**ADVANCED MACHINE LEARNING**

**ASSIGNMENT 3: TIME SERIES DATA**

**Rishitha Rapuri**

**811283595**

**SUMMARY:**

This study investigates the efficacy of neural network architectures for temperature prediction using the Jena Climate dataset as a case study. The primary objective is to develop and evaluate various neural network designs for efficient forecasting of future temperature values based on historical climate data. In our time series data analysis, a total of 14 models were developed. The initial model served as a baseline, employing conventional methods and yielding a Mean Absolute Error (MAE) of 2.62. Following this, we implemented a basic machine learning model incorporating a dense layer, resulting in a slightly higher MAE of 2.67. However, the performance of the dense layer model was deemed suboptimal due to the loss of temporal context resulting from the flattening of the time series data. Additionally, an attempt was made with a convolutional model, but it produced unsatisfactory results as it treated all data segments uniformly, even post-pooling, thereby disrupting the sequential order of the data.

Following our analysis, it became evident that Recurrent Neural Networks (RNNs) are better suited for time series data. A key advantage of RNNs is their ability to incorporate past information into current decision-making processes, facilitating the discovery of sequential data dependencies and patterns. The internal state of an RNN acts as a memory, enabling it to retain information from previous inputs and model sequences of varying lengths. However, the basic Simple RNN often falls short in practical applications. Notably, Simple RNN consistently performs poorly compared to other models, primarily due to the "vanishing gradient problem," making it challenging to train effectively. To overcome this limitation, more advanced RNN variants like Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) have been developed and integrated into frameworks like Keras. Our experimentation highlighted that the simple GRU model outperformed other models, thanks to its ability to capture long-range dependencies in sequential data while maintaining computational efficiency compared to LSTMs.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Dense units | Dropout | Loss | Test MAE |
| Basic Machine Learning Model | 16 | No | 11.4907 | 2.67 |
| 1D convolutional model | 16 | No | 15.1228 | 3.08 |

**RNN Models**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LSTM Models | 16 | No | 10.4045 | 2.52 |
| LSTM Models | 16 | Yes | 10.7263 | 2.58 |
| GRU (later replaced with LSTM)- not needed but did for comparison | 16 | Yes | 9.8000 | 2.49 |
| Bidirectional LSTM Model | 16 | No | 10.9553 | 2.61 |

The renowned architecture for proficiently handling time series data is LSTM, and we conducted experiments with six distinct LSTM models featuring varied units in stacked recurrent layers (8, 16, and 32). Surprisingly, the model with 8 units exhibited the most optimal performance. Furthermore, we utilized recurrent dropout to mitigate overfitting and explored bidirectional data presentation to improve accuracy and mitigate the forgetting issue. Remarkably, all these LSTM models showcased comparable MAE values, consistently outperforming the common-sense model.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Dense Units | Dropout | Loss | Test MAE |
| LSTM | 8 | No | 10.2210 | 2.49 |
| LSTM | 16 | No | 10.6859 | 2.56 |
| LSTM | 32 | No | 11.1510 | 2.63 |

LSTM with 8 units:

A graph of training and validation

Description automatically generated

LSTM with 16 units:

A graph of training and validation

Description automatically generated

LSTM with 32 units:

A graph of training and validation

Description automatically generated

Out of all the various combinations attempted, except for GRU, the LSTM model with a dropout rate of 0.5 yields the best MAE of 2.58 and a loss function of 10.7263.

A graph of training and validation

Description automatically generated

In our final analysis, we attempted to merge a 1D convolution model with an RNN. However, this hybrid approach resulted in a higher MAE of 3.92, likely due to the convolution's inability to preserve the sequential order of information effectively. Based on my findings, it is advisable to avoid employing simple RNNs for time series analysis due to their susceptibility to the vanishing gradient problem and limited capability to capture long-term dependencies. Instead, focus on utilizing more advanced RNN architectures such as LSTM and GRU, which are designed to overcome these challenges. While LSTM is commonly preferred for time series data handling, our experiments suggest that GRU may offer more efficient outcomes. To optimize GRU models, consider fine-tuning hyperparameters such as the number of units in stacked recurrent layers, recurrent dropout rates, and the incorporation of bidirectional data presentation. Additionally, prioritize RNN architectures specifically tailored for sequential data, as our investigation indicates that combining 1D convolution with RNN did not yield satisfactory results. Convolutional approaches tend to disrupt the sequential nature of data, making them less suitable for time series analysis.

**Combination of 1d\_Convent and LSTM model with dropout**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Combination | 16 | Yes | 23.9505 | 3.92 |

A graph of a line graph

Description automatically generated

**Recommendations:**

* Utilizing Mean Absolute Error (MAE) is suitable for time-series data analysis, particularly when the objective involves forecasting continuous numerical values such as temperature.
* The 1D Convolution model exhibits a higher MAE compared to certain RNN models, suggesting that RNNs could be better suited for the provided time-series data.
* The utilization of dropout can aid in mitigating overfitting, as demonstrated by the LSTM model with dropout attaining a decreased MAE and loss.
* The integration of LSTM with dropout and 1D Convolution layers yields the most favourable MAE of 3.92 and a reduced loss function of 23.9505, indicating that this combined approach stands out as a promising contender for temperature forecasting.
* There isn't a consistent performance improvement by increasing the number of dense units in the hidden layers. Sometimes, models with fewer units achieve higher accuracy. It's critical to find a balance and carefully consider the trade-off between model complexity and performance.
* To enhance the precision of temperature predictions, it is recommended to focus on refining the LSTM model with dropout and explore combinations of different architectures, such as integrating LSTM with 1D Convolution. Moreover, given the task's context, prioritizing Mean Absolute Error (MAE) over accuracy is advisable. Further experimentation and meticulous fine-tuning hold potential for optimizing temperature prediction models even further.

**Conclusion:**

After testing different neural network designs for predicting future temperatures using climate data, Out of all the models, stacked versions of GRU and LSTM networks did the best. These models were particularly good at finding hidden patterns in the long-term temperature trends. Additionally, a technique called dropout helped prevent the models from overfitting the data. This whole study using real climate data shows a step-by-step process for designing and testing these neural networks for time series forecasting. The results show that stacked GRU and LSTM models are good at finding complex patterns in climate data, compared to the other models tried.