A Neural Network-Evolutionary Computation framework for Remaining Useful Life Estimation

David Laredo

7 June, 2018

Introduction

- Maintenance of mechanical systems is, traditionally, carried out based on scheduling strategies.
  - This strategies are costly and less capable of meeting the increasing demand of efficiency and reliability.
- Intelligent Prognostics and Health Management (PMH) allow for maintenance based on the current health of the system.
- Here we define prognostics as the estimation of the remaining useful life (RUL) of a certain mechanical system.

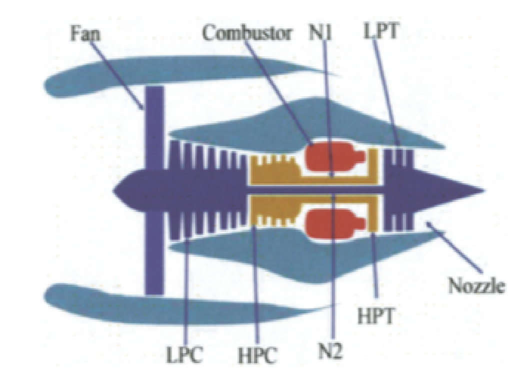
- RUL can be estimated based on history trajectory data (data driven).
  - Model-based, data-driven and hybrid.
- Here we present a framework for estimating the RUL of mechanical systems.

The framework consists of an artificial neural network (ANN) as base regressor coupled with and evolutionary algorithm for fine tuning the data-related hyperparameters.

The performance and reliability of the approach was tested using the CMAPS dataset. The results show that the proposed framework is competitive with the approaches shown in the current literature.

RUL of a commercial aircraft engine

The CMAPS simulator is a NASA turbofan engine simulator that allows the user to specify different a number of operational settings and health parameters.



RUL of a commercial aircraft engine

The CMAPS dataset is a collection time-series where each element is a cycle for an specific engine settings.

- C-MAPS has 14 inputs (fuel flow and 13 health parameters).
- It can produce up to 58 outputs but for the avilable public dataset only 21 are published.
- The output is produced as time-series where each entry is an engine cycle.
- The goal is to estimate the number of cycles the engine can run before failure.

CMAPS dataset

The CMAPS dataset is divided into 4 subsets

	C-MAPSS			
Dataset	FD001	FD002	FD003	FD004
Train Trajectories	100	260	100	248
Test Trajectories	100	259	100	248
Operating Conditions	1	6	1	6
Fault Modes	1	1	2	2

Table 1: C-MAPSS Dataset details

Performance metrics

The performance of the proposed method is evaluated using the Root Mean Squared Error (RMSE) and the RHS score defined as:

$$s = \frac{1}{N}\sum_{i=1}^{N} s_i$$
$$s_i = \begin{cases} e^{-\frac{d_i}{13}} - 1 & d_i < 0, \\ e^{\frac{d_i}{10}} - 1 & d_i \geq 0 \end{cases} \tag{1}$$

where $d_i = RUL_i^p - RUL_i$.

Data pre-processing

The following steps are applied to the data before being processed by the regressor

- Data is gropued into time-windows of size n_w and stride n_s.
- A piecewise linear degradation model is used for the RUL of the engines.
  - R_e is used as the upper limit for the piecewise linear degradation model.
- The data is standardized using min-max method to make the data be in the range (-1, 1).
- For the test set a time window is generated from the last n_w data instances.

Estimating RUL using ANN as regressor

The chosen ANN architecture complies with the following parameters

- It was chosen among 6 different architectures.
- Two objectives were pursued, that the architecture was compact and efficient.
- FD001 was used for evaluating the performance of the regressor.
- The chosen architecture must have a good compromise between the scores and its size.
- Each architecture was run 10 times and their results were averaged.

The chosen architecture is

Layer	Shape	Activation	Additional Information
Fully connected	30	ReLU	L2-$\lambda = 0.2$
Fully connected	10	ReLU	L2-$\lambda = 0.2$
Fully connected	1	Linear	

Table 2: Proposed Neural Network architecture

and its scores for dataset FD001 are

Tested Architecture	Avg. RMSE	Avg. RHS
Chosen	18.31	10.76

Table 3: Average results for chosen architecture

Choosing the optimal data-related parameters

The performance of the regressor can be improved by choosing the appropriate set of data-related parameters. Here we make use of evolutionary algorithms to pick the best possible choice of such parameters.

Let $v = (n_w, n_s, R_e)$, where $n_w \in [1, b]$, $n_s \in [1, 10]$ and $R_e \in [90, 140]$ and all the intervals are integer intervals. The value of b is dependent upon the specific subset.

The task is now to pick v such that the RMSE for each subset, and using the architecture defined in Table 2, is minimized. .

For this end Differential Evolution was used, the following rules were followed for the optimization process:

- A population of 30 elements was run for 20 generations.
- The values of the components of v were rounded to the nearest integer at each iteration.
- The Neural Network wasn't trained for the entire 100 epochs but for 20 instead.
- best1bin strategy was used for crossover and elitism.
- Maximum allowable window sizes were used for each dataset.

Since subsets FD001/FD003 and FD002/FD004 are similar in the number of fault modes the optimization process was only run for subsets FD001 and FD002 and the results obtained were applied to FD003 and FD004. Table 4 displays the obtained values for v.

Dataset	n_w	n_s	R_e
FD001	30	2	120
FD002	20	2	120
FD003	30	2	120
FD004	18	2	120

Table 4: Data-related parameters for each subset as obtained by DE.

The average results over 10 trials obtained by training the network using the parameters shown in Table 4 are displayed in Table 5.

	RMSE			RHS		
Data Subset	Min.	Max.	Avg.	Min.	Max.	Avg.
FD001	14.78	15.25	14.90	3.41	4.40	3.94
FD002	29.76	31.55	30.67	59.25	95.36	69.25
FD003	15.05	16.05	15.54	3.24	4.98	3.86
FD004	34.61	37.75	35.58	55.46	91.94	69.06

Table 5: Scores for each dataset using the data-related parameters obtained by DE.

Comparison with state-of-the-art

The proposed method is compared with some of the state-of-the-art methods proposed for this dataset. Only FD001 is used for comparison as it is the subset whose results are consistently reported in the majority of the approaches compared.

Method	RMSE
ESN-Kalman [PWW+12]	63.45
SVM Classifier [LDK13]	29.82
Time Window ANN [LGT16]	15.16
Multi-objective Networks Ensemble [ZLQT16]	15.04
Deep CNN [BZL16]	18.45
Proposed method	14.90

Table 6: Performance comparisons of the proposed method.

Conclusions

Here we presented a method for estimating the RUL of jet-engines, the highlights of the presented method are:

- A systematic method for estimating the RUL of mechanical components was presented.
- The proposed method is competitive against other state-of-the-art methods in terms performance indicators.
- The chosen ANN architecture is lightweight.
- The presented methodology is, in theory, applicable to other mechanical systems with little to no change.

References I

[BZL16] G. S. Babu, P. Zhao, and X. Li, Deep convolutional neural network based regression approach for estimation of remaining useful life, 21st International Conference on Database Systems for Advanced Applications (Springer International Publishing, ed.), 2016, pp. 214–228.

[LDK13] C. Louen, S. X. Ding, and C. Kandler, A new framework for remaining useful life estimation using support vector machine classifier, Conference on Control and Fault-Tolerant Systems, 2013, pp. 228–233.

References II

[LGT16] P. Lim, C. K. Goh, and K. C. Tan, A time window neural networks based framework for remaining useful life estimation, Proceedings International Joint Conference on Neural Networks, 2016, pp. 1746–1753.

[PWW+12] Y. Peng, H. Wang, J. Wang, D. Liu, and X. Peng, A modified echo state network based remaining useful life estimation approach, IEEE Conference on Prognostics and Health Management, 2012, pp. 1–7.

References III

[ZLQT16] C. Zhang, P. Lim, A.K. Qin, and K.C. Tan, Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics, IEEE Transactions on Neural Networks and Learning Systems 99 (2016), 1–13.