**Time-Series Data**

Name: Priyanka chinthakuntla

Student Id: 811317866

**Introduction:**

The goal of this is temperature weather forecasting using time series data in conjunction with RNNs. This homework compares several neural network architectures such as LSTM, GRU, and Conv1D/LSTM-based models by trying to figure out which of the models does better in predictability.

Background in the Time Series Data:

Time-series forecasting is one of the most important tasks in several domains, such as weather prediction, where one must forecast future values based on historical patterns. In fact, this will turn out to be a natural fit for the RNNs because they store temporal information quite well from the previous values, which are crucial to predict sequences over time correctly.

**Dataset and Preprocessing:**

The dataset I used in this project is weatherHistory.csv, which contains historical weather data, including features such as:

* Temperature (C): Actual temperature in degrees Celsius.
* Humidity: Relative humidity.
* Wind Speed (km/h): Wind speed in kilometers per hour.
* Pressure (millibars): Atmospheric pressure.

**Data Preprocessing Steps:**

1. Feature Scaling and Selection: Data is loaded into the program, and features are filtered to retain only the relevant features about forecasting - Temperature, Humidity, Wind Speed, and Pressure.
2. Handling Missing Values: The missing value is filled by forward filling so that continuity in the data can be maintained.
3. Normalization: This will normalize the features in a range between 0 and 1 using MinMaxScaler; this increases the performance of the model by reducing the variance among the feature values.
4. Sequence Generation: The length of a sequence is 60-time steps; therefore, each input fed to the model is a 60-step sequence in generating the next temperature value.

**Methodology:**

Model Configurations: In this project, three different models were being compared. These are LSTM, GRU, and Conv1D & LSTM. As one will notice, each model in its configuration is designed as follows:

* LSTM Model: It is a stacked LSTM having two LSTM layers followed by dropout for regularization against overfitting.
* The GRU Model: Configuration This is very much the same as in the LSTM model, but in the layer configuration, it now uses GRU layers, which are, at least in computation, more efficient and might give similar accuracy.
* Conv1D + LSTM Model: This model extracts features from the input by using the Conv1D layer, followed by dense layers. The Conv1D layer can extract the local pattern, which may be helpful in enhancing the performance of LSTMs.

**Hyperparameters and Training:**

* **Epochs**: 25 epochs with early stopping to prevent overfitting.
* **Batch Size**: 32, which balances computation speed and memory usage.
* **Early Stopping**: Used to stop training if the validation loss does not improve over 10 consecutive epochs.
* **Evaluation Metrics**: The models are evaluated on Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), with lower values indicating better performance.

**Results:**

**Evaluation Metrics:** Each model is evaluated based on MAE and RMSE. Here’s a summary of the results:

|  |  |  |
| --- | --- | --- |
| Model | MAE | RMSE |
| LSTM | 0.830 | 1.2992 |
| GRU | 1.3865 | 1.83681 |
| Conv1d+LSTM | 1.4780 | 2.01400 |

**Screenshots:**

Temperature Forecasting with LSTM:

A graph showing the temperature of a weather forecast

Description automatically generated

Temperature Forecasting with GRU:

A graph showing the weather forecast

Description automatically generated

Temperature Forecasting with Conv1D+LSTM:

A graph showing the temperature of a temperature

Description automatically generated with medium confidence

Each model has plotted the actual temperature values against the predicted values generated. The visualizations show the closeness of the predictions for each model against real data, highlighting the patterns and inaccuracies.

**Conclusion:**

This proved that the RNNs were indeed a good architecture, particularly the LSTM and GRU, for weather time-series forecasting. Each of these configurations has their advantages: LSTM and GRU worked perfectly for sequence-based predictions, while the Conv1D + LSTM contributed useful feature extraction capabilities.

The best model, based on the lowest MAE, was the LSTM which is 0.830