Assignment 3: Time-Series Data

Exploring Time-Series Forecasting: A Journey Through 14 Models

Introduction

In the field of time-series forecasting, 14 models were developed and assessed in order to estimate if a particular stock would gain or decline over the course of the following week. A baseline model that did not use machine learning was the starting point of this trip. Basic machine learning and convolutional models were then developed. The time component of the data was difficult to sustain in these early attempts, though.

The Quest for Temporal Understanding

Due to the shortcomings of the first models, Recurrent Neural Networks (RNNs), which are well known for processing sequential input, were adopted. RNNs are excellent at extracting intricate patterns and dependencies from time-series data, allowing decision-makers to use historical data to guide their current decisions.

Navigating RNN Complexity

The vanishing gradient problem and other practical issues limited the usefulness of the basic RNN model in deep networks, despite its theoretical ability to preserve knowledge from all prior time periods. Graphical analysis supported this result by showing that the simple RNN was the least effective model out of all of them.

The Emergence of GRU

In response to the limitations of basic RNNs, Gated Recurrent Unit (GRU) models were developed. With an MAE of 2.47, the best-performing model was found to be an RNN version called GRUs. Compared to Long Short-Term Memory (LSTM) models, GRUs require less computing power to effectively capture long-range dependencies in sequential data.

Optimizing LSTM

Simultaneously, models of Long Short-Term Memory (LSTM), well known for their effectiveness in managing time-series data, were investigated. Six distinct LSTM models were created, each having a different number of units in stacked recurrent layers. The arrangement with eight units performed the best, resulting in an MAE of 2.55.

Enhancements for Improvement

Methods like bidirectional data consumption and recurrent dropout were used to improve the model's performance even more. The commonsense baseline model's MAE values were consistently higher after these improvements.

Challenges of Combining Models

The poor performance (MAE of 3.81) of the attempts to integrate RNN and 1D convolution models was ascribed to convolutional difficulties in maintaining information order.

Conclusion

To sum up, the investigation into LSTM and GRU architectures was successful, with GRU turning out to be the most efficient option. To maximize GRU performance, important hyperparameters such as the number of units in stacked recurrent layers, recurrent dropout rate, and bidirectional data use should be properly adjusted.