CS230

Prognostic Health Management for Turbofan Engines Using Deep Learning

Aditya Gulati
adgulati@stanford.edu

Jeetsagar Ghorai
jghorai@stanford.edu

1 Introduction

Prognostic Health Management (*PHM*) is a unified framework for forecasting system health and reliability. Most systems of interest are composed of multiple components. Failure of a component in a system can result in adverse outcomes such as stoppage of operation, destruction of the system or loss of life. In most cases, the failure of a component results from the degradation of said component over the course of operation. Prognostic Health Management is concerned with forecasting potential failures of systems by monitoring the status of the components and the performance of the system.[1] PHM is an active area of research in reliability engineering and PHM techniques have been applied to a variety of systems such as hydraulic pumps, Lithium ion batteries, MOSFETs etc.

In most problems of interest, the data is available in the form of a time series of sensor readings. Given this time series data, the aim of PHM is to predict the Remaining Useful Life (*RUL*) of the system. Predicting the potential failure of a component allows the operator to plan for repair or replacement, mitigate downtime and ensure the safety of the equipment and the environment. Overestimating RUL leads to an unplanned failure, whereas underestimating RUL leads to underutilization of the component.

2 Dataset

The aim of our work is to develop a prognostics model for turbofan aircraft engines using deep learning. We have used the Turbofan Engine Degradation Simulation Data Set-2 published by the Prognostics Center of Excellence at NASA. This dataset contains run-to-failure trajectories of a number of turbofan aircraft engines.[3]

The published repository contains multiple datasets. One representative dataset, `DS02`, consists of run-to-failure simulation data for nine engines. In this dataset, the operating conditions are described using 4 attributes. The model outputs the values of 14 measured physical properties and the readings from 14 virtual sensors. Together, there are 32 features at every time-step. In the dataset, the different engines are referred to as units. The units with $u = 2, 5, 10, 16, 18, 20$ are the six training units and the units with $u = 11, 14, 15$ are the three test units. Some characteristics of the dataset are shown in Figure 1 and Figure 2.

3 Problem Statement

The dataset consists of the run-to-failure trajectories of $M + N$ units. There are s parameters which denote the operating condition of the aircraft and the output of p sensors are monitored at each time-step.

CS230: Deep Learning, Winter 2021, Stanford University, CA. (LaTeX template borrowed from NIPS 2017.)

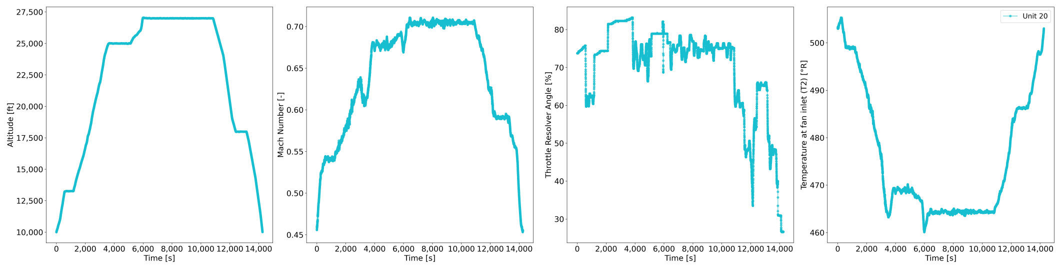

Figure 1: How the altitude, Mach number, throttle-resolver angle and temperature at fan inlet changes throughout a single flight of unit 20

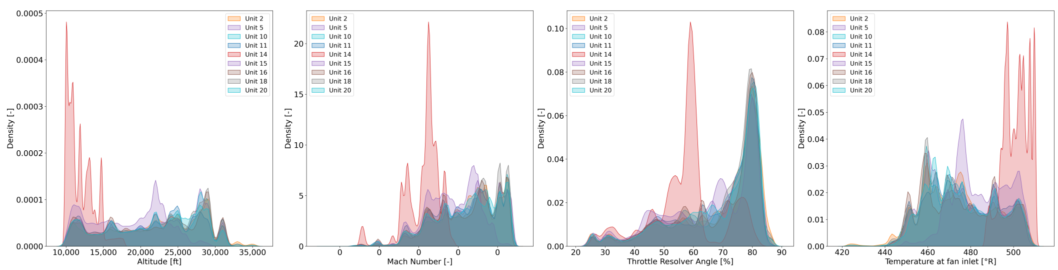

Figure 2: Kernel density estimations of altitude, Mach number, throttle resolver angle and temperature at fan inlet for 6 training and 3 test units. The flight characteristics of units 14 and 15 are different from those of the training units.

Suppose that the unit i was operated for m_i time-steps. The sensor readings at time-step t is denoted by $x_i^{(t)} \in \mathbb{R}^p$ and the operating condition at time-step t is denoted by $w_i^{(t)} \in \mathbb{R}^s$. The RUL at time-step t is denoted by $y_i^t \in \mathbb{R}$.

Suppose that $X_i = [x_i^{(1)}, \dots, x_i^{(m_i)}]^T$, $W_i = [w_i^{(1)}, \dots, w_i^{(m_i)}]^T$ and $Y_i = [y_i^1, \dots, y_i^{m_i}]^T$ Then, $\mathcal{D}_{train} = \{W_i, X_i, Y_i\}_{i=1}^N$ constitutes the training dataset and $\mathcal{D}_{test} = \{W_i, X_i, Y_i\}_{i=1}^M$ constitutes the test dataset.

The aim of our project is to use $\mathcal{D}_{train}$ to train a deep learning model $\mathcal{G}$ that predicts $\hat{Y}$, the RUL of the units in the test dataset and minimizes some loss function L.

Suppose that the test unit j is run for m'_j time-steps and $m = \sum_{j=1}^M m'_j$. Suppose that $\Delta^{(j)}$ is the difference between the true RUL and the predicted RUL at time-step j, i.e., $\Delta^{(j)} = y^{(j)} - \hat{y}^{(j)}$. We have used two functions for evaluating the models, the root-mean-square error (RMSE) and NASA's scoring functions S. These two functions are defined as follows.

$$\text{RMSE} = \sqrt{\frac{1}{m}\sum_{j=1}^{m}(\Delta^{(j)})^2}$$

$$\text{S} = \sum_{j=1}^{m} \alpha_j \exp(|\Delta^{(j)}|) \text{ where, } \alpha_j = \begin{cases} \frac{1}{13} & \text{if } \Delta^{(j)} \geq 0 \\ \frac{1}{10} & \text{if } \Delta^{(j)} < 0 \end{cases}$$

The coefficient α_j in NASA's scoring function is designed to penalize over-estimation more than under-estimation.

References

[1] Kwok L. Tsui, Nan Chen, Qiang Zhou, Yizhen Hai, Wenbin Wang, "Prognostics and Health Management: A Review on Data Driven Approaches", Mathematical Problems in Engineering, vol. 2015, Article ID 793161, 17 pages, 2015. https://doi.org/10.1155/2015/793161

[2] Biggio Luca, Kastanis Iason, ”Prognostics and Health Management of Industrial Assets: Current Progress and Road Ahead”, Frontiers in Artificial Intelligence,VOL 3, 2020, 88 pages, https://www.frontiersin.org/article/10.3389/frai.2020.578613, 10.3389/frai.2020.578613, 2624-8212

[3] M. Chao, C.Kulkarni, K. Goebel and O. Fink (2021). ”Aircraft Engine Run-to-Failure Dataset under real flight conditions”, NASA Ames Prognostics Data Repository (http://ti.arc.nasa.gov/project/prognostic-data-repository), NASA Ames Research Center, Moffett Field, CA

[4] Chao, Manuel Arias et al. “Fusing Physics-based and Deep Learning Models for Prognostics.” ArXiv abs/2003.00732 (2020)

[5] Shuochao Yao and Shaohan Hu and Yiran Zhao and Aston Zhang and Tarek Abdelzaher, ”DeepSense: A Unified Deep Learning Framework for Time-Series Mobile Sensing Data Processing”, ArXiv abs/1611.01942