Urban Air Quality Prediction Using Deep Learning

By - Atharva Talegaonkar

Project Overview

The goal of this project is to develop a deep learning model to predict urban air quality, specifically focusing on forecasting nitrogen dioxide (NO2) levels in the air. Using historical data, the project implements a Long Short-Term Memory (LSTM) neural network, a type of recurrent neural network, to forecast NO2 levels based on features such as temperature, humidity, and other environmental factors. The prediction model can help cities and governments take proactive measures to improve air quality and reduce pollution exposure.

Objective

- **Air Quality Prediction**: Predict NO2 concentration levels based on historical air quality data using a deep learning model.
- **Real-time Forecasting**: Develop a model that can predict future air quality on a daily basis.
- **Enhance Accuracy**: Utilize the power of LSTM networks for time-series prediction, optimizing performance with hyperparameter tuning.

Methodology

1. **Data Collection and Preprocessing**
    - **Dataset**: The dataset used contains historical air quality data, including features like NO2 levels, temperature, humidity, and other pollutants.
    - **Missing Data Handling**: Missing values in the dataset were handled using forward fill techniques.
    - **Normalization**: The features and target variables were scaled using MinMax scaling to ensure proper input range for the LSTM model.
2. **Feature Engineering**
    - The features included in the model were:
        - `OPEN`: Opening price of air quality-related instruments.
        - `HIGH`: High readings.
        - `LOW`: Low readings.
        - `CLOSE`: Closing prices of air quality instruments.
    - Time-series sequences were created using a sliding window approach.
3. **Model Architecture**

    - **LSTM Network**: A simple LSTM model with one layer was used to capture temporal dependencies.
    - **Optimizers**: Adam optimizer was used for training the model.
    - **Loss Function**: Mean Squared Error (MSE) was chosen as the loss function due to its effectiveness for regression tasks.
    - **Training Configuration**:
        - Number of Epochs: 10
        - Batch Size: 16
        - Hidden Layer Units: 64
4. **Model Training**
    - The training process ran for 10 epochs with a batch size of 16.
    - A simple LSTM-based architecture was used with regularization techniques such as dropout to prevent overfitting.
5. **Evaluation**
    - **Performance Metrics**: The model was evaluated using Mean Absolute Error (MAE) and Mean Squared Error (MSE) on both the training and testing sets.
    - The results show that the model was able to predict air quality fairly well with a MAE and MSE value that indicated reasonable accuracy.

Results

1. **Loss**: The training loss decreased steadily across the epochs, which indicates that the model was learning the patterns in the data.
2. **Performance on Test Data**: The model was able to predict the NO2 concentration levels reasonably well on unseen data, achieving the following metrics:
    - **Train MAE**: 0.1498
    - **Train MSE**: 0.0313
    - **Test MAE**: 0.1532
    - **Test MSE**: 0.0341
3. **Visualizations**:
    - Plots were generated to visualize the actual vs. predicted NO2 concentrations, showing that the model captured the general trend of the data but left some room for improvement in precision.

Deployment

1. **Google Colab Setup**: The model was trained and tested on Google Colab, providing an accessible environment for running and testing the model with high computational power.
2. **Flask API**: A Flask API can be used to deploy the model for real-time predictions where users can input historical data and receive NO2 forecasts for future time periods.

Challenges and Solutions

1. **Challenge**: Handling large datasets with missing or inconsistent data.
   - **Solution**: Missing data was handled with forward fill and proper data normalization techniques to ensure smooth model training.
2. **Challenge**: Model interpretability.
   - **Solution**: The LSTM model was kept simple, but further enhancement with attention mechanisms could be implemented for better insights into temporal patterns.
3. **Challenge**: Model performance.
   - **Solution**: The model performance was limited by the architecture and hyperparameter settings, but experimentation with different architectures and hyperparameters may improve the results.

Future Work

1. **Data Expansion**: Adding more features such as weather conditions, traffic data, and additional pollutants like PM2.5 could improve the model's prediction power.
2. **Advanced Architectures**: Exploring advanced architectures like bidirectional LSTMs, GRU networks, or incorporating attention mechanisms could improve accuracy.
3. **Real-Time Data Integration**: Integrating real-time data streams for continuous predictions would be beneficial for proactive air quality monitoring.
4. **Cloud Deployment**: Deploying the model on cloud platforms like AWS or Google Cloud would enable scalability and accessibility.

Conclusion

This project demonstrated the power of deep learning, specifically LSTM networks, in predicting urban air quality based on historical data. The model provided useful insights into NO_2 concentration trends and could be a valuable tool for public health officials and city planners to monitor and improve air quality. However, there is room for future improvements in model accuracy and scalability.