TASK SUMMARY

APPLYING RNN TO JENA CLIMATE DATASET

Introduction:

In this project, we used Machine learning technique called Recurrent Neural Network (RNN) to make predictions on dataset "Jena Climate Data." This dataset consists of data of time period running from 2009 to 2016 and has important information about the weather patterns and circumstances. The Jena Climate Data has insights of climatic variations and weather conditions, which helps us in understanding how the temperature, humidity, pressure, and other factors effects with change of time. For handling time series data, RNN Neural Network is utilized to understand patterns and forecast weather conditions.

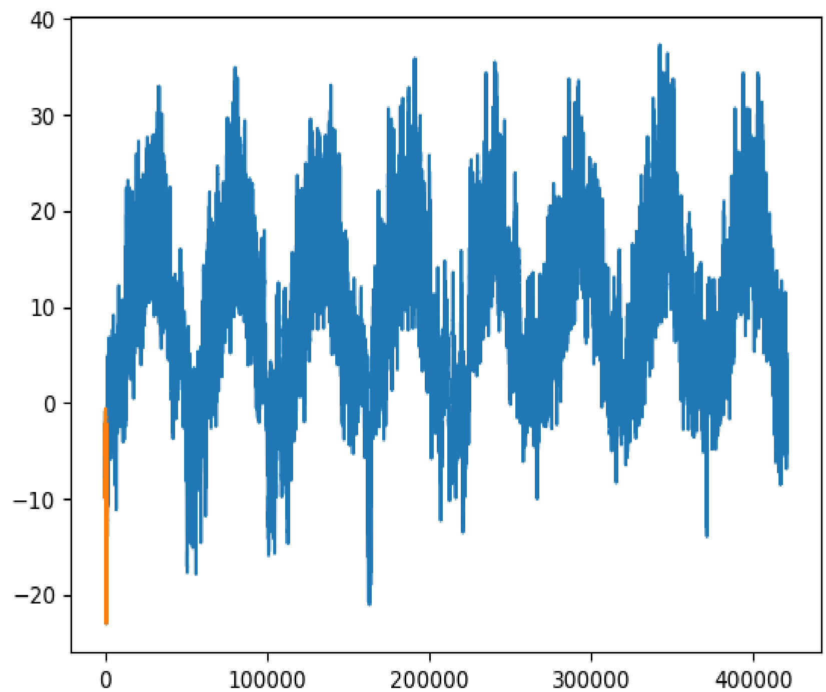
Model Training and Evaluation:

An RNN model was built, having an LSTM (Long Short-Term Memory) layer, which is well suited for time series forecasts. The model was trained for 10 epochs. Throughout training, the loss, measured as mean squared error, showed a falling trend on both training and validation sets.

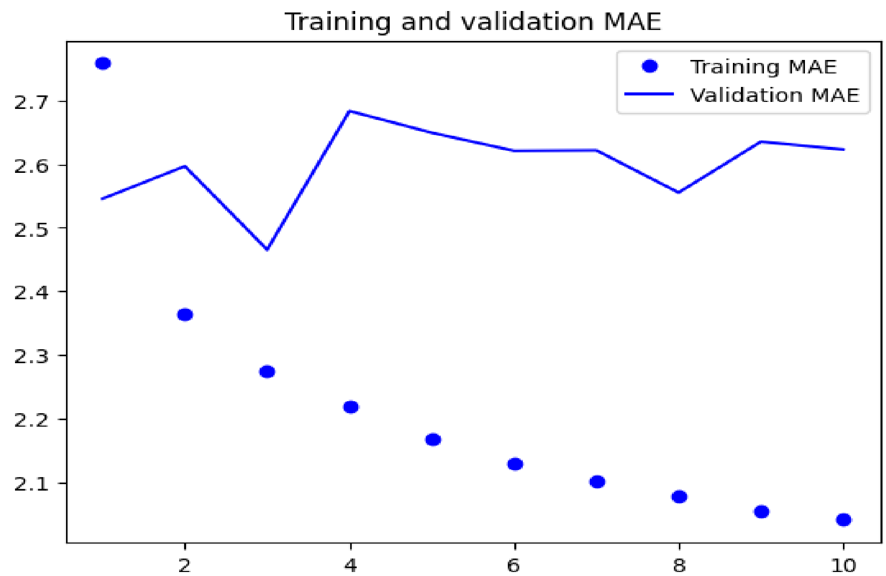
num_train_samples: 210225

num_val_samples: 105112

num_test_samples: 105114



Results and Data Visualization:



1. Adjusting the number of units in each recurrent layer in the stacked setup.

Time	value	Epoch	Loss	MAE
31 Secs	16	10	10.44	2.55
29 secs	32	10	11.13	2.65
47 secs	128	10	10.60	2.58

By considering the loss and MAE, the model with 16 units per recurrent layer is the best model by taking more time to train compared to the second model. In terms of loss and MAE, the third model which used 128 units per layer slightly reduced in performance. Identifying the most suitable setup for your application requires careful consideration of the trade-off between model complexity and performance, as well as the amount of time needed for training.

2. Using layer_lstm() instead of layer_gru().

For time series prediction, I used LSTM layers instead of GRU layers. With an improved Mean Absolute Error (MAE) of 2.54 and a loss of 10.44, the LSTM-based model exceeded the GRU-based model, which had an MAE of 2.65. This proves that long-term dependencies in the dataset are captured best by LSTM layers, focusing on the importance of selecting the right recurrent layer for increased predictive accuracy.

3. Using a combination of 1d_convnets and RNN.

For time series prediction, I employed a model that combines 1D Convolutional Neural Networks (ConvNets) and Recurrent Neural Networks (RNN). It took 46 seconds for the model to train, and its Mean Absolute Error (MAE) of 2.58 indicates that it is reasonably accurate. By combining the advantages of RNN for temporal pattern recognition and ConvNets for feature extraction, this hybrid approach shows promise for enhancing time series predictions. Performance may be optimized for tasks and datasets with additional refinements.

Summary:

The keyway to achieve optimal performance in time series prediction is to find the correct balance between predictive accuracy and model complexity. While overly simple models run the risk of missing patterns, overly complex models run the risk of overfitting. Training time is another factor to consider because complicated models might not be appropriate for real-time applications. Predictive accuracy is affected by the choice between GRU and LSTM layers. Using hybrid models can be useful, but careful setup is required to achieve their possibities.