Time-Series Forecasting using RNN

1. **Provide brief details about the nature of your dataset. What is it about? What type of data are we encountering? How many entries and variables does the dataset comprise?**

Dataset : [Individual household electric power consumption]

The dataset provides a comprehensive view of household power consumption:

1. Nature of the Dataset:
    a. The dataset is a collection of measurements related to power consumption in a household.
    b. It includes various metrics such as active power, reactive power, voltage, current intensity, and active energy consumption for different appliances or areas within the household.
2. Type of Data: The dataset primarily consists of numerical data types (float64), which are used to represent measurements of power consumption and related metrics.
3. Entries and Variables:
    a. The dataset comprises 113,606 entries (rows) and 7 variables (columns).
    b. Each entry corresponds to a specific timestamp, indicating when the measurements were taken, and each variable represents a different aspect of power consumption or related metrics.
    c. Detailed description of the dataset:
        i. Global Active Power: This column represents the total active power consumed by the household.
        ii. Global Reactive Power: The global reactive power column indicates the total reactive power consumed by the household.
        iii. Voltage: This column provides the average voltage level in the household.
        iv. Global Intensity: Global intensity represents the average current intensity in the household.
        v. Sub-metering 1, 2, 3: These columns represent the active energy consumption for specific areas or appliances within the household. Sub-metering 1 is for the kitchen, sub-metering 2 is for the laundry, and sub-metering 3 is for climate control systems.

Time-series forecasting using Recurrent Neural Networks (RNNs) on the Global Active Power variable is essential for forecasting future global active power consumption helping in better energy management.

2. Provide the details related to your RNN architecture.

The RNN architecture used in this example consists of three LSTM layers followed by four fully connected (linear) layers.

Breakdown of the architecture:

1. LSTM Layers:
   a. The first LSTM layer (`self.lstm1`) takes an input of `input_size` and has a hidden size of `hidden_size` with `num_layers` layers. It is bidirectional, meaning it processes the input sequence in both forward and backward directions.
   b. The output of the first LSTM layer is passed to the second LSTM layer (`self.lstm2`), which has the same `hidden_size` and `num_layers` but takes the output of the first layer as input.
   c. Similarly, the output of the second LSTM layer is passed to the third LSTM layer (`self.lstm3`), which has a `hidden_size` of 32 and `num_layers` layers, and is also bidirectional.
2. Fully Connected Layers:
   a. After the LSTM layers, the output is reshaped to select only the output of the last time step (`out[:, -1, :]`).
   b. The reshaped output is then passed through four fully connected layers (`self.fc1` to `self.fc4`) with decreasing hidden units (128, 64, 32, and 1 for the output layer).
   c. Each fully connected layer is followed by a ReLU activation function (`self.relu`) and a dropout layer with a dropout probability of 0.25 (`self.dropout`).
3. Activation:
   The final output is passed through a sigmoid activation function (`torch.sigmoid`) to ensure the output is between 0 and 1, suitable for regression tasks.

This architecture is designed for time-series forecasting task, where the model learns to predict the next value in a sequence based on past observations.

3. **Discuss the results and provide relevant graphs:**

   i. **Report training accuracy, training loss, validation accuracy, validation loss, testing accuracy, and testing loss.**
   Note:
   For regression or time series forecasting tasks, accuracy is not typically used as a metric because it is more suitable for classification tasks

   We have focused on MAE as the primary metric for evaluating the model's performance on the validation and test sets. However, to get test accuracy, we have also calculated the accuracy for the test set by applying a threshold to the predicted values.

   Training MAE: 0.2054
   Training loss: 0.0954
   Validation MAE: 0.1324
   Validation loss: 0.1324
   Testing MAE: MAE: 0.2045
   Testing loss: Test Loss: 0.0919
   Test Accuracy: 78.2585%

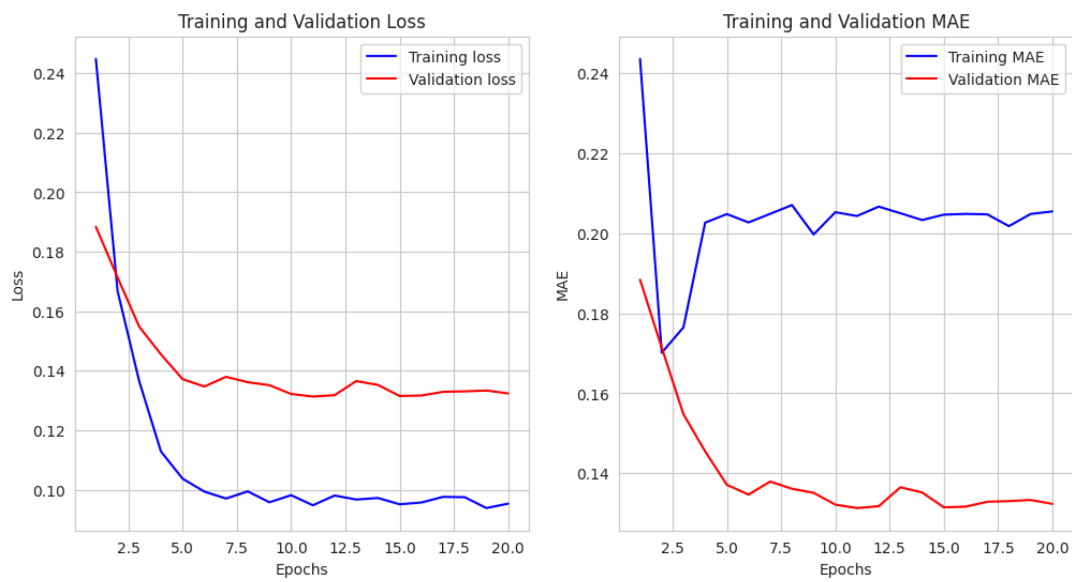
   Overall, the model demonstrates good performance on both the validation and test sets, with MAE values indicating accurate predictions.

   The test accuracy of 78.2585% indicates that the model performs well in predicting the target variable compared to a baseline.

   ii. **Plot the training and validation accuracy over time (epochs).**

   iii. **Plot the training and validation loss over time (epochs).**
   Accuracy is not suitable for time series forecasting tasks because it does not consider the temporal dependencies, imbalance, error sensitivity, and the need to capture trends and seasonality in the data.
   Mean Absolute Error (MAE) is often used to measure the performance of time series forecasting models for several reasons:
   1. Interpretability: MAE provides a clear average of prediction errors, making it easy to understand the model's accuracy.

2. Robustness to Outliers: It is less affected by extreme values in the data compared to other metrics like MSE.
3. Scale Independence: MAE treats errors equally regardless of the scale of the target variable, allowing for fair comparisons between models.
4. Loss Function: MAE can be used as a loss function for training models, focusing on minimizing absolute errors.



Observation:

1. Epochs 1-4: The training and validation loss decrease steadily, indicating that the model is learning and improving its performance. The MAE also decreases, showing that the model's predictions are becoming more accurate.
2. Epochs 5-9: Both the training and validation loss continue to decrease, but the rate of decrease slows down compared to earlier epochs. The MAE decreases as well, but again, the rate of improvement slows down.
3. Epochs 10-13: The training loss remains relatively stable, while the validation loss starts to fluctuate slightly, potentially indicating that the model is starting to overfit the training data. The MAE shows a similar trend, with a slower rate of improvement.
4. Epochs 14-20: The training and validation loss fluctuate, and the MAE also shows fluctuations. This behavior could be due to the model overfitting the training data, leading to reduced performance on the validation set.

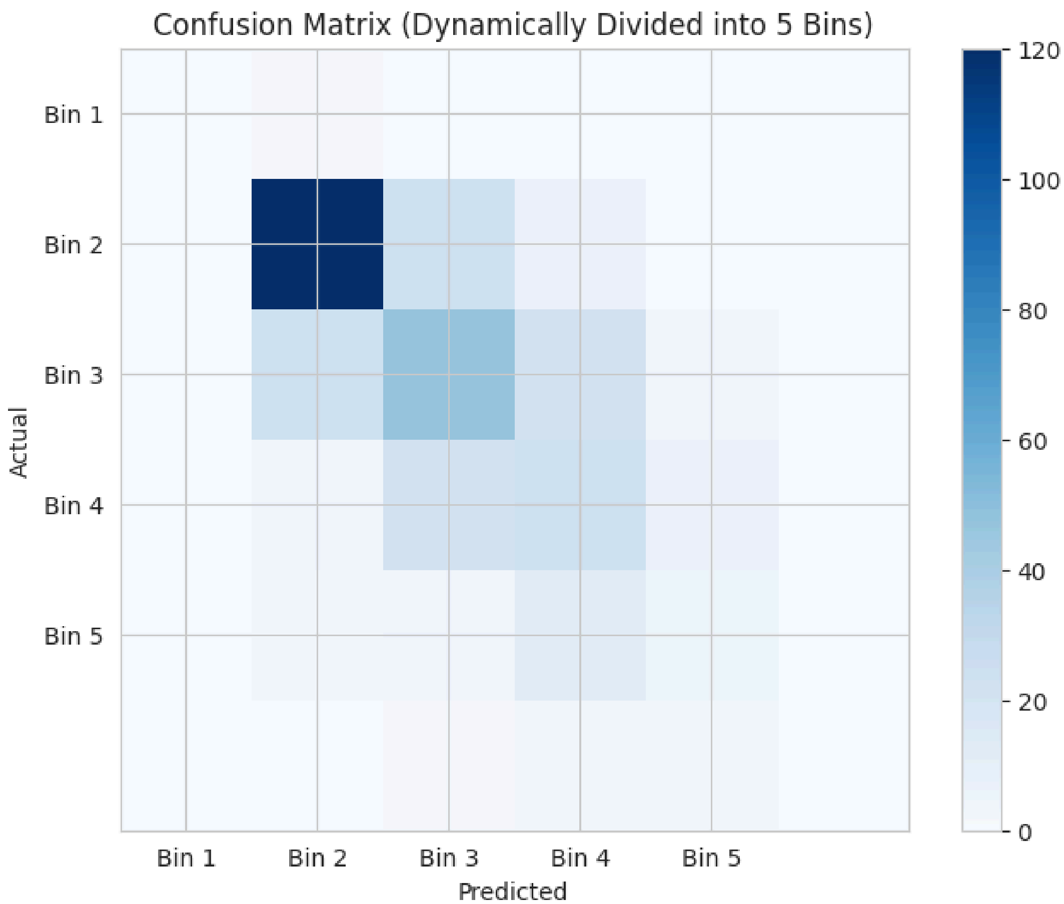
5. Early stopping: The training is stopped early.

Overall, the trend suggests that the model initially learns and improves its performance but starts to overfit after a certain number of epochs, leading to a decrease in generalization performance on the validation data.
Early stopping is applied to prevent further overfitting and to select the model with the best performance on the validation set.

iv. Generate a confusion matrix using the model's predictions on the test set.

Confusion matrices are not applicable to regression tasks because they are designed for evaluating classification models, which predict discrete classes, whereas regression tasks involve predicting continuous values.

However, it's possible to create a confusion matrix by binning the predicted values into discrete intervals or bins and comparing them to the actual values:



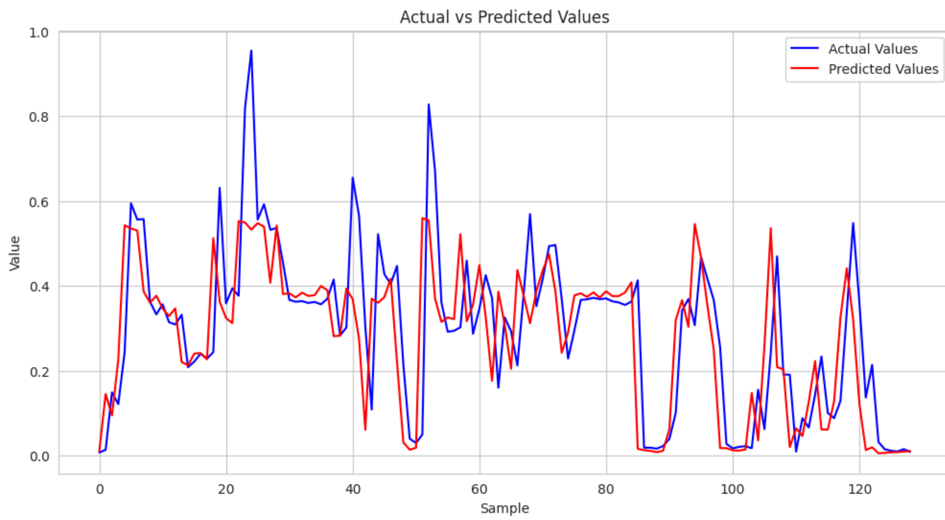
v. **Report any other evaluation metrics used to analyze the model's performance on the test set.**

Mean Absolute Percentage Error (MAPE) : metric used to evaluate the accuracy of a forecasting model. It measures the average percentage difference between the predicted values and the actual values.

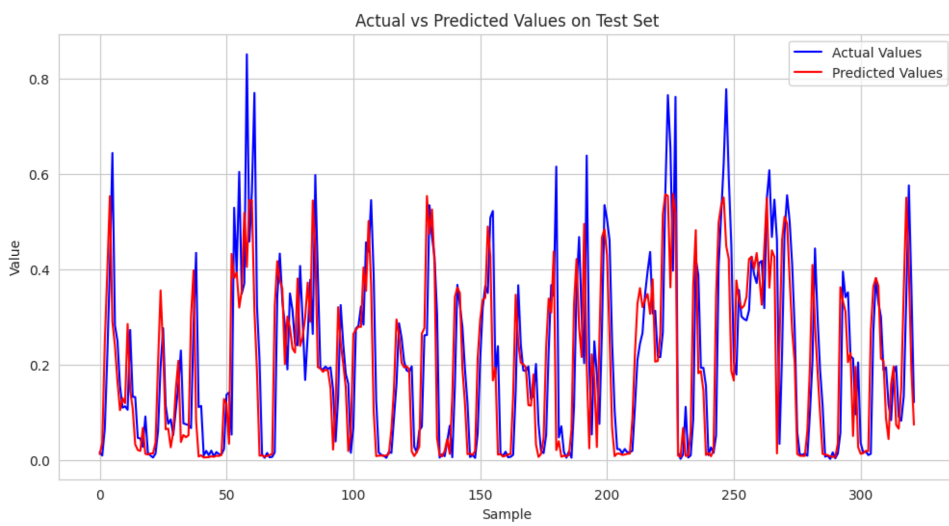
MAPE: 323.9051%

This value suggests that the model's performance is making predictions but not very good.

Validation Data Prediction Trend:



Test Data Prediction Trend:



The graphs indicate that the trained model has effectively learned the underlying patterns in the data. This is evident from the close alignment between the predicted values and the actual values in both the validation and test datasets, showing that the model's predictions closely follow the trend of the actual values.