Deep Learning (CS365) Project Report

Title: Multi-Step Air Quality Forecasting Using Robust Deep Learning Model

Submitted by:

Bhavika Choudhary (2101AI10)

Pragya Harsh (2101AI23)

Below are the steps we followed to build this deep learning model:

1. Importing Libraries:

- The code begins by importing necessary libraries, including NumPy, Pandas, TensorFlow (Keras), scikit-learn, and Matplotlib.

2. Loading Data:

- Two CSV files, 'B_train.csv' and 'B_test.csv', are loaded into Pandas DataFrames (`df` and `df2`).
- The index column is set to 'dt', and the 'Unnamed: 0' column is parsed as datetime.

3. Data Preprocessing:

- The train and test data are converted to NumPy arrays (`train_data` and `test_data`).
- The last 24 rows of the training data are stored in the `hell` variable and saved as a JSON file named 'sample.json'.

4. Sequence Creation Functions:

- Two functions are defined for creating training and test sequences: `create_training_sequence` and `create_test_sequence`.
- These functions take a sequence length, step size, and input data to create sequences and corresponding target values.

5. Sequence Generation:

- Training sequences (`X_train_1` and `y_train_1`) are generated using the `create_training_sequence` function.
- Test sequences (`X_test_1` and `y_test_1`) are generated using the `create_test_sequence` function.

6. Model Building:

- A Sequential model (`model_1`) is created using Keras.
- It consists of an LSTM layer with 50 units, ReLU activation, a dropout layer (0.2), and a Dense layer with the output size of `time_step_1`.
- The model is compiled with the Adam optimizer and mean squared error loss.

7. Model Training:

- The model is trained using the training sequences (`X_train_1` and `y_train_1`) for 50 epochs with a batch size of 32.

8. Model Prediction:

- The trained model is used to predict values for the test sequences (`X_test_1`), and the results are stored in `y_pred_1`.

9. Evaluation Metrics:

- Mean squared error (`mse_1`) and mean absolute error (`mae_1`) are calculated by comparing the predicted values with the actual test values.

10. Plotting:

- A line graph is plotted to visualize the predicted and actual PM25 concentrations over time.

11. Model Saving and Loading:

- The trained model is saved to a file ('/content/drive/MyDrive/DL/B_model_1').
- The saved model is loaded back into the `loaded_model_1` variable.

Result:

- **For City B:**
  - Step size 1:
    - Mean Squared Error: 1249.6264273855602
    - Mean Absolute Error: 21.85729994699402
  - Step size 7:
    - Mean Squared Error: 2047.5699365525395
    - Mean Absolute Error: 28.458937031328656
  - Step size 14:
    - Mean Squared Error: 3162.3310951302738
    - Mean Absolute Error: 39.42053731564399
  - Step size 30:
    - Mean Squared Error: 4218.2972843881325
    - Mean Absolute Error: 47.03788681251863
  - Step size 60:
    - Mean Squared Error: 4813.049321020322
    - Mean Absolute Error: 53.58896590709626

- **For City G:**
  - Step size 1:
    - Mean Squared Error: 69.7389385051649
    - Mean Absolute Error: 5.718678518077771
  - Step size 7:
    - Mean Squared Error: 272.4485921334055
    - Mean Absolute Error: 11.519846253904253
  - Step size 14:
    - Mean Squared Error: 377.8469970006502
    - Mean Absolute Error: 13.005623942552209
  - Step size 30:
    - Mean Squared Error: 481.98117388324516
    - Mean Absolute Error: 14.985334630946001
  - Step size 60:
    - Mean Squared Error: 610.1883062966467
    - Mean Absolute Error: 19.036128567912712

- **For City S:**
  - Step size 1:
    - Mean Squared Error: 29.670975697666695
    - Mean Absolute Error: 3.8402858369556063
  - Step size 7:
    - Mean Squared Error: 96.66557194993541
    - Mean Absolute Error: 6.998083803335506

- Step size 14:
  - Mean Squared Error: 135.76530328147183
  - Mean Absolute Error: 8.385779420420564
- Step size 30:
  - Mean Squared Error: 173.27380701401435
  - Mean Absolute Error: 10.069234659856802
- Step size 60:
  - Mean Squared Error: 255.5384599837925
  - Mean Absolute Error: 12.565951332908698

- **For City T:**
  - Step size 1:
    - Mean Squared Error: 2886.8425736276754
    - Mean Absolute Error: 40.927781265550024
  - Step size 7:
    - Mean Squared Error: 2735.2509027590568
    - Mean Absolute Error: 39.65764083226575
  - Step size 14:
    - Mean Squared Error: 2427.023904053923
    - Mean Absolute Error: 37.725476474552366
  - Step size 30:
    - Mean Squared Error: 2544.963159156641
    - Mean Absolute Error: 39.567649830462884
  - Step size 60:
    - Mean Squared Error: 2691.0042344663725
    - Mean Absolute Error: 40.338434925736074

Report Summary:

The code implements a time series forecasting model using an LSTM neural network. It loads historical and test data, preprocesses it into sequences, builds and trains an LSTM model, evaluates its performance, and finally saves and loads the trained model. The model is trained and tested for various values of step sizes (1,7,14,30,60). The code is well-structured and follows standard practices for building and training LSTM models on time series data.