Time Series Summary

Bharat Chennu
bchennu@kent.edu

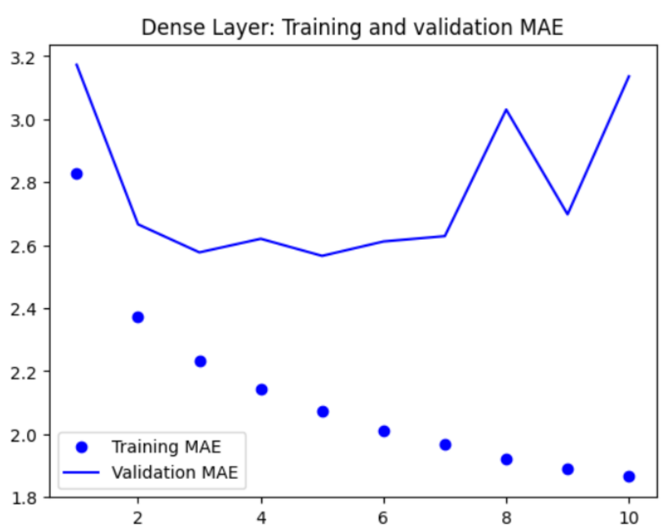
Using this temperature-forecasting exercise, we will demonstrate the key distinctions between timeseries data and the other dataset types we've already worked with. we will observe that convolutional and highly connected networks are insufficient to handle this kind of dataset, while recurrent neural networks (RNNs), a new form of machine learning approach, absolutely thrive at solving this kind of issue.

A total of 50% of the data will be used for training, 25% for validation, and the remaining 25% for testing in all our studies. Since our goal is to predict the future based on the past rather than the contrary, it is crucial to use validation and test data that is more recent than the training data when dealing with timeseries data. The validation/test splits should reflect this.
The following will be the precise formulation of the issue: In a day, is it possible to estimate the temperature based on data collected once an hour for the last five days?

Let us start by preprocessing the data so that a neural network can learn to use it. Simple enough—we do not need to perform any vectorization because the data is already numerical.

The first model relied on common sense methods and yielded a Mean Absolute Error (MAE) of 2.44 as a baseline. When we created a basic machine learning model with a thick layer, the MAE of 2.62 was somewhat higher. The performance of the thick layer model was poor as the time series data was flattened and the temporal context was lost. While not consistently, some of the validation losses are around the no-learning baseline.

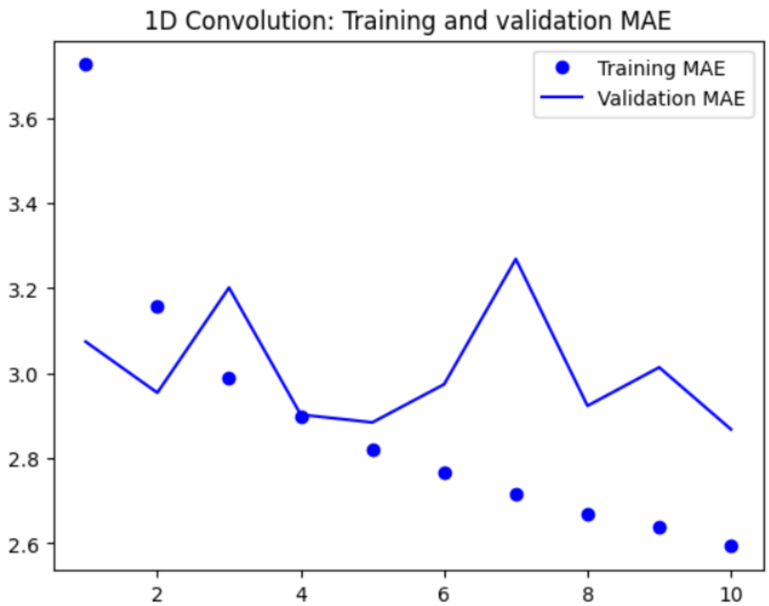
Models :
Basic Dense layer



1D convolutional model

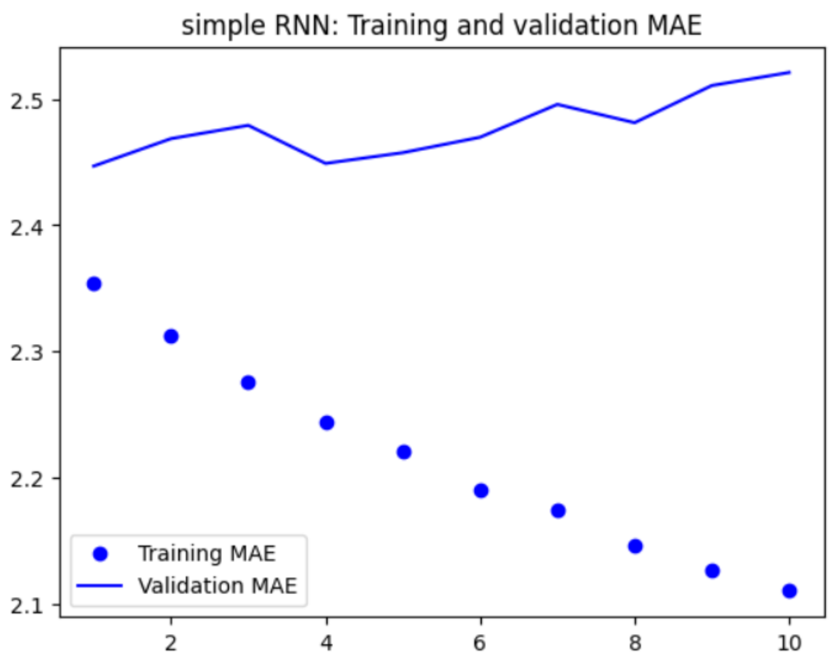
In reference to utilizing appropriate architectural priors, given that the input sequences consist of daily cycles, a convolutional model would be suitable. As a spatial convolutional network may reuse the same representations across multiple places in an image, so too might a temporal convolutional network use the same representations over different days.

Indeed, this model's performance is considerably poorer than that of the densely linked model; it only manages a validation MAE of around 3.09, which is far from the sensible baseline, because not all meteorological data adheres to the translation invariance assumption.



A Simple RNN

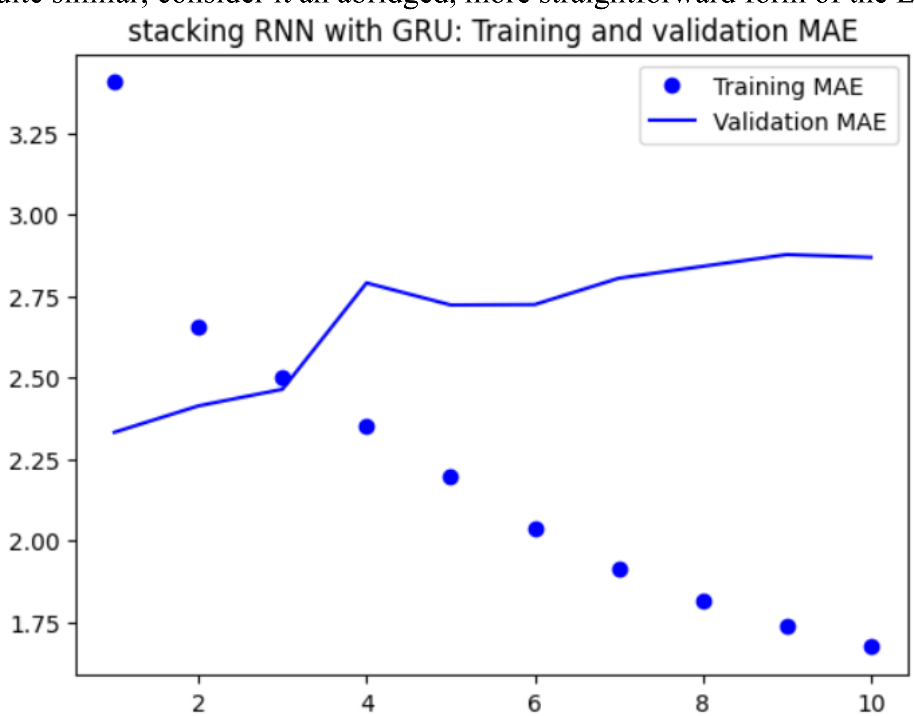
Recurrent neural networks (RNNs) has the exceptional capacity to incorporate historical time step information into present-day decision-making procedures, enabling them to discern complex associations and trends in sequential data. It is possible to describe sequences of different lengths since an RNN's internal state functions as a memory of previous inputs. Practical difficulties arise even though a simple RNN may theoretically preserve data from all prior times. This causes training for deep networks to be challenging due to the vanishing gradient problem. Furthermore, the graph indicates that the simplest RNN performs the poorest out of all of them.



To overcome this problem, as a part of Keras, we must create LSTM and GRU RNNs.

GRU

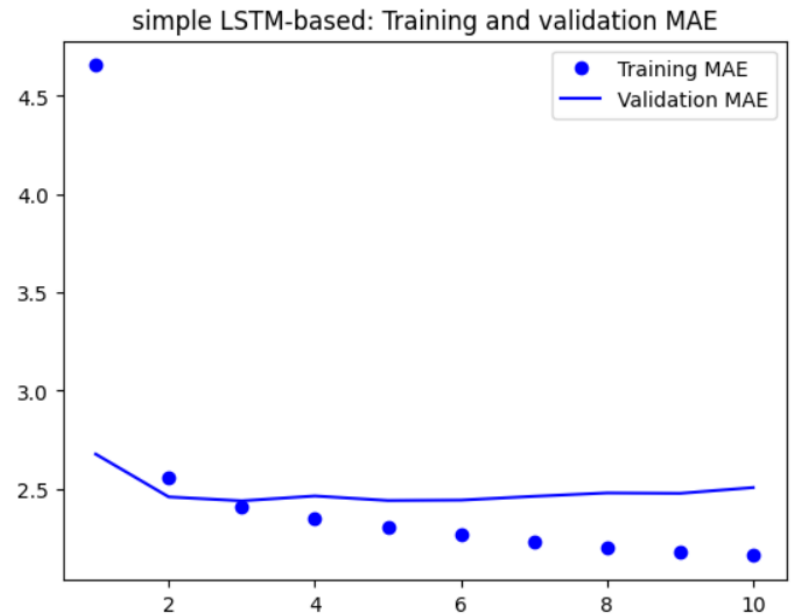
Instead of LSTM layers, we'll utilize Gated Recurrent Unit (GRU) layers. GRU and LSTM are quite similar; consider it an abridged, more straightforward form of the LSTM architecture.



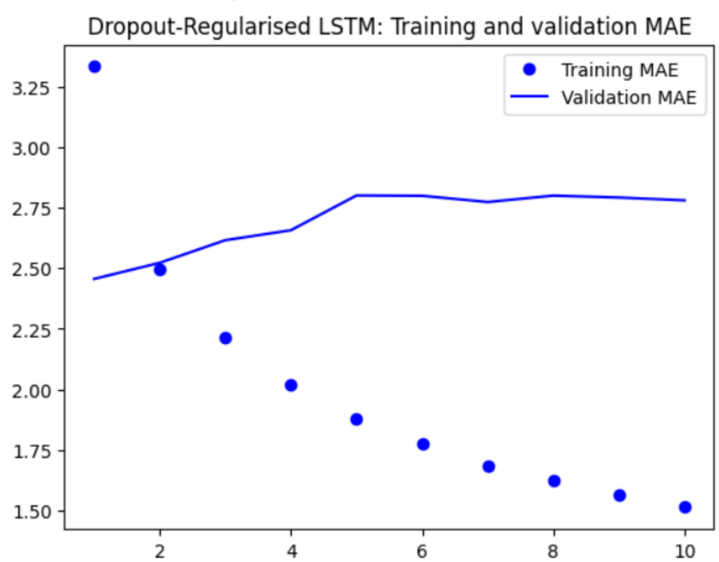
We achieved Test MAE – 2.52 is discovered to be the most effective model, is less computationally costly than Long Short-Term Memory (LSTM) models and effectively captures long-range dependencies in sequential data when compared to the other models.

LSTM

Recurrent neural networks provide a class of neural network topologies particularly created for this use application. One of the most well-liked of them is the Long Short-Term Memory (LSTM) layer. Let us test out the LSTM layer first, and we will see how these models function in a moment. We obtained a test MAE of 2.62.

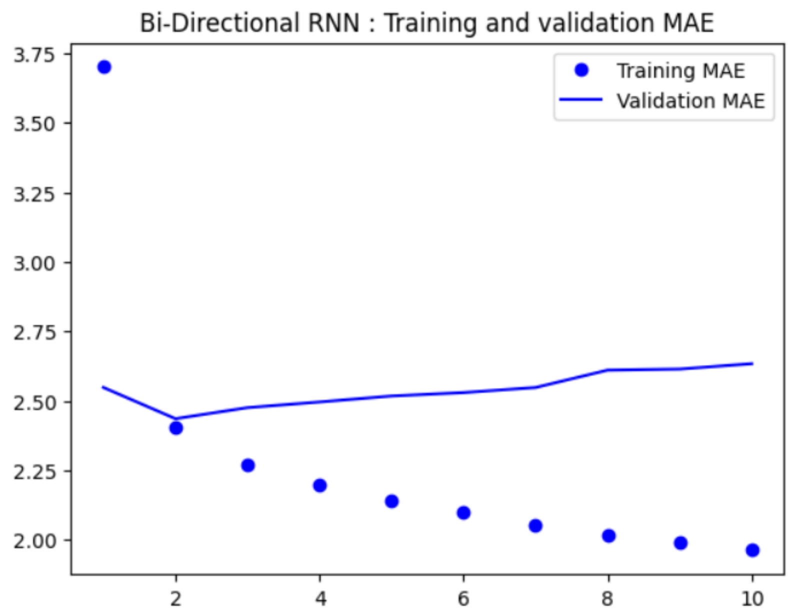


LSTM - dropout Regularization



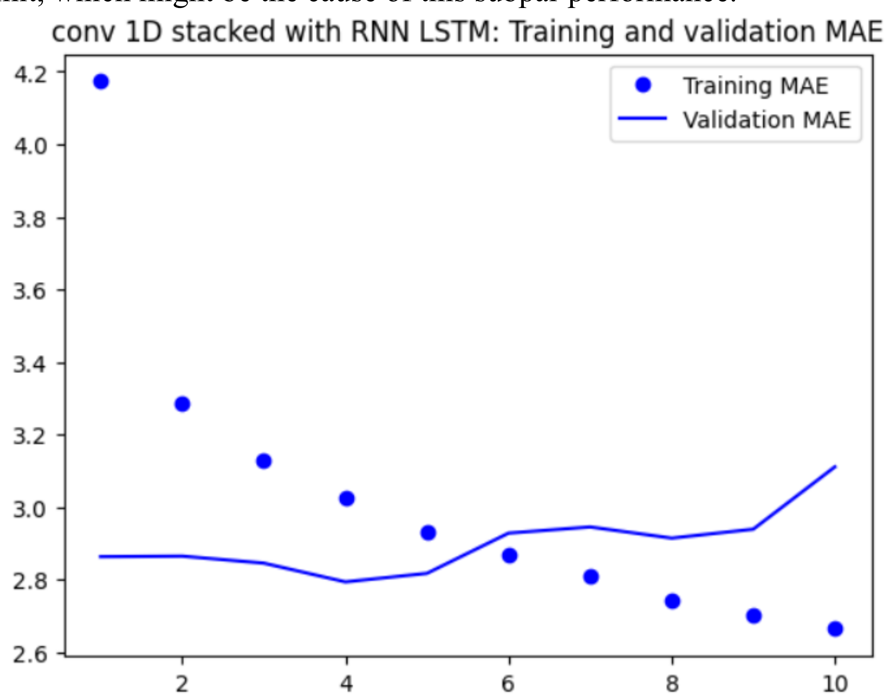
- With LSTM, the test MAE is 2.61 which is little similar to GRU.
- Both LSTM and GRU performed similarly with slight changes.

Bidirectional LSTM



1D Convnets and LSTM together

The model, which I developed using both RNN and 1D convolution, produced inadequate outcomes with 2.93 MAE. The information order may be being destroyed by the convolution limit, which might be the cause of this subpar performance.



Results :

In summary, The information explores a number of neural network topologies, each with special advantages and disadvantages:

MODEL	Test MAE
Dense model	**2.68**
1D convolutional Model	**3.09**
Simple RNN	**2.58**
GRU	**2.52**
LSTM simple	**2.62**
LSTM -dropout Regularization	**2.61**
Bidirectional LSTM	**2.78**
1D convolutional and LSTM	**2.93**

Among the models tested, the stacked GRU and LSTM models are shown to be the best performing architectures, with the lowest test MAE. Their improved performance can be attributed to their capacity to detect long-term dependencies in the time series data as well as the regularisation dropout provides.

Using the Jena Climate dataset as a useful case study, this thorough examination, in its whole, offers a methodical approach to developing and accessing several neural network designs for time series forecasting. The outcomes demonstrate how well stacked GRU and LSTM models perform in comparison to other architectures investigated in identifying complex patterns and connections in the climate data.