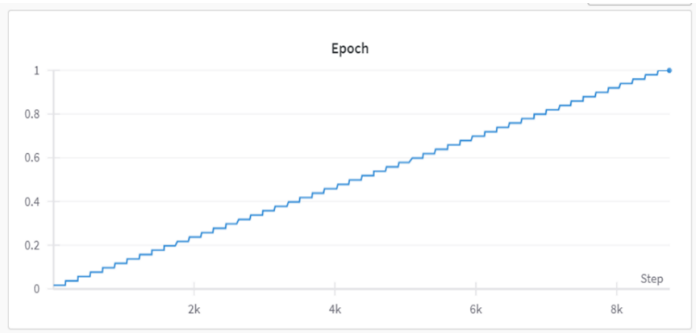

Figure 03

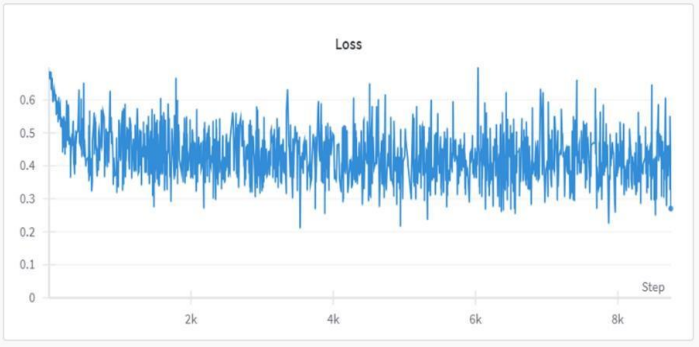

Figure 04

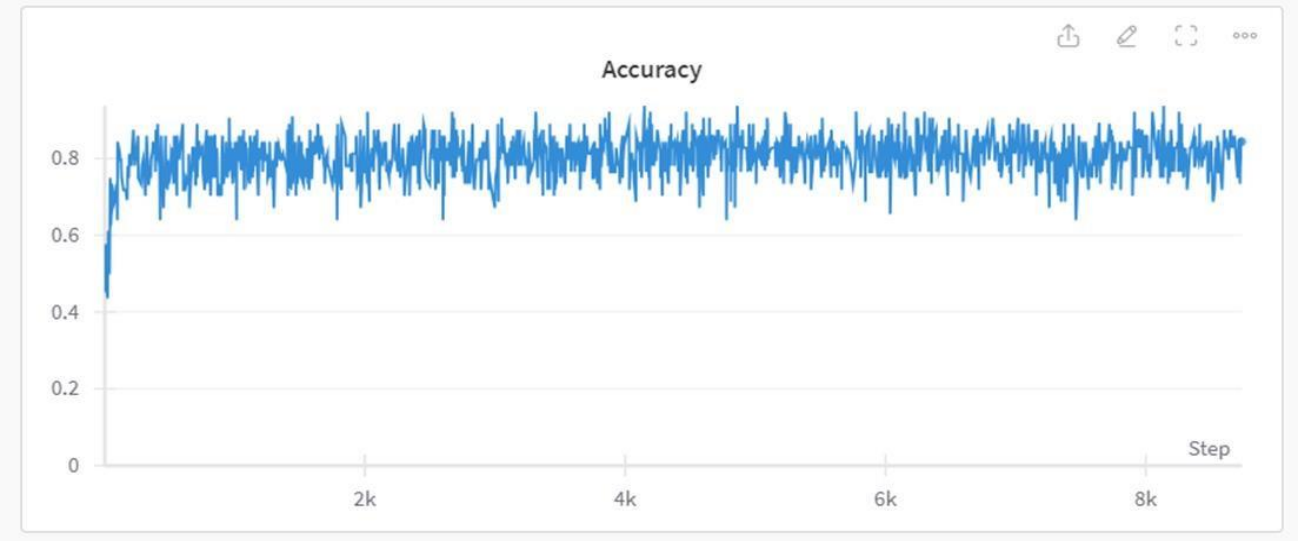

Figure 05

05. The Problem Faced During the Implementation

I. To address the challenge of handling categorical data in the dataset, two methods were explored: one-hot encoding and mapping dictionary method.

Initially, one-hot encoding was employed to convert categorical variables into numerical representations. However, this approach resulted in an expansion of the feature space, leading to an increase in the number of features to 51, despite the dataset containing only 16 original features. This expansion could potentially introduce issues related to dimensionality and computational complexity.

To mitigate this issue, an alternative approach was adopted using the mapping dictionary method. This method involved creating a separate dictionary for each categorical variable and mapping each category to a numerical value. By doing so, all categorical variables were effectively converted into numerical representations, aligning with the formulation of the neural network.

II. The absence of a distinct testing set

Typically, best practices dictate the division of a dataset into training and testing sets to ensure a robust evaluation of a model's performance. However in here, the entire dataset was used exclusively for training the multi-layer perceptron neural network for the convenience. Consequently, the absence of a distinct testing set has implications for the evaluation phase, as there is no separate dataset to assess the model's generalization to new, unseen data. This streamlined approach accelerate the training process but does limit the comprehensive evaluation of the model's effectiveness.