Technical Report on the solution of the CMAPSS-RUL dataset using Neural Networks

David Laredo*[1] and Jian-qiao Sun†[1]

[1]School of Mechanical Engineering, University of California, Merced

Abstract

In this report we present an a data-driven approach for estimating the Remaining Useful Life (RUL) of aero-engines. A "strided" time window is employed to generate training and test sets to be used with a conventional Multi-layer Percepton (MLP) which will serve as the main regressor for this application, no model for the engine is required . The proposed method is evaluated on the publicly available C-MAPSS dataset. The accuracy of the proposed method is compared against other state-of-the art methods available in the literature.

Index terms— Artificial Neural Networks (ANN), Moving Time Window, RUL Estimation, C-MAPSS, Prognostics

1. Introduction

Traditionally, maintenance of mechanical systems has been carried out based on scheduling strategies, nevertheless strategies such as breakdown corrective maintenance and scheduled preventive maintenance are often costly and less capable of meeting the increasing demand of efficiency and reliability [1, 2]. Condition Based Maintenance (CBM) also known as intelligent Prognostics and Health Management (PMH) allows for maintenance based on the current health of the system, thus cutting costs and increasing the reliability of the system [3]. To avoid confusion, here we define prognostics as the estimation of remaining useful component life. The Remaining Useful Life (RUL) of a system can be estimated based on history trajectory data, this approach which we refer here as data-driven can help improve maintenance schedules to avoid engineering failures and save costs [4].

The existing PMH methods can be grouped into three different categories: model-based [5] , data-driven [6, 7] and hybrid approaches [8, 9].

*dlaredorazo@ucmerced.edu

†jqsun3@ucmerced.edu

Model-based approaches attempt to incorporate physical models of the system into the estimation of the RUL. If the system degradation is modeled precisely, model-based approaches usually exhibit better performance than data-driven approaches [10], nevertheless this comes at the expense of having extensive a prior knowledge of the underlying system and having a fine-grained model of such system (which usually involve expensive computations). On the other hand data-driven approaches tend to use pattern recognition to detect changes in system states. Data-driven approaches are appropriate when the understanding of first principles of system operation is not comprehensive or when the system is sufficiently complex (i.e. jet engines, car engines, complex machinery) such that developing an accurate model is prohibitively expensive. Common disadvantages for the data-driven approaches are that they usually exhibit wider confidence intervals than model-based approaches and that a fair amount of data is required for training. Many data-driven algorithms have been proposed and good prognostics results have been achieved, among the most popular algorithms we can find Artificial Neural Networks (ANN) [11], Support Vector Machine (SVM) [12], Markov Hidden Chains (MHC) [13].

Over the past few years, data-driven approaches have gained more attention in the PMH community. A number of machine learning techniques, especially neural networks have been successfully applied to the estimate RUL of diverse mechanical systems. ANNs have demonstrated good performance when applied for modeling highly nonlinear, complex, multi-dimensional system without any prior expertise on the system's physical behavior [14]. While the confidence limits for the RUL predictions can not be naturally provided [15], the neural network approaches are promising on prognostic problems.

In this paper we propose a Multi-layer Perceptron (MLP) architecture coupled with a strided time-window approach for estimating the RUL of aero-engines. The publicly available NASA C-MAPSS dataset [16]. Raw sensor measurements with normalization are directly used as inputs to the MLP which then outputs the RUL of the jet engine in terms of cycles.

The use of Neural Networks for estimating the RUL of jet engines has been previously explored in [17] where the authors propose a Multi-layer Perceptron MLP coupled with a Feature Extraction (FE) method and a time window for the generation of the features for the MLP. In the publication the authors demonstrate that a moving window combined with a moving time-window and suitable feature extraction they can improve the RUL prediction reported by other similar methods in the literature. In [14] the authors explore an even newer ANN architecture, the so-called Convolutional Neural Networks CNNs, where they demonstrate that by using a CNN without any pooling layers coupled with a time-window as described in [17] cite the predicted RUL is further improved.

The present work takes inspiration from [17] and [14] in the sense that an ANN architecture coupled with a time-window is used to produce the RUL predictions, nevertheless this research also considers the use of a *strided* time-window which allows for more accurate predictions of the RUL of the jet-engine than the reported in [17] and [14]. Furthermore, this paper presents an optimization framework, based on the Root Mean Squared Error RMSE of the predictions, for the fine tuning of data related hyperparameters such as window-size, stride-size, etc. Such approach allows a simple MLP to obtain even better results than those reported in the current literature using less computing power.

The remainder of this paper is organized as follows:...

2. NASA C-MAPSS Dataset

The NASA C-MAPSS dataset [16] is used to evaluate the proposed method. The C-MAPSS dataset contains simulated data produced using a model based simulation program (Commercial Modular Aero-Propulsion System Simulation) developed by NASA. The dataset is further divided into 4 subsets composed of multi-variate temporal data obtained from 21 sensors.

For each of the 4 subsets a training and a test set is provided. The training sets include run-to-failure sensor records of multiple aero-engines collected under different operational conditions and fault modes as described in Table 1

The data is arranged in an $n \times 26$ matrix where n corresponds to the number of data points in each subset. The first two variables represent the engine and cycle numbers respectively. The following three variables are operational settings which correspond to the operating conditions in Table 1 and have a substantial effect on engine performance. The remaining variables represent the 21 sensor readings that model the engine degradation throughout time.

	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

Each trajectory within the train and test trajectories is assumed to represent the life-cycle of an engine. Each engine is simulated with different initial health conditions (no faults). For each trajectory of an engine the last data entry corresponds to the moment the engine is declared faulty. On the other hand the trajectories within the test sets terminate at some point prior to failure and the aim is to predict the Remaining Useful Life (RUL) of each engine in the test set. The actual RUL value of each test trajectories were also included in the dataset for verification purposes. Further discussion of the dataset and details on how the data is generated are given in [18].

3. Artificial Neural Networks

Although Artificial Neural Networks (ANNs) are widely know nowadays in this section we will briefly introduce the basic concepts behind them to provide the reader a better understanding of the tools used in this work. We will follow the notation and conventions used in [19]

Artificial Neural Networks (ANNs) are systems vaguely inspired by the biological Neural Networks in the brain. Neurons are the building blocks of any type of ANN. The Artificial Neuron AN, or neuron, implements a nonlinear mapping from $\mathbb{R}^n$ to $[a, b]$ where n is the number of inputs the AN receives and a and b depend on the chosen activation function. Usual combinations for a and b are $[0, 1]$ or $[-1, 1]$

$$f_{AN} : \mathbb{R}^n \to [a, b] \tag{1}$$

An AN receives a vector of n input signals, $\mathbf{z} = (z_1, z_2, \cdots, z_n)$, either from the environment or from other ANs. The AN computes the net input signal, and uses an activation function f_{AN} to compute the output signal, o, given the net input. The strength of the output signal is further influenced by a bias value θ. Figure 1 presents an illustration of an AN.

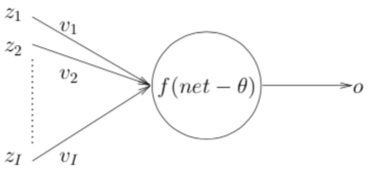

Figure 1: An Artificial Neuron

The net input signal to an AN is usually computed as the weighted sum of all input signals.

$$net = \sum_{i=1}^{n} z_i v_i \tag{2}$$

The function f_{AN} receives the net input signal and bias, and determines the output of the neuron. This function is referred as the *activation function*. Different types of activation functions can be used [19], among the most popular ones we find the sigmoid function, tanh function and the newer Rectified Linear Unit (ReLU) function. The values of the weights v_i and the bias θ are adjusted through an optimization process. In supervised learning the AN is provided with a dataset consisting of input vectors and a target (desired output) associated with each input vector. This data is referred as training set. The aim is then to adjust the weight values and bias such that the error between the real output, $o = f(net - \theta)$, of the neuron and the target output, t, is minimized.

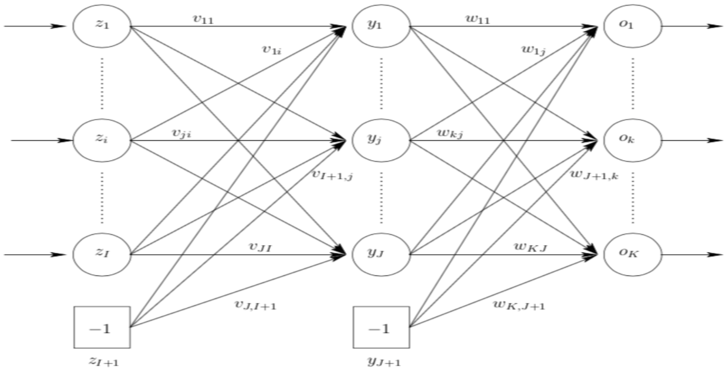

Figure 2: A Multi-layer Perceptron

By stacking several neurons together to form an output vector $\mathbf{o} = (o_1, o_2, \cdots_m)$ (a layer) and then using $\mathbf{o}$ as the input to another set of neurons and so on we can form the so-called Multi-layer Perceptron (MLP). The layers of the MLP in between the input and output layers are called hidden layers. Figure graphically depicts a MLP with one hidden layer. The process of adapting the weight matrices at each layer is called training of the Neural Network and is described in detail in [19].

4. Estimating Remaining Useful Life using Multi-Layer Perceptron as Regressor

In this section the proposed ANN-based method for prognostics is presented. Our method uses a Multi-Layer Perceptron (MLP) as the main regressor for estimating the RUL of the engines at each subset of the C-MAPSS dataset. For the training sets, the feature vectors are generated by using a strided time window while the labels vector is generated using a constant RUL for the early cycles of the simulation and then linearly decreasing the number of remaining cycles [14, 17]. For the test set, a time window is taken from the last sensor readings of the engine and used to predict the RUL of the engine.

The window size n_w, window stride n_s and constant RUL C_r hyperparameters have a considerable impact in the quality of the predictions made by the regressor [14, 17]. Hand picking the best parameters for our application is time consuming, furthermore, a grid search approach as the ones used for hyperparameter tuning in Neural Networks is computationally expensive given the search space inherent to the aforementioned parameters. In this paper we propose the use of an evolutionary algorithm, i.e. Differential Evolution (DE) [20], to fine tune the parameters. The optimization framework here proposed allows for the use of simpler architectures of Neural Networks while attaining better results in terms of the quality of the predictions made.

4.1. The Neural Network Architecture

For this study we propose to use a rather simple MLP architecture. All the implementations were used in python using the Keras/Tensorflow environment. The structure of the Network remained consisted for all the four datasets, our structure consists of two hidden layers in total; in the first hidden layer we have placed 250 neurons while in the second we have placed 50 neurons. Table 2 yields a description of the used network.

Layer	Shape	Activation	Additional Information
Fully connected	250	ReLU	Dropout(0.6)
Fully connected	250	ReLU	Dropout(0.2)
Fully connected	1	Linear	

Table 2: Proposed Neural Network architecture

References

[1] N. Z. Gebraeel, M. A. Lawley, R. Liu, and J. K. Ryan. Residual-life distributions from component degradation signals: a bayesian approach. *IEEE Transactions*, 37(6):543–557, 2005.

[2] M.A. Zaidan, A.R. Mills, and R.F. Harrison. Bayesian framework for aerospace gas turbine engine prognostics. In IEEE, editor, *Aerospace Conference*, pages 1–8, 2013.

[3] Z. Zhao, L. Bin, X. Wang, and W. Lu. Remaining useful life prediction of aircraft engine based on degradation pattern learning. *Reliability Engineering & System Safety*, 164:74–83, 2017.

[4] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel. Prognostics and health management design for rotary machinery systems - reviews, methodology and applications. *Mechanical Systems and Signal Processing*, 42(12):314–334, 2014.

[5] W. Yu and H. Kuffi. A new stress-based fatigue life model for ball bearings. *Tribology Transactions*, 44(1):11–18, 2001.

[6] J. Liu and G. Wang. A multi-state predictor with a variable input pattern for system state forecasting. *Mechanical Systems and Sygnal Processing*, 23(5):1586–1599, 2009.

[7] A. Mosallam, K. Medjaher, and N. Zerhouni. Nonparametric time series modelling for industrial prognostics and health management. *The International Journal of Advanced Manufacturing Technology*, 69(5):1685–1699, 2013.

[8] M. Pecht and Jaai. A prognostics and health management roadmap for information and electronics rich-systems. *Microelectronics Reliability*, 50(3):317–323, 2010.

[9] J. Liu, M. Wang, and Y. Yang. A data-model-fusion prognostic framework for dynamic system state forecasting. *Engineering Applications of Artificial Intelligence*, 25(4):814–823, 2012.

[10] Y. Qian, R. Yan, and R. X. Gao. A multi-time scale approach to remaining useful life prediction in rolling bearing. *Mechanical Systems and Signal Processing*, 83:549–567, 2017.

[11] N. Z. Gebraeel, M. A. Lawley, R. Li, and V. Parmeshwaran. Residual-life distributions from component degradation signals: a neural-network approach. *IEEE Transactions on Industrial Electronics*, 51(3):150–172, 2004.

[12] T. Benkedjouh, K. Medjaher, N. Zerhouni, and S. Rechak. Remaining useful life estimation based on nonlinear feature reduction and support vector regression. *Engineering Applications of Artificial Intelligence*, 26(7):1751–1760, 2013.

[13] M. Dong and D. He. A segmental hidden semi-markov model (hsmm)-based diagnostics and prognostics framework and methodology. *Mechanical Systems and Signal Processing*, 21(5):2248–2266, 2007.

[14] X. Li, Q. Ding, and J. Sun. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering and System Safety*, 172:1–11, 2018.

[15] J. Z. Sikorska, M. Hodkiewicz, and L. Ma. Prognostic modelling options for remaining useful life estimation by industry. *Mechanical Systems and Signal Processing*, 25(5):1803–1836, 2011.

[16] A. Saxena and K. Goebel. Phm08 challenge data set. [Online] Available at: https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/.

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

[18] A. Saxena, K. Goebel, D. Simon, and N. Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation. In IEEE, editor, *International Conference On Prognostics and Health Management*, pages 1–9, 2008.

[19] Douglas P. Engelbrecht. *Computational Intelligence. An Introduction*. Wiley, 2007.

[20] Rainer Storn and Kenneth Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, Dec 1997.