TIME SERIES FORCASTING

REPORT

Priyanka Veepuri

In my analysis of time series data, I developed a total of 11 models, each aimed at understanding and predicting patterns within the data. I began with a baseline model, employing straightforward methods based on common sense. This initial model provided a reference point with a Mean Absolute Error (MAE) of 2.62. Moving forward, I developed a basic machine learning model featuring a dense layer. However, this approach resulted in a slightly higher MAE of 2.70, as it struggled to capture the temporal context of the data due to flattening. Subsequently, I experimented with a convolutional model, hoping to leverage its ability to extract features from sequential data. Unfortunately, this approach proved ineffective, as it treated all data segments uniformly, disrupting the sequential order and yielding poor results.

Realizing the suitability of Recurrent Neural Networks (RNNs) for time series data, I explored their capabilities. While the basic Simple RNN theoretically should have been effective in retaining information from previous time steps, it often struggled practically due to the "vanishing gradient problem." This led me to investigate more advanced RNN variants such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). My experimentation revealed that the simple GRU model performed the best among all models, as it efficiently captured long-range dependencies while remaining computationally efficient.

Additionally, I tested six different LSTM models with varying units in stacked recurrent layers, and the model with 8 units demonstrated the best performance. Employing recurrent dropout and experimenting with bidirectional data presentation further enhanced the accuracy of these LSTM models.

Despite these successes, my attempts to combine a 1D convolution model with an RNN yielded suboptimal results, likely due to the convolution's limitations in maintaining the order of information.

In conclusion, I recommend avoiding simple RNNs for time series analysis due to their challenges with the vanishing gradient problem and inability to effectively capture long-term dependencies. Instead, I suggest utilizing more advanced RNN architectures such as LSTM and GRU, which are better equipped to overcome these challenges. Fine-tuning hyperparameters and focusing on RNN architectures tailored for sequential data are crucial for optimizing model performance.