Air Quality Prediction

Introduction:

Contains the responses of a gas multisensor device deployed on the field in an Italian city. Hourly responses averages are recorded along with gas concentrations references from a certified analyzer.

The Dataset:

The dataset contains 9358 instances of hourly averaged responses from an array of 5 metal oxide chemical sensors embedded in an Air Quality Chemical Multisensor Device. The device was located on the field in a significantly polluted area, at road level,within an Italian city. Data were recorded from March 2004 to February 2005 (one year)representing the longest freely available recordings of on field deployed air quality chemical sensor devices responses. Ground Truth hourly averaged concentrations for CO, Non Metanic Hydrocarbons, Benzene, Total Nitrogen Oxides (NOx) and Nitrogen Dioxide (NO2) and were provided by a co-located reference certified analyzer. Evidences of cross-sensitivities as well as both concept and sensor drifts are present as described in De Vito et al., Sens. And Act. B, Vol. 129,2,2008 (citation required) eventually affecting sensors concentration estimation capabilities. Missing values are tagged with -200 value.

Dataset URL: https://archive.ics.uci.edu/dataset/360/air+quality

Features:

- Date (DD/MM/YYYY)
- Time (HH.MM.SS)
- PT08.S1 (tin oxide) hourly averaged sensor response (nominally CO targeted)
- True hourly averaged overall Non Metanic HydroCarbons concentration in microg/m^3 (reference analyzer)
- True hourly averaged Benzene concentration in microg/m^3 (reference analyzer)
- PT08.S2 (titania) hourly averaged sensor response (nominally NMHC targeted)
- True hourly averaged NOx concentration in ppb (reference analyzer)
- PT08.S3 (tungsten oxide) hourly averaged sensor response (nominally NOx targeted)
- True hourly averaged NO2 concentration in microg/m^3 (reference analyzer)
- PT08.S4 (tungsten oxide) hourly averaged sensor response (nominally NO2 targeted)
- PT08.S5 (indium oxide) hourly averaged sensor response (nominally O3 targeted)
- Temperature in Â°C
- Relative Humidity (%)
- AH Absolute Humidity

Target Variable:

- True hourly averaged concentration CO in mg/m^3 (reference analyzer)

Data Preprocessing:

Before employing model training, it's crucial to preprocess the data, ensuring its quality and format align with the requirements of the modeling process. The following steps were undertaken:

- ***Date-Time Conversion:***

  The 'Date' and 'Time' columns in the dataset are first converted to string data type to ensure they're in a format suitable for concatenation. A new 'DateTime' column is created by combining the 'Date' and 'Time' columns. The combined strings are converted to a DateTime object with a specified format of '%Y-%m-%d %H:%M:%S'.

- ***Data Cleaning:***

  All instances of the value -200 in the dataset are replaced with NaN (Not a Number). This suggests that -200 was used in the original dataset as a placeholder for missing or unrecorded data. Missing values in the dataset (now represented by NaN) are interpolated. Interpolation infers the missing value based on surrounding data. It's a common technique for filling in gaps in time-series data.

- ***Data Normalization:***

  The data is scaled to lie between 0 and 1 using the MinMaxScaler from the Scikit-learn library. Normalization is important for certain machine learning models that are sensitive to the scale of input data. After scaling, the normalized data is put back into a Pandas DataFrame with the same column names as before.

- ***Creating Sequences:***

  A function create_sequences is defined to convert the dataset into overlapping sequences of a specified length (in this case, 24). This is useful for problems like sequence prediction where we want to use a sequence of past observations to predict the next observation.

Model Architecture:

1. Feedforward Neural Network (FFN):
   - Input layer: 64 neurons with ReLU activation.
   - Dropout layer with a rate of 0.2.
   - Hidden layer: 32 neurons with ReLU activation.
   - Output layer: 1 neuron for regression output.

2. Long Short-Term Memory (LSTM):
   - First LSTM layer: 64 neurons with ReLU activation and returns sequences.
   - Second LSTM layer: 32 neurons with ReLU activation.
   - Dropout layer with a rate of 0.2.
   - Output layer: 1 neuron for regression output.

3. Gated Recurrent Units (GRU):
    - First GRU layer: 64 neurons with ReLU activation and returns sequences.
    - Second GRU layer: 32 neurons with ReLU activation.
    - Dropout layer with a rate of 0.2.
    - Output layer: 1 neuron for regression output.

Experimental Setup:

- Optimization Algorithm: Adam optimizer.
- Loss Function: Mean Squared Error.
- Used ReLU in the hidden layer and sigmoid in the output layer.
- Number of Epochs: 50.
- Batch Size: 32

Results & Discussion:

The performance of the three neural network architectures — Feedforward Network (FFN), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU) — was evaluated using Mean Absolute Error (MAE) and Mean Squared Error (MSE). The following are the obtained results:

Model	Mean Absolute Error (MAE)	Mean Squared Error (MSE)
Feedforward Network	0.0920	0.0138
LSTM Network	0.0662	0.0073
GRU Network	0.0839	0.0110

1. **Performance Evaluation:**

- The LSTM Network outperformed the other models, yielding the lowest MAE of 0.0662 and the lowest MSE of 0.0073. This underscores the LSTM's ability to remember long sequences and dependencies, which is essential for time series data, making it a suitable choice for this dataset.
- The Feedforward Network had the highest error metrics among the three models, with an MAE of 0.0920 and an MSE of 0.0138. This could be attributed to its architecture, which doesn't inherently have a memory component for sequence data. The feedforward network might not capture time dependencies as effectively as the LSTM or GRU architectures.
- The GRU Network showed performance in between the LSTM and Feedforward models, with an MAE of 0.0839 and an MSE of 0.0110. GRUs, although designed to solve the vanishing gradient problem like LSTMs, have a simpler architecture, which can sometimes perform at par with or better than LSTMs depending on the dataset's complexity.

2. **Model Complexity and Data Characteristics:**

- The superior performance of the LSTM suggests that capturing temporal dependencies in the data is crucial for making accurate predictions. Given that LSTMs are designed to recognize long-term patterns, the data might contain such patterns that are beneficial for the prediction of CO(GT).

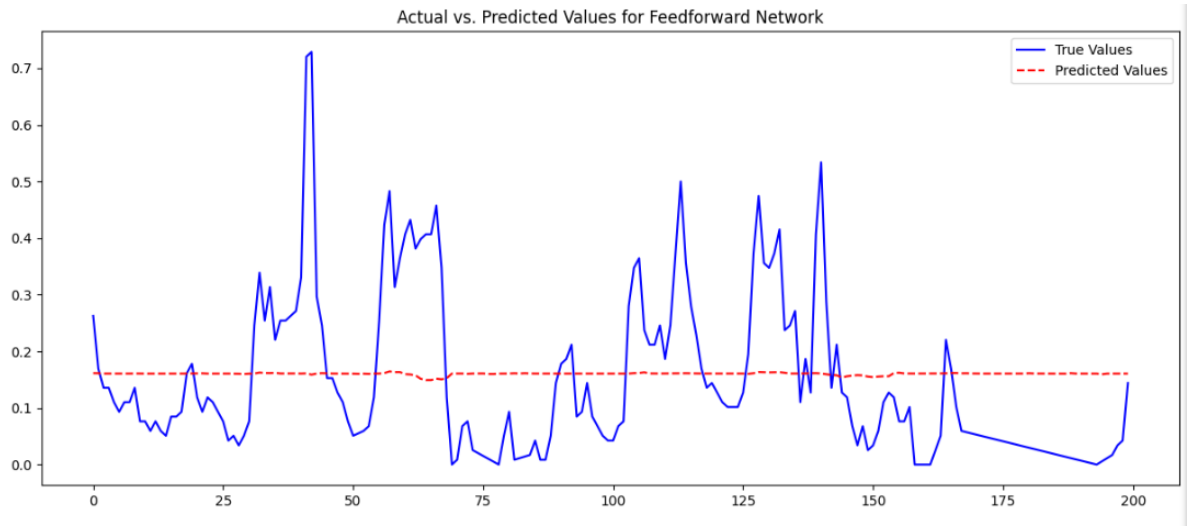
- GRUs, while also designed to capture temporal dependencies, differ in architecture from LSTMs. The slightly inferior performance of GRUs in comparison to LSTMs in this context may suggest that certain LSTM-specific characteristics are more suitable for this data.
- Feedforward Networks do not have any memory of previous inputs. The considerably higher error from the FFN indicates that purely relying on current input without any memory might not be sufficient for accurate predictions in this dataset.

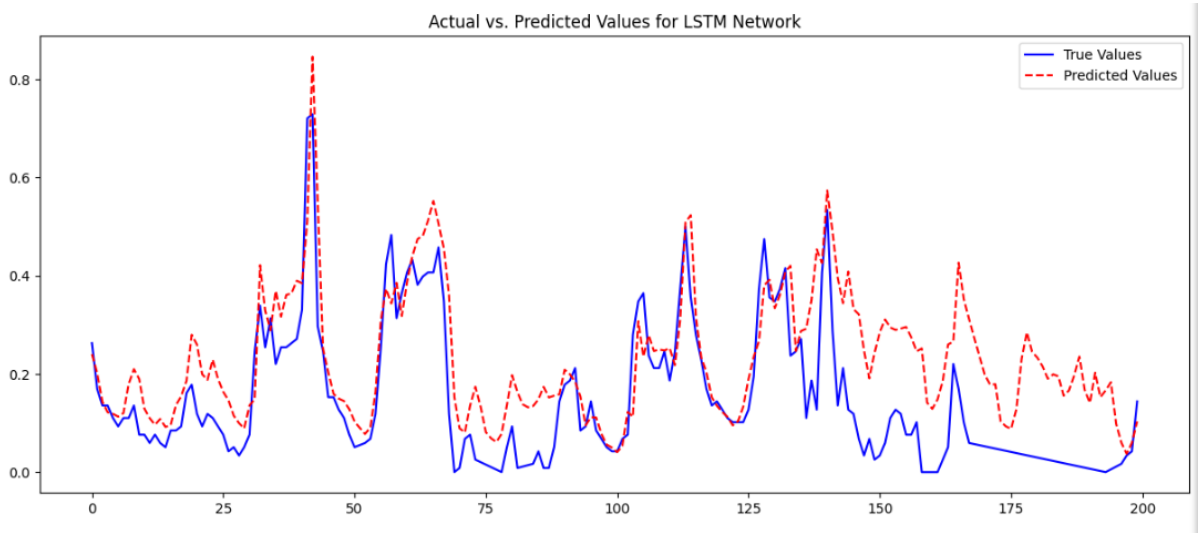
3. Potential Improvements:

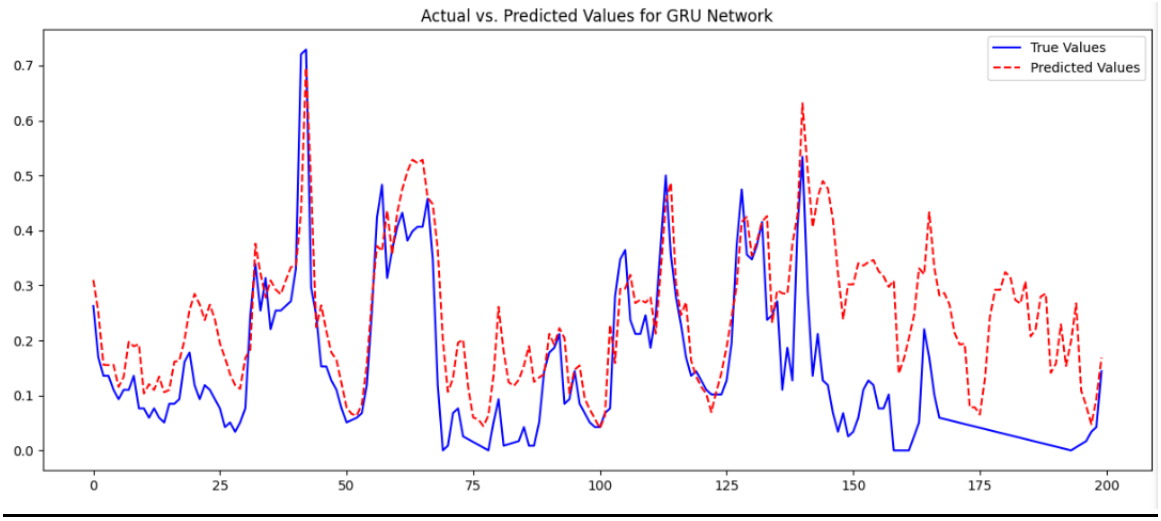
- Hyperparameter tuning, including altering the number of layers, neurons, and learning rates, could further optimize the performance of each model.
- Advanced regularization techniques might also help in achieving better generalization and potentially better validation results.
- Experimenting with different sequence lengths can provide insights into the optimal amount of historical data required for predictions.

Visualization:

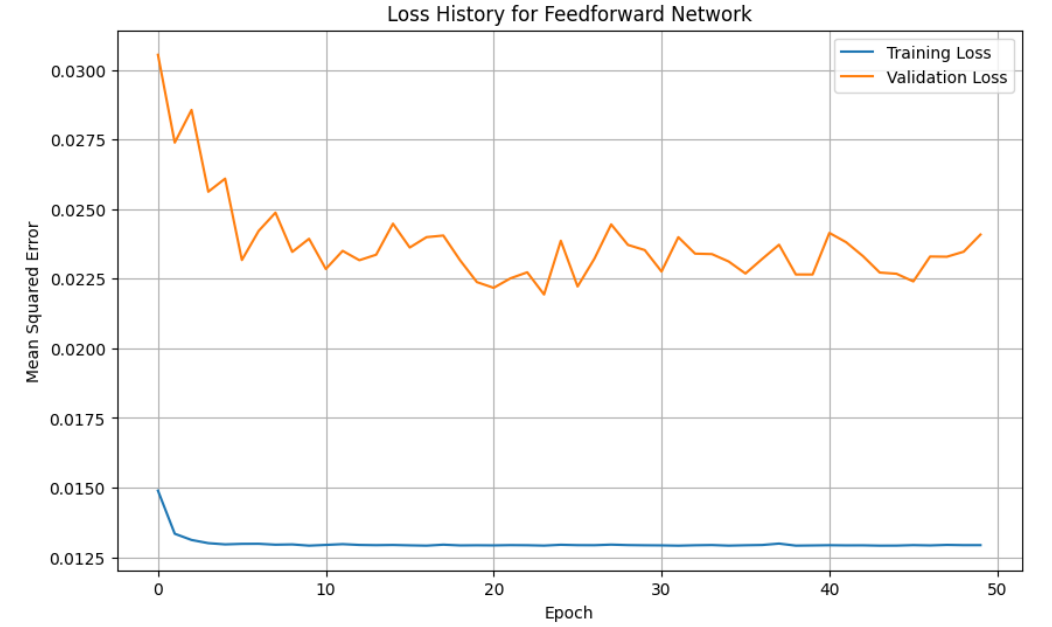
1. Actual vs Predicted Values:

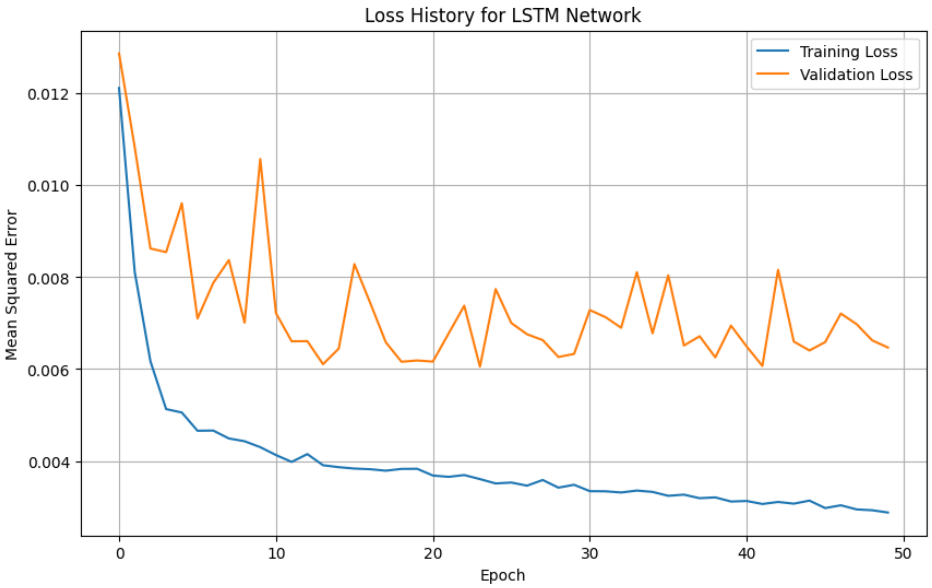


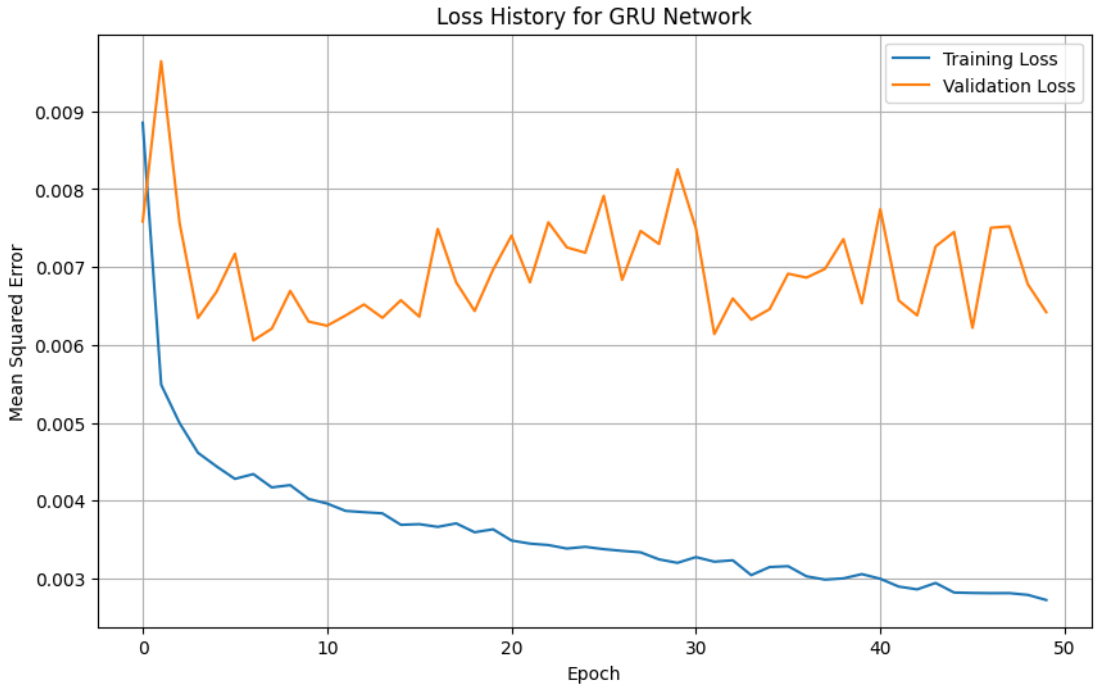




2. Loss History:







Conclusion:

In our endeavor to predict air quality metrics using deep learning models, we have demonstrated the efficacy of three distinct neural network architectures: Feedforward Networks, LSTM Networks, and GRU Networks. Our results indicate a clear superiority of the LSTM network for this particular task, emphasizing its adeptness in handling time series data by capturing long-term dependencies.

The comparative analysis showed that not all neural architectures are created equal, especially when handling sequential data. The Feedforward Network, although a staple in many machine learning tasks, was outperformed by the recurrent architectures, suggesting the importance of considering the nature of data and the inherent characteristics of the model when choosing an architecture.

Furthermore, our experiment underscores the value of model evaluation. By evaluating the models using both MAE and MSE, we gained a comprehensive view of each model's performance, ensuring our conclusions are both robust and informative.

As air quality prediction remains a critical area for both environmental and public health perspectives, the insights gained from this study are invaluable. Adopting the right deep learning model can lead to more accurate predictions, which, in turn, can guide policymakers and stakeholders in making informed decisions to improve and safeguard the environment.