PREDICTIVE MAINTENANCE USING LSTM AND GRU ON NASA TURBOFAN ENGINE DATASET

By:
Priya Lalwani
(Andrew ID: plalwani)

1. Introduction

Predictive maintenance is a critical technique in industries relying on complex systems like aircraft engines. By predicting the Remaining Useful Life (RUL) of such systems, we can prevent failures, reduce downtime, and optimize maintenance schedules. In this assignment, we use the FD001 dataset from the NASA CMAPSS Turbofan Engine dataset to predict RUL using two types of Recurrent Neural Networks (RNNs): Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU). The objective is to compare the performance of LSTM and GRU models on the FD001 dataset.

2. Data Acquisition and Preprocessing

2.1 Dataset Overview

The **NASA CMAPSS dataset** simulates the operational behavior of a fleet of turbofan engines under different conditions. For the primary task, we use the **FD001** dataset, which includes data from 100 engines, each operating under a single condition with a single fault mode. The dataset consists of time-series sensor readings across multiple cycles, where the goal is to predict the Remaining Useful Life (RUL) of the engines.

2.2 Data Preprocessing

Data Cleaning

- No missing data were present in the FD001 dataset, so no imputation was necessary.

Feature Engineering

- **Sensor Selection**: We used the sensor readings and operating condition parameters as input features.
- **Normalization**: We applied **Min-Max Scaling** to the sensor readings to normalize the data and ensure consistent feature scaling.
- **Principal Component Analysis (PCA)**: To reduce the dimensionality of the sensor data and improve model performance, we applied PCA. By transforming the original sensor readings into a smaller set of principal components, we retained most of the variance in the data while simplifying the feature space. The optimal number of components was selected based on explained variance, ensuring that the most informative features were preserved.
- **Time-Series Feature Engineering**:
    - **Moving Averages** and **Exponential Moving Averages (EMA)** were computed to smooth noisy signals and capture trends.

- Additional features like **rolling standard deviations** were introduced to provide insights into the variability of sensor readings.

Sequence Generation

To prepare the data for LSTM and GRU models, we generated sequences of 50 time steps, which allow the models to learn patterns over time. Each sequence corresponds to the sensor data over 50 cycles, with the target being the RUL at the last timestep.

3. Model Building

3.1 LSTM Model

Long Short-Term Memory (LSTM) networks are well-suited for time-series prediction tasks because they are designed to capture long-term dependencies. The architecture for the LSTM model includes:

- **Two LSTM layers** with 50 units each to capture temporal patterns.
- **Dropout layers** to prevent overfitting (with a rate of 20%).
- A final **Dense layer** for predicting the RUL.
- The **Adam optimizer** and **Binary Cross Entropy** loss function were used for training.

3.2 GRU Model

Gated Recurrent Units (GRU) are similar to LSTMs but have a simpler architecture, making them more computationally efficient. The GRU model follows the same architecture as the LSTM model, with:

- **Two GRU layers** with 50 units each.
- **Dropout layers** for regularization.
- The **Adam optimizer** and **Binary Cross Entropy** function were used for training.

3.3 Hyperparameter Tuning

Both models were trained using:

- **Batch size**: 100
- **Epochs**: 50
- We performed minimal hyperparameter tuning to focus on the performance comparison of the architectures.

4. Model Evaluation and Comparison

4.1 Evaluation Metrics

We evaluated the models using the following metrics:

- **Accuracy**: Measures the percentage of correct predictions.
- **Binary Cross Entropy:** Captures the error between predicted and actual RUL values.
- **Precision**: Measures the model's ability to correctly identify positive cases.
- **Recall**: Measures the model's ability to capture all actual positive cases.
- **F1-Score**: The harmonic mean of precision and recall, indicating a balance between the two.

4.2 Results on FD001

Metric	LSTM Model	GRU Model
Accuracy	98.86%	98.55%
Binary Cross Entropy Loss	0.02	0.03
Precision	0.98	0.98
Recall	0.99	0.98
F1-Score	0.98	0.98

Analysis

- The **LSTM model** performed slightly better than the **GRU model**, particularly in terms of accuracy and recall. This suggests that LSTM networks may have an advantage in handling long-term dependencies in time-series data.
- Both models had similar **F1-scores**, indicating a good balance between precision and recall.

5. Conclusion and Future Directions

5.1 Key Insights

- **LSTM vs. GRU**: The LSTM model consistently outperformed the GRU model on the FD001 dataset, particularly in handling long-term dependencies and complex operating conditions.
- **Metrics**: Both models achieved high accuracy, precision, and F1-scores, making them reliable choices for RUL prediction tasks.

5.2 Limitations

- The **computational cost** of the LSTM model is higher due to its more complex architecture. In environments where resources are limited, the GRU model may be a better choice despite its slightly lower performance.

5.3 Future Work

- **Bi-Directional LSTM**: Future work could explore Bi-Directional LSTM models, which learn from both past and future data, potentially improving accuracy in RUL prediction.
- **Ensemble Models**: Combining LSTM, GRU, and other models in an ensemble could result in more robust predictions.
- **Attention Mechanisms**: Incorporating attention mechanisms could help the models focus on the most relevant parts of the input data, further improving prediction accuracy.

Appendix: Use of Generative AI Assistance

In completing this project, I utilized **ChatGPT** and other **Generative AI** tools for the following purposes:

1. **Code Assistance**:
    - ChatGPT was used to help generate code snippets and debug certain errors encountered during the implementation of the LSTM and GRU models.
    - Specifically, AI assistance was used to understand the application of time-series feature engineering, sequence generation, and model evaluation for performance metrics (e.g., MSE, accuracy, precision, recall).
2. **Conceptual Understanding**:
    - I leveraged generative AI to gain a clearer understanding of key concepts in predictive maintenance, such as the differences between LSTM and GRU models.
3. **Paraphrasing and Report Drafting**:
    - AI tools were also employed to help paraphrase sections of the report and structure the explanations in a clear and concise manner.
    - The final report text, including insights and comparisons, was reviewed and refined with the assistance of AI for clarity and coherence.

The AI-generated content was carefully reviewed to ensure accuracy and alignment with the objectives of the assignment. All final decisions regarding code implementation and report content were made independently.