Assignment 3 – Time Series

Objective - The objective is to explore the application of Recurrent Neural Networks (RNNs) to time-series data, investigate strategies for improving the performance of RNN-based models when dealing with time-series data, and examine the application of various deep learning layers to enhance the analysis of time-series data.

Model Building – In total we have built total of 15 models, we are training and evaluating a densely connected neural network model for time series forecasting. The model is trained on the Jena temperature dataset, and the MAE is used as the evaluation metric.

The model has a single input layer with 14 features (the number of features in the Jena temperature dataset), a single hidden layer with 16 units, and a single output layer with 1 unit (the target variable, which is the temperature).

The results show that the **densely connected neural network** model outperforms the common-sense baseline method on both the validation and test datasets. The MAE of the neural network model is 2.67 on the test dataset, while the MAE of the common-sense baseline method is 2.62.

In this case, it is likely that the dense layer model is overfitting the training data. This is because the dense layer model has a relatively large number of parameters (16 units in the hidden layer) and a relatively small amount of training data (819 samples).

In order to reduce the problem of Overfitting we need to proceed with regularization techniques, such as dropout or L1/L2 regularization.

The **1D convolutional neural network** (CNN) model with 10 epochs achieved a test MAE of 3.12. This is slightly higher than the test MAE of the densely connected neural network model (2.67). This might be because of pooling layers in CNNs are used to reduce the dimensionality of the data and to extract spatial features. However, pooling layers can remove long-term dependencies in time series data.

RNNs are a suitable choice for time series data because they are specifically designed to handle sequences and can capture the temporal relationships within the data. **Simple RNN** model achieved a test MAE of 9.92. This is significantly higher than the test MAE of the densely connected neural network model (2.67) and the CNN model (3.12). training results indicate that the model's validation MAE decreases slightly over the epochs, but the final test MAE is around 9.92 suggesting that the model might not be performing well on the task. This is the least performing model among the 15 due to their vanishing gradient" problem, which makes them less effective for many sequential data tasks compared to more advanced RNN variants like LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit).

Adding more layers to a neural network can potentially increase its representational capacity and help it learn more complex patterns, it appears that the addition of **stacked Simple RNN** layers did not lead to significant improvement in performance. The mean absolute error (MAE) on the test dataset remains relatively high at around 9.91. This suggests that the model might be struggling to capture the underlying patterns in the data.

Simple GRU model is a basic recurrent neural network (RNN) architecture that uses a single GRU layer. The LSTM-Simple model is a more complex RNN architecture that uses a single LSTM layer. The LSTM (RNN)

- dropout model is a regularized version of the LSTM-Simple model that uses dropout regularization to prevent overfitting.

All three models achieve similar test MAE scores, but the LSTM-Simple model takes slightly longer to train. The LSTM (RNN) - dropout model takes even longer to train, but it achieves the best test MAE score.

Now we have been trying to evaluate the performance by number of units in the stacked LSTM layers. The 4 models you have trained are all LSTM models with different numbers of units. The models with more units (32 and 64) have highest MAEs than the models with less units (8 and 16). This suggests that the data is relatively

Bidirectional RNNs were employed to capture dependencies in both the forward and backward directions of the input sequence. These models consistently exhibited lower Mean Absolute Error (MAE) values compared to a common-sense baseline model, demonstrating their effectiveness in time series forecasting.

Combined a convolutional model with an RNN, resulting in a hybrid model, yielded unsatisfactory results with a high MAE. This suboptimal performance was attributed to the inherent limitation of convolutional approaches, which disrupt the sequential order of information within the data. This highlights the importance of choosing architectures that are specifically designed for handling sequential data, such as RNNs.

Model combines 1D convolutional layers and an LSTM layer. The convolutional layers are used to capture local patterns in the input sequence, followed by an LSTM layer for capturing longer-term dependencies.
The test MAE of 3.90 is higher compared to the bidirectional LSTM model. This could be due to the complexity introduced by the combination of convolutional layers and LSTM

LSTM and GRU models are better suited for time series forecasting than simple RNN models due to their ability to capture long-term dependencies in data. A hybrid model combining a convolutional model with an RNN did not perform well, likely due to the disruptive effect of the convolutional approach on the sequential order of information in time series data.