



**Disclaimer: The content is curated from online/offline resources and used for educational purpose only**

**LAB MANUAL**

**Unit V – Deep Learning**

**Unit V – Deep Learning**

**Lab 2. To create the Feedforward Neural Network (FNN) using TensorFlow and Keras to predict climate data**

**Objective**

* Build a Feedforward Neural Network (FNN) capable of predicting climate metrics such as temperature, humidity, or precipitation using historical climate data.
* Design an FNN architecture using TensorFlow and Keras with appropriate layers, neurons, and activation functions to capture the underlying relationships in the data.
* Evaluate the model using metrics such as Mean Squared Error (MSE) or Mean Absolute Error (MAE) and optimize it for better performance through hyperparameter tuning.

**Problem**

Design and implement a Feedforward Neural Network (FNN) using TensorFlow and Keras to predict climate data, leveraging historical climate metrics such as temperature, humidity, and precipitation to generate future predictions. The model should aim to identify patterns and provide accurate predictions to support environmental monitoring and climate change research.

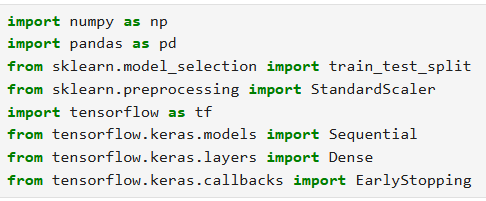
**Solution**

we'll go through the following steps:

1. Import required libraries
2. Load Dataset
3. Select features (X) and target variable (y)
4. Split the data into training and testing sets
5. Scale the data for better performance
6. Build the FNN Model
   1. Initialize the model
   2. Input layer
   3. Hidden layers
   4. Output layer (for regression tasks)
   5. Compile the model
7. Train the Model
8. Evaluate the Model
9. Make Predictions
10. Display the first few predictions
11. Visualize Training History

**Procedures**

1. Import required libraries



1. Load Dataset



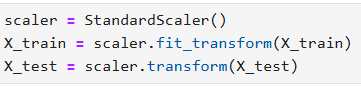
1. Select features (X) and target variable (y)



1. Split the data into training and testing sets



1. Scale the data for better performance



1. Build the FNN Model
   1. Initialize the model



* 1. Input layer



* 1. Hidden layers



* 1. Output layer (for regression tasks)

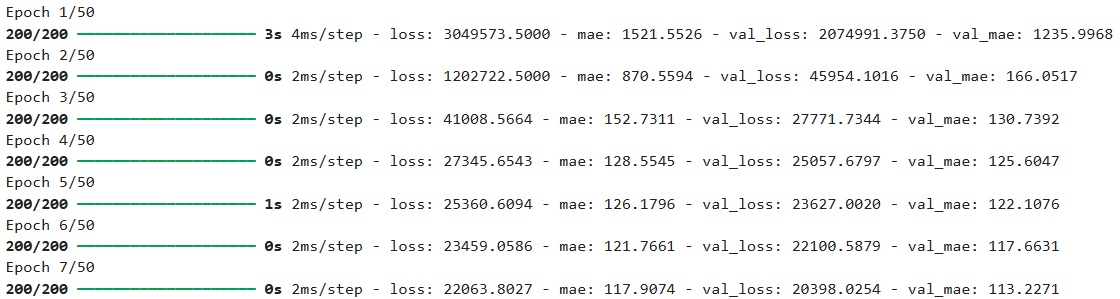


* 1. Compile the model



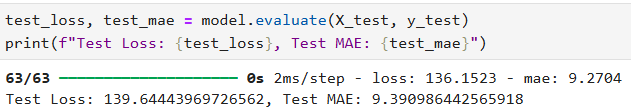
1. Train the Model





Running until 50th epoch.

1. Evaluate the Model

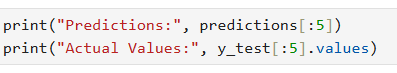


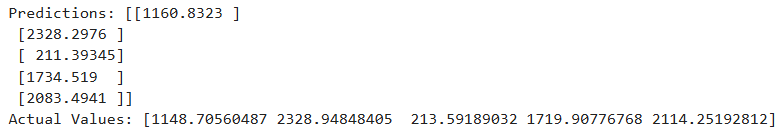
1. Make Predictions





1. Display the first few predictions





1. Visualize Training History

