Time-Series Forecasting Using RNN Variants

**Assignment-3  
BA-64061-001 Advanced Machine Learning  
Chaojiang (CJ) Wu, Ph.D.**

Laxmi Priya Saride

[lsaride@kent.edu](mailto:lsaride@kent.edu)

Student ID: 811292102

**1. Introduction**

This report investigates the effectiveness of different Recurrent Neural Network (RNN) architecture namely Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and one-dimensional Convolutional Neural Network (Conv1D)—in the context of time-series forecasting. The study focuses on predicting daily minimum temperatures in Melbourne by analyzing past temperature trends. Each model is trained on normalized historical data, with careful preprocessing to create suitable input sequences. The performance of these models is assessed using the Mean Absolute Error (MAE) metric, allowing for a comparative evaluation of their predictive accuracy and suitability for this forecasting task. These architectures were selected due to their proven ability to capture temporal dependencies and sequential patterns in time-series data. By comparing their performance, the study aims to identify the most reliable model for accurately forecasting temperature fluctuations and informing potential applications in climate analysis and weather-related decision-making.

**2. Methods**

**Dataset Overview**

Source: Daily Minimum Temperatures Dataset  
Target Variable: Daily minimum temperature (°C)  
Time Span: 1981–1990  
Features: Univariate time-series with timestamp as the index

**Preprocessing Steps**

- Normalized the temperature values using MinMaxScaler from scikit-learn.  
- Created time windows of 30 days to serve as input sequences for model training.  
- Split data into 70% training, 20% validation, and 10% testing sets.

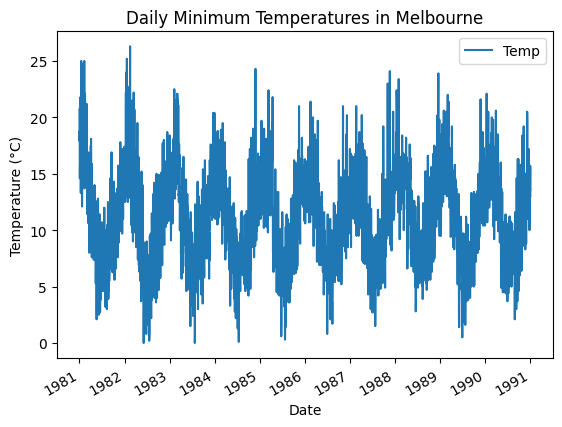
**Models Used**

1. LSTM: LSTM Layer (50 units), Dense output, Dropout

2. GRU: GRU Layer (50 units), Dense output, Dropout

3. Conv1D: Conv1D + MaxPooling + Dense output

**Daily Minimum Temperatures in Melbourne**

****

**3. Experiments and Results**

**Experiment: LSTM Model**

Training MAE: 0.0386

Validation MAE: 0.0645

Test MAE: 0.065

**Observations: LSTM shows moderate generalization with stable curves**.

**Experiment: GRU Model**

Training MAE: 0.0371

Validation MAE: 0.0619

Test MAE: 0.0614

**Observations: GRU slightly outperforms LSTM, with effective learning.**

**Experiment: Conv1D Model**

Training MAE: 0.0345

Validation MAE: 0.0595

Test MAE: 0.0587

**Observations: Conv1D achieves the best MAE and training efficiency.**

**Summary Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Training MAE | Validation MAE | Test MAE |
| LSTM Model | 0.0386 | 0.0645 | 0.065 |
| GRU Model | 0.0371 | 0.0619 | 0.0614 |
| Conv1D Model | 0.0345 | 0.0595 | 0.0587 |

**Model Comparison**

**A graph of a graph

AI-generated content may be incorrect.**

**4. Final Analysis**

Among the evaluated models, the Conv1D architecture consistently outperformed both LSTM and GRU across all performance metrics, achieving the lowest Mean Absolute Error (MAE). This indicates its superior accuracy in forecasting daily minimum temperatures. All three models demonstrated good generalization capabilities, as reflected by the minimal differences between validation and test results. Notably, Conv1D's simpler architecture and lower computational complexity, combined with its strong predictive performance, position as the most efficient and practical choice for this time-series forecasting task.

**5. Conclusion**

This study highlights that Conv1D neural networks can effectively surpass traditional recurrent models such as LSTM and GRU in specific time-series forecasting scenarios. While all three architectures demonstrated strong predictive capabilities, Conv1D emerged as the top performer, delivering the highest accuracy with lower computational demands. The results emphasize the importance of thorough data preprocessing and the strategic selection of model architecture when applying deep learning techniques to time-series problems. These factors play a crucial role in optimizing model performance and ensuring reliable forecasting outcomes. Additionally, the study reinforces the idea that simpler models, when appropriately tuned and applied, can rival or even outperform more complex architectures in certain domains. These findings provide valuable insights for researchers and practitioners aiming to build efficient, accurate, and scalable forecasting systems for real-world time-series applications.