Name: Yash Gaikwad
Roll No: 381071
PRN: 22420141

Assignment 1

Problem Statement: Implementing Feedforward neural networks in Python using Keras and TensorFlow

Introduction:

Artificial Neural Networks (ANNs) are the backbone of modern deep learning applications, inspired by the functioning of the human brain. Among them, Feedforward Neural Networks (FNNs) are one of the simplest yet most powerful architectures. In this model, information moves in one direction—from the input layer through hidden layers to the output layer—without any cycles or loops.
With the advancement of frameworks like **TensorFlow** and its high-level API **Keras**, building and training neural networks has become simpler and more efficient. In this assignment, we implement a feedforward neural network using these tools to understand the concepts of deep learning practically.

Objective:

The main objectives of this assignment are:

- To understand the architecture of a feedforward neural network.
- To implement a feedforward neural network in Python using **Keras** with **TensorFlow** backend.
- To train the model on a dataset (e.g., MNIST or any chosen dataset) and evaluate its performance.
- To gain hands-on experience in defining layers, activation functions, compiling, training, and testing a model.

Theory:

1 Feedforward Neural Network

- A Feedforward Neural Network is an ANN where the connections between nodes do not form cycles.
- It consists of three main layers:
    1. **Input Layer** – Accepts input features from the dataset.
    2. **Hidden Layers** – Perform weighted computations and transformations using activation functions like ReLU, Sigmoid, or Tanh.
    3. **Output Layer** – Produces the final predictions using appropriate activation (e.g., softmax for classification).

2 Working Principle

1. Each input is multiplied by weights, and a bias term is added.
2. The result passes through an **activation function** to introduce non-linearity.
3. Forward propagation continues layer by layer until the output is produced.
4. The error (difference between predicted and actual output) is calculated using a **loss function**.
5. Using **backpropagation** and an **optimizer** (like SGD or Adam), weights are updated to minimize the error.

3 Keras and TensorFlow

- **TensorFlow**: An open-source library developed by Google for numerical computation and large-scale machine learning.
- **Keras**: A high-level API running on top of TensorFlow that simplifies building and training deep learning models with an easy-to-use interface.

4 Example Implementation Steps

1. Import necessary libraries (tensorflow, keras).
2. Load and preprocess the dataset.
3. Define a Sequential feedforward model.
4. Add input, hidden, and output layers with suitable activation functions.
5. Compile the model with a loss function and optimizer.
6. Train the model on training data.
7. Evaluate performance using test data.

Conclusion:

Through this assignment, we successfully implemented a Feedforward Neural Network using Keras and TensorFlow. The exercise helped in understanding the key concepts of forward propagation, activation functions, and backpropagation. By training and testing the network, we observed how deep learning models can learn from data and generalize to unseen inputs. This foundational knowledge of FNNs serves as the building block for exploring more advanced architectures like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs).