# Assignment 1 – Implementing Feedforward neural networks in Python using Keras and TensorFlow

Name : Aayush Bokde

Roll No: 382016

PRN: 22310316

## 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

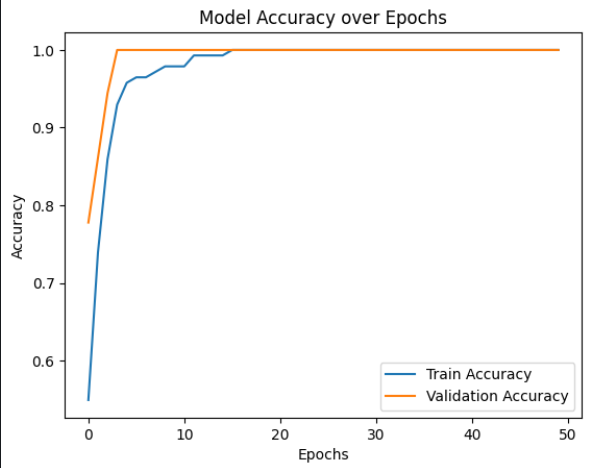
## Theory

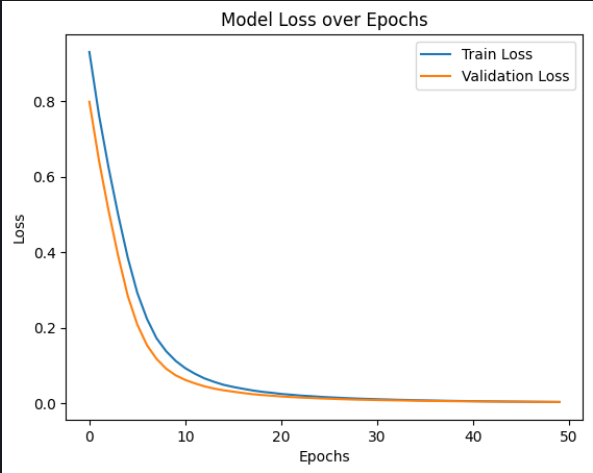
A feedforward neural network (FNN) is a supervised learning algorithm where information flows in one direction—from input to output—without loops or cycles.  
  
Input Layer: Accepts the input data (13 wine features).  
Hidden Layers: Perform transformations using weights, biases, and activation functions (ReLU).  
Output Layer: Uses Softmax activation for multiclass classification (3 classes of wine).  
Backpropagation: Adjusts weights by propagating error backward to optimize performance.

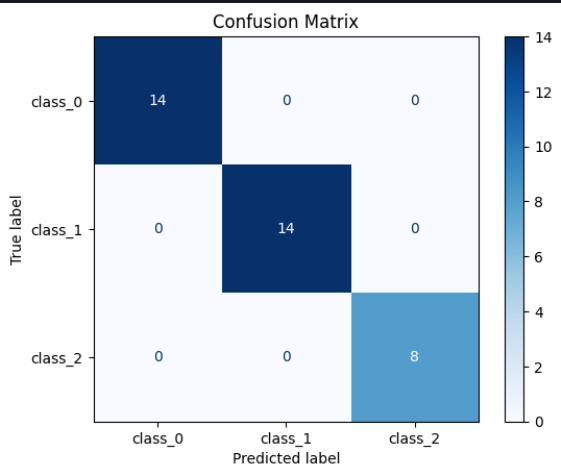
## Methodology

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 and labels are converted into one-hot encoded format. The dataset is split into 80% training and 20% testing.
3. Model Architecture: Input layer with 13 neurons, hidden layers with 32 and 16 neurons (ReLU activation), and output layer with 3 neurons (Softmax activation).
4. Model Compilation: Optimizer = Adam, Loss Function = Categorical Crossentropy, Metric = Accuracy.
5. Model Training: Trained for 50 epochs with batch size of 8 and validation data.
6. Model Evaluation: Test accuracy measured, sample predictions demonstrated.
7. Performance Analysis: Accuracy/Loss curves plotted, confusion matrix generated.

## Graphs and Visualizations

* Accuracy vs. Epochs: Training and validation accuracy plotted to observe learning progress.  
  
* Loss vs. Epochs: Training and validation loss curves plotted to check convergence and overfitting.



* Confusion Matrix: Shows classification results for the 3 wine classes.

## Advantages

* Captures non-linear relationships in data.
* Achieves high accuracy on structured datasets.
* Provides interpretability through confusion matrix and plots.

## Limitations

* Sensitive to hyperparameter tuning (epochs, layers, learning rate).
* Risk of overfitting on small datasets.
* Training deeper networks may increase computation time.

## Applications

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

## Working / Algorithm

1. Import required libraries.
2. Load Wine dataset using Scikit-Learn.
3. Preprocess features (scaling, one-hot encoding).
4. Split dataset into training and testing sets.
5. Build Sequential model with Dense layers.
6. Compile model with optimizer, loss, and metrics.
7. Train model for 50 epochs with validation.
8. Evaluate model on test set and print accuracy.
9. Make predictions on sample data.
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 insights through accuracy/loss plots and confusion matrix visualization. This demonstrates the effectiveness of FNNs for multiclass classification tasks.