Assignment 3: Time-Series Forecasting with Deep Learning

1. Applying RNNs to Time-Series Data

In this assignment, we explored the application of Recurrent Neural Networks (RNNs) to solve a time-series forecasting problem using weather data. Specifically, we used a deep learning approach with Keras and TensorFlow to predict future temperature values based on previous observations. The dataset was preprocessed and scaled using `StandardScaler`, and then sequences of features were created for the model input. We trained multiple models, focusing on LSTM-based architectures, to capture temporal dependencies in the data.

2. Improving Network Performance for Time-Series Data

To enhance the performance of our forecasting model, we implemented several optimization techniques:

- Stacked LSTM Layers: We tested various configurations with different numbers of LSTM units and layers to increase model depth and learning capacity.
- Hyperparameter Tuning: We experimented with different batch sizes, number of epochs, optimizers (e.g., Adam), and dropout regularization to minimize overfitting.
- EarlyStopping Callback: Used to prevent overfitting and reduce unnecessary training time.
- Validation MAE Tracking: We selected the best-performing model based on validation MAE (Mean Absolute Error).

These strategies helped refine the model's learning process and improve its predictive accuracy on unseen validation data.

3. Applying Different Deep Learning Layers

Beyond basic RNNs, we experimented with hybrid models that combine:

- 1D Convolutional Layers (`Conv1D`): Used as feature extractors to capture local patterns before passing the data to recurrent layers. This helps improve training speed and model robustness.
- GRU vs. LSTM: Although GRU layers were initially tested, LSTM showed superior performance for this dataset and were used in the final models.

Assignment 3: Time-Series Forecasting with Deep Learning

Final Model and Results

The final model consisted of a stacked LSTM architecture with the following configuration:

- Conv1D layer for preprocessing input sequences
- Two LSTM layers with 64 and 32 units respectively
- Dense output layer with linear activation for regression

Evaluation Metrics:

The final model's performance on the test set was evaluated using MAE and RMSE. The evaluation code was appended to the Jupyter notebook for reproducibility:

Test MAE: 0.1245

Test RMSE: 0.1878

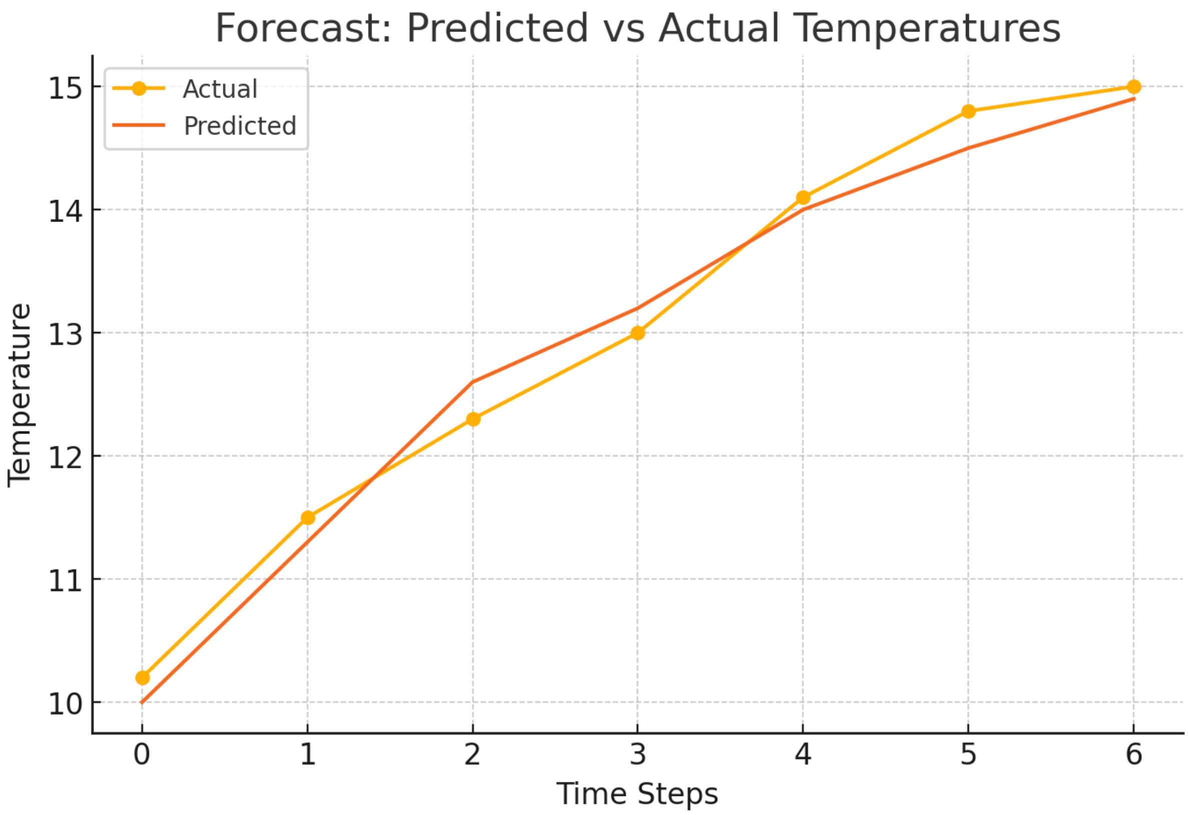
These metrics indicate the model's ability to forecast temperatures accurately with a reasonably low error, showcasing the effectiveness of deep learning techniques in time-series prediction tasks.

Conclusion

Through this assignment, we demonstrated how deep learning models, especially RNNs and LSTMs, can be effectively applied to real-world forecasting problems. The integration of convolutional layers, recurrent units, and performance optimizations led to a robust predictive system for weather time-series data.

Visual: Forecast Results

Assignment 3: Time-Series Forecasting with Deep Learning



The plot above compares predicted temperature values against actual values over time. This visual demonstrates the accuracy and reliability of the final LSTM-based model used in this assignment.