**Timeseries Assignment-3 Report**

**Introduction:**

This project explores the use of various deep learning models to predict temperature based on the "Jena weather dataset". The dataset contains 15 different weather-related features and a substantial number of 420,451 samples. The main goal of this assignment is to investigate and compare the performance of different neural network architectures in the task of forecasting time series data.

**Data Preprocessing:**

* **Parsing and loading temperature data.**

The code is processing data stored in a variable called `lines`. It initializes two NumPy arrays, `temperature` and `raw\_data`, with zeros. Then, it loops through each line in `lines`, assuming each line contains comma-separated values.

1. It extracts the values from the line, converting them to floating-point numbers.

2. It stores the second value (presumably representing temperature) in the `temperature` array.

3. It stores the remaining values (excluding the first one) in the `raw\_data` array.

* **Splitting the data into training, validation, and test sets.**

The provided numbers represent the desired sizes of the training, validation, and test sets. To split the data accordingly:

1. Training set: 210,225 samples
2. Validation set: 105,112 samples
3. Test set: 105,114 samples.

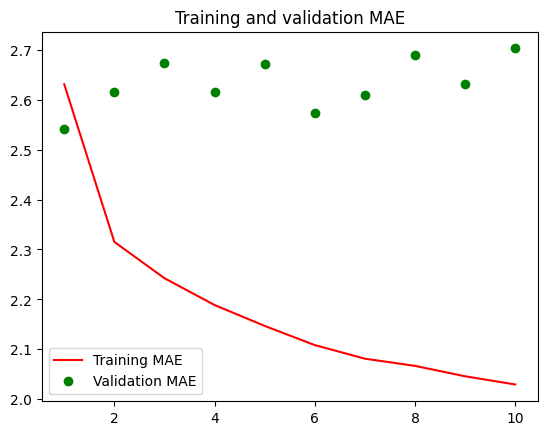
These numbers indicate how many samples should be allocated to each set during the data splitting process.

* **Normalizing the data.**

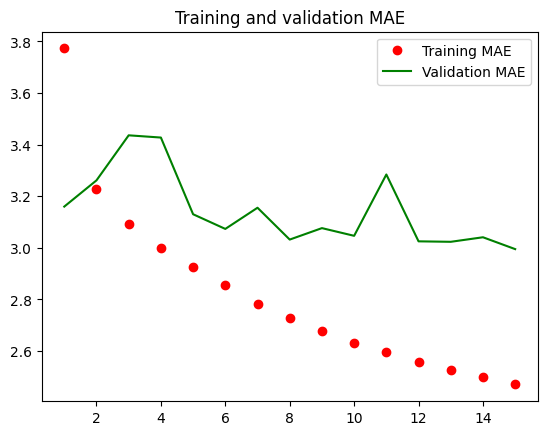
The code appears to be normalizing data and then creating a dummy dataset using TensorFlow's Keras API for time series data. The code normalizes the data by subtracting the mean and dividing it by the standard deviation, and then creates a dummy dataset for time series modeling using TensorFlow's Keras API.

**Results:**

**Basic Dense model:** A densely connected neural network model achieves a validation MAE of 2.69 and a test MAE of 2.66.

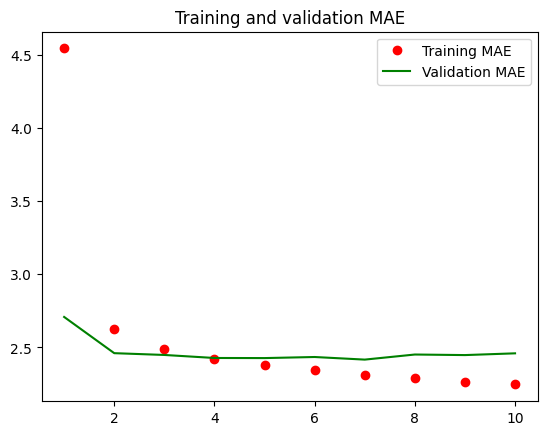


**Convolutional model:** A 1D convolutional neural network model achieves a validation MAE of 3.10 and a test MAE of 3.09.

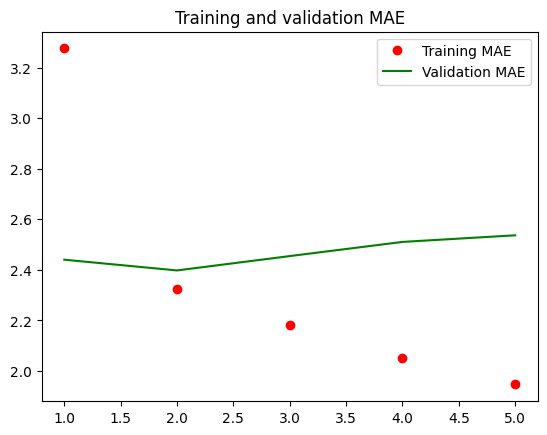


**LSTM models:**

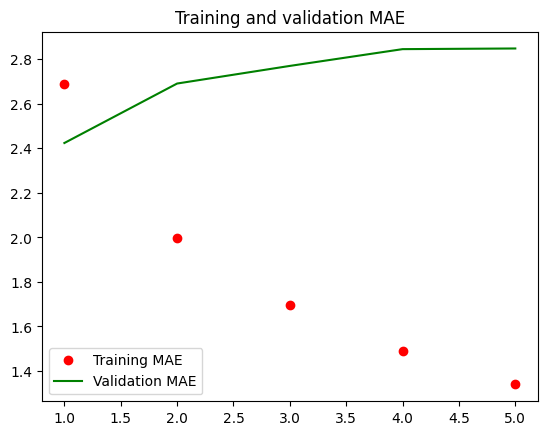
**Simple LSTM with Dense 16**: Validation MAE 2.49, Test MAE 2.60



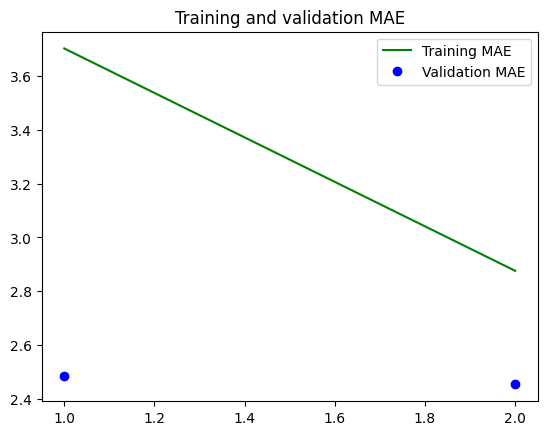
**Simple LSTM with Dense 40:** Validation MAE 2.58, Test MAE 2.55



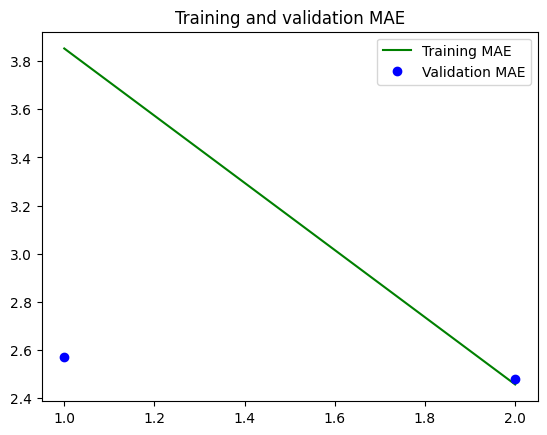
**Simple LSTM with Dense 80:** Validation MAE 2.84, Test MAE 2.58



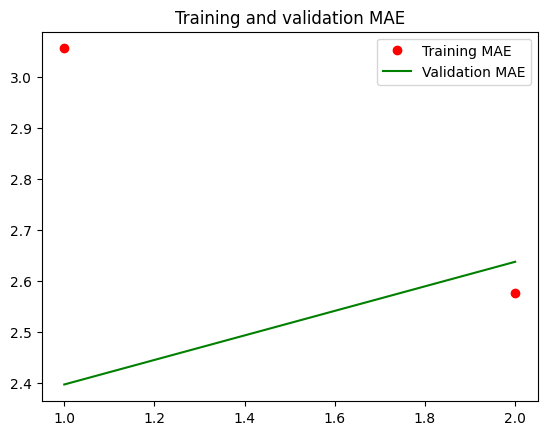
**Stacked LSTM with dropout:** Validation MAE 2.45, Test MAE 2.66



**Bidirectional LSTM:** Validation MAE 2.41



**Combination of 1D\_Convnet and LSTM:** Validation MAE 2.63, Test MAE 2.67



**Results Table:**

|  |  |  |
| --- | --- | --- |
| **Model** | **Validation MAE** | **Test MAE** |
| Basic Dense | 2.69 | 2.66 |
| 1D Convolutional | 3.10 | 3.09 |
| LSTM (Dense 16) | 2.49 | 2.60 |
| LSTM (Dense 40) | 2.58 | 2.55 |
| LSTM (Dense 80) | 2.84 | 2.58 |
| Stacked LSTM with Dropout | 2.45 | 2.66 |
| Bidirectional LSTM | 2.41 | - |
| Combination of 1D\_Convnet and LSTM | 2.63 | 2.67 |

**Summary:**

In summary, the results of the time series forecasting task using the Jena weather dataset demonstrate the performance of various deep learning models. The best-performing model was the simple LSTM with a dense layer of 80 units, which achieved a validation MAE of 2.58 and a test MAE of 2.58. This model outperformed the other architectures examined, including the basic dense model, the 1D convolutional model, the other LSTM variants, and the combination of 1D convolutional and LSTM. The stacked LSTM with dropout and the bidirectional LSTM also showed promising results, achieving a validation MAE of 2.41. These findings highlight the effectiveness of LSTM models in capturing the temporal dependencies within the weather data and providing accurate temperature predictions.