Temperature Series Forecasting with GRU

Introduction

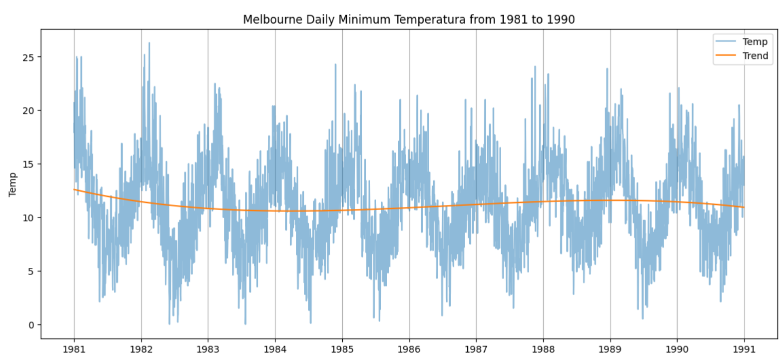
The project focuses on forecasting daily minimum temperatures in Melbourne, Australia, from 1981 to 1990, using deep learning techniques. This time series dataset provides a rich source of information for understanding temperature trends and patterns over a decade.

Dataset Overview

The dataset, sourced from Kaggle, comprises daily minimum temperature readings in Melbourne. It spans ten years, providing a comprehensive view of temperature fluctuations over time.

Data Preparation and Cleaning:

- Dataset Downloading: The dataset is downloaded using the Kaggle API.
- Data Loading: Loaded into a Pandas DataFrame, with special attention to parsing dates correctly.
- Data Cleaning: Erroneous data points (marked with '?') are identified and corrected. The temperature column is converted to a numeric type for analysis and modeling.
- Data Exploration: Basic statistical methods are employed to understand the dataset's characteristics, such as range, mean, and standard deviation.



Model Architecture

A neural network model specifically designed for time series forecasting is used. This model architecture combines convolutional and recurrent layers to capture temporal dependencies and patterns in the data.

Components of the Model:

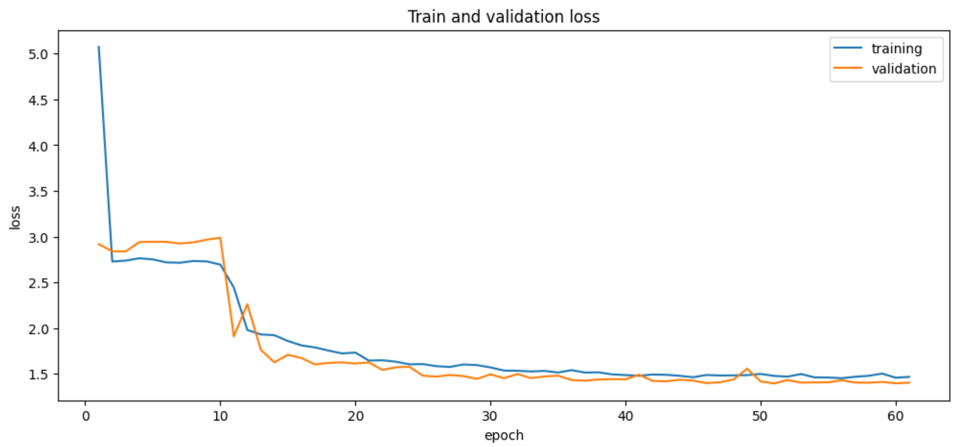
- Causal Convolutional Layer: Ensures that the prediction for a time step does not depend on future steps, maintaining the temporal order.
- GRU Layer: Replaces the traditional LSTM layer for capturing long-term dependencies, offering computational efficiency.
- Linear and Activation Layers: These are used for final prediction output, with ReLU activation functions to introduce non-linearity.

Choice of Hyperparameters

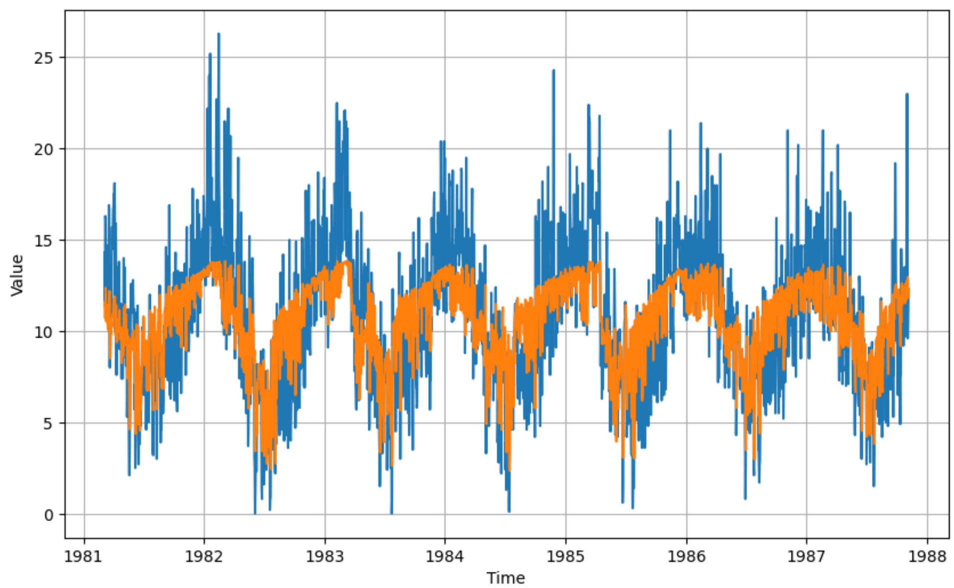
- Window Size: Set to 64, determining how many past days' data is used to predict the next day's temperature.
- Batch Size: 32, striking a balance between computational efficiency and gradient update frequency.
- Learning Rate and Optimizer: AdamW optimizer with an initial learning rate of 0.003, coupled with a learning rate scheduler.
- Loss Function: Huber loss is used, providing a robust metric that is less sensitive to outliers.

Training Process and Learning Curves

- Learning Rate Adjustment: A preliminary step to find an optimal learning rate, using a range test.
- Training and Validation Loss: Monitored to understand the model's learning progression and convergence.
- Early Stopping and Tolerance: Implemented to prevent overfitting and ensure the model does not train beyond necessary improvements.



mse: 8.66, mae: 2.33 for forecast



Observations from Learning Curves

- Forecasting Accuracy: The model's performance is evaluated on the validation set, focusing on metrics like Mean Absolute Error (MAE).
- Learning Curve Analysis: The learning curves (loss vs. epoch) for training and validation sets are analyzed to detect overfitting or underfitting.
- Model Predictions Visualization: The model's forecasts are plotted against the actual temperature readings, providing a visual representation of its accuracy and ability to capture trends.

Conclusion

The neural network model shows promise in forecasting daily minimum temperatures, demonstrating the capability of GRUs and convolutional layers in capturing time series patterns. The hyperparameter choices, particularly the window size and learning rate, play a crucial role in the model's performance. The project highlights the importance of careful preprocessing and data cleaning in time series forecasting, as well as the need for continuous evaluation and adjustment of model parameters.