Remaining Useful Life Prediction for Aircraft Engines

Ritu Thombre BT17CSE084
Sanket Gajbhiye BT17CSE067
Shubhankar Tripathi BT17CSE075
Suraj Kumar BT17CSE081

1 Abstract

Prognostics and health management is an important topic in industry for predicting state of assets to avoid downtime and failures. For example, consider run-to-failure degradation of NASA turbofan jet engines, where in characterization of fault evolution has to be studied to avoid critical failures. Each engine starts with different degrees of initial wear and manufacturing variation which is unknown to the user which evolves over multiple cycles. At each cycle, we have to specify the remaining useful cycles for each engine for which it will remain in an operational state, which is called as RUL (Remaining Useful Life). So, at the end of all cycles for an engine, we would have a RUL value which can be used to assign the health score to that engine.

2 Introduction

2.1 Problem Statement

Given a cyber physical system that contains a smart equipment which sends data about health of the system, assign the health score between 0-1 based on this data.

2.2 Prediction Goal

Goal is to predict the remaining useful life (RUL) of each engine in the test dataset. RUL is equivalent of number of flights remained for the engine after the last datapoint in the test dataset.

2.3 Dataset Description

This data set is the Kaggle version of the very well known public data set for asset degradation modeling from NASA. It includes Run-to-Failure simulated data from turbo fan jet engines.

Four different sets were simulated under different combinations of operational conditions and fault modes. Records several sensor channels to characterize fault evolution. The data set was provided by the Prognostics CoE at NASA Ames.

1. Data sets consists of multiple multivariate time series.
2. Each data set is further divided into training and test subsets.
3. Each time series is from a different engine i.e., the data can be considered to be from a fleet of engines of the same type.
4. Each engine starts with different degrees of initial wear and manufacturing variation which is unknown to the user. This wear and variation is considered normal, i.e., it is not considered a fault condition.
5. There are three operational settings that have a substantial effect on engine performance. These settings are also included in the data. The data is contaminated with sensor noise.
6. The engine is operating normally at the start of each time series, and develops a fault at some point during the series.
7. In the training set, the fault grows in magnitude until system failure. In the test set, the time series ends some time prior to system failure.
8. The objective is to predict the number of remaining operational cycles before failure in the test set, i.e., the number of operational cycles after the last cycle that the engine will continue to operate.

9. Also provided a vector of true Remaining Useful Life (RUL) values for the test data.

10. The data are provided as a zip-compressed text file with 26 columns of numbers, separated by spaces. Each row is a snapshot of data taken during a single operational cycle, each column is a different variable. The columns correspond to:

    (a) Col 1 - Machine ID

    (b) Col 2 - Current operational cycle

    (c) Col 3 - Operational setting 1

    (d) Col 4 - Operational setting 2

    (e) Col 5 - Operational setting 3

    (f) Col 6 - Sensor measurement 1

    (g) Col 7 - Sensor measurement 2 ……

    (h) Col 26 - Sensor measurement 21

3 Preprocessing

1. **Data Analysis** : To analyse the various statistics like minimum,maximum,standard deviation of training dataset.

2. **Normalization** : Done using MinMax scaler

3. **Noise removal** : Power transform was used to remove noise and also shift the distribution of dataset to Gaussian distribution.

4. **Supervised Learning** algorithms were used in prediction as the dataset was labelled. To convert this problem to Supervised Learning :

    (a) In training set,every engine with an unique Machine ID runs for a particular number of cycles and engine fails at the end of all these cycles.

    (b) So RUL for every cycle of an engine is defined as :
    (total number of cycle Engine runs for) - (current operational cycle)

    (c) This data is then used for training various models used for predicting RUL for machines in test file.

4 Prediction Using Regression Models

Models used for RUL prediction are Random Forest,Linear Regression and Logistic Regression.

1. These models were easy to implement.
2. And they also provide rough lower bound on the accuracy for models like LSTM neural network which will be implemented in the future

These models were implemented with and without k-fold cross validation.

4.1 Train Results :

Following are the training accuracies :

Type	Random Forest	Logistic Regression	Linear Regression
Without k-fold cross validation	0.7170763383768	0.016234552944027	0.674821671908
With k-fold cross validation	0.7072318910122	0.019142234068330	0.6609759771024

4.2 Test Results :

Type	Model	Mean Squared Error	Mean Absolute Error
Without k-fold cross validation	Random Forest	3164.789381316	49.260185701099
	Logistic Regression	2525.0105581658	41.55513566568
	Linear Regression	2744.267887564	44.35883696222
With k-fold cross validation	Random Forest	3179.9254265165	49.39877155461
	Logistic Regression	2455.3683190649	40.86946162852
	Linear Regression	2741.0420192114	44.358332200641

4.3 Drawbacks of Regression Models :

For time series data we're usually interested in predicting the future i.e future RUL in our case, which lies outside the scope of these regression models, which cannot learn from the previous inputs.

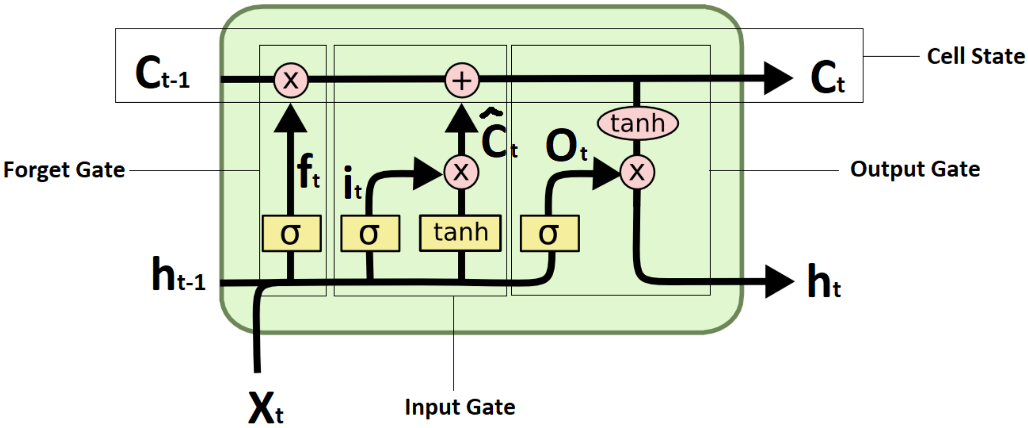
5 Prediction Using LSTM

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems.

A common LSTM unit is composed of a cell state, an input gate, an output gate and a forget gate. The cell state remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

The advantage of an LSTM cell compared to a common recurrent unit is its cell memory unit. The cell vector has the ability to encapsulate the notion of forgetting part of its previously stored memory, as well as to add part of the new information.

5.1 Working of LSTM



1. previous hidden state : h_{t-1}
2. input state : x_t
3. previous cell state : C_{t-1}
4. Forget gate :
   (a) forget state : $f_t = \sigma(x_t \mid h_{t-1})$ where $\mid$ represents concatenation.
5. Input gate :
   (a) input state : $i_t = \sigma(x_t \mid h_{t-1})$
   (b) candidate state : $\vec{C}_t = i_t$ x $tanh(x_{t-1})$
6. new cell state : $C_t = (f_t$ x $C_{t-1}) + \vec{C}_t$
7. Output gate :
   (a) output state in : $o_t = \sigma(x_t \mid h_{t-1})$
8. new hidden state : $h_t = tanh(C_{t-1})$ x o_t

5.2 Model Summary

Layer	Input shape	Output shape
LSTM	Columns x Lookback	128
LSTM	128	64
Dense	64	16
Dense	16	1

where
Number of Columns = 17
Lookback = 5,10,20
Lookback cannot be increased beyond 20 because :

1. Lookback max _cycles of any engine
2. Minimum max _cycle observed was 31
3. Increasing lookback causes overfitting and reduces accuracy of the prediction
4. Increasing lookback leads to slower computation.

5.3 Train results

Lookback	Mean Squared Error	Mean Absolute Error
5	647.08	14.07
10	468.77	8.44
20	413.23	5.77

5.4 Test reults

Lookback	Mean Squared Error	Mean Absolute Error
5	2146.21	38.07
10	2014.12	36.06
20	1751.18	33.72

6 Future Implementation

1. ARIMA, short for 'Autoregressive Integrated Moving Average' is actually a class of models that 'explains' a given time series based on its own past values, that is, its own lags and the lagged forecast errors, so that equation can be used to forecast future values.

2. Predict RUL using ARIMA model.

3. Develop a model that assigns health score to an engine based on its predicted RUL.

7 References

1. A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation", in the Proceedings of the 1st International Conference on Prognostics and Health Management (PHM08), Denver CO, Oct 2008.

2. Heimes, F.O., "Recurrent neural networks for remaining useful life estimation", in the Proceedings of the 1st International Conference on Prognostics and Health Management (PHM08), Denver CO, Oct 2008

3. Xiang Li, Qian Ding, Jian-Qiao Sun , "Remaining Useful Life Estimation in Prognostics Using Deep Convolution Neural Networks" in the journal "RELIABILITY ENGINEERING AND SYSTEM SAFETY", Apr 2018