**Assignment 1 – Feedforward Neural Network for Wine Dataset Classification**

Name : Shubham Gautam Dahane

Roll No: 382015

PRN: 22310312

**Problem Statement**

Implement a feedforward neural network in Python using Keras and TensorFlow to classify wine samples into three categories based on their chemical features.

**Objectives**

* To understand feedforward neural networks (FNNs) and their architecture.
* To preprocess the Wine dataset using scaling and one-hot encoding.
* To build, compile, and train a neural network with Keras and TensorFlow.
* To evaluate performance using test accuracy and confusion matrix.
* To visualize model training with accuracy and loss plots.

**Requirements**

* **Operating System:** Windows/Linux/MacOS
* **Python Version:** 3.x
* **Tools:** Jupyter Notebook / Anaconda / Google Colab
* **Libraries Used:**
  + TensorFlow, Keras
  + NumPy
  + Matplotlib
  + Scikit-Learn

**Theory**

A **feedforward neural network (FNN)** is a supervised learning algorithm where information flows in one direction—from input to output—without loops or cycles.

* **Input Layer:** Accepts the input data (13 wine features).
* **Hidden Layers:** Perform transformations using weights, biases, and activation functions (ReLU).
* **Output Layer:** Uses Softmax activation for multiclass classification (3 classes of wine).
* **Backpropagation:** Adjusts weights by propagating error backward to optimize performance.



**Methodology**

1. **Data Acquisition**
   * The **Wine dataset** from Scikit-Learn is used, containing 178 samples with 13 features and 3 classes.
2. **Data Preparation**
   * Features are scaled using StandardScaler for faster convergence.
   * Target labels are converted into one-hot encoded format.
   * Dataset split into **80% training** and **20% testing**.
3. **Model Architecture**
   * **Input Layer:** 13 neurons (for 13 features).
   * **Hidden Layer 1:** 32 neurons, ReLU activation.
   * **Hidden Layer 2:** 16 neurons, ReLU activation.
   * **Output Layer:** 3 neurons, Softmax activation (for 3 classes).
4. **Model Compilation**
   * Optimizer: Adam
   * Loss Function: Categorical Crossentropy
   * Metric: Accuracy
5. **Model Training**
   * Trained for 50 epochs, batch size = 8.
   * Validation data used to monitor performance.
6. **Model Evaluation**
   * Accuracy on test set measured.
   * Sample prediction demonstrated.
7. **Performance Analysis**
   * Accuracy and loss plotted across epochs.
   * Confusion matrix generated to analyze classification errors.

**Graphs and Visualizations**

1. **Accuracy vs. Epochs**
   * Training and validation accuracy plotted to observe learning progress.

A graph of a graph

AI-generated content may be incorrect.

1. **Loss vs. Epochs**
   * Training and validation loss curves plotted to check convergence and possible overfitting.

A graph of a graph with blue and orange lines

AI-generated content may be incorrect.

1. **Confusion Matrix**
   * Visualizes correct and incorrect classifications across the 3 wine classes.
   * Helps identify which classes are more difficult for the model to classify.

A diagram of a confusion matrix

AI-generated content may be incorrect.

**Advantages**

* Captures **non-linear relationships** in data.
* High classification accuracy on structured datasets.
* Provides insights using confusion matrix and performance plots.

**Limitations**

* Sensitive to **hyperparameter tuning** (epochs, layers, learning rate).
* May overfit on small datasets.
* Computationally expensive with larger and deeper networks.

**Applications**

* **Food Quality Control** – wine, beverages, or food classification.
* **Healthcare** – disease detection/classification.
* **Finance** – fraud detection and risk analysis.
* **Agriculture** – crop or soil classification.

**Working / Algorithm**

**Step 1:** Import required libraries.  
**Step 2:** Load Wine dataset using Scikit-Learn.  
**Step 3:** Preprocess features (scaling, one-hot encoding).  
**Step 4:** Split dataset into training and testing sets.  
**Step 5:** Build Sequential model with Dense layers.  
**Step 6:** Compile model with optimizer, loss, and metrics.  
**Step 7:** Train model for 50 epochs with validation.  
**Step 8:** Evaluate model on test set and print accuracy.  
**Step 9:** Make predictions on sample data.  
**Step 10:** Plot accuracy/loss graphs and confusion matrix.

**Conclusion**

The feedforward neural network was successfully implemented to classify wines into three categories. The model achieved strong accuracy on the test dataset and provided meaningful insights through accuracy/loss graphs and confusion matrix visualization.

This demonstrates the effectiveness of FNNs for multiclass classification tasks when proper preprocessing and training strategies are applied.