RUL Prediction by LSTM Model with Bayesian Parameter Optimization for Turbine Engines

Shuli Hu, Shunong Zhang, Xing Xu*

School of Reliability and Systems Engineering, Beihang University

Email: xingxu@buaa.edu.cn

Abstract. Effectively predicting the remaining useful life (RUL) of a product is significant for reasonable reliability planning and maintenance activities. The long-short term memory (LSTM) model, which belongs to deep learning methods, was applied for RUL prediction of turbine engines, and a parameter optimization method with Bayesian theory was studied.

1. Introduction

At present, the main prediction methods for remaining useful life (RUL) based on data driven for products can be divided into two categories: statistics and machine learning. Early research mainly focused on statistical methods. The advantage is that it has a solid statistical theoretical basis. The model is reliable and stable, and the algorithm is efficient. However, the prediction result is sensitive to the parameter, which is very complicated for estimating, and also, it is difficult to deal with the problem of drift. Typical mathematical statistical models for RUL prediction include ARIMA model, Markov, and Wiener process[1], etc. Since the 1990s, with the development of machine learning technology, more and more researches have focused on it to make predictions, which includes neural network(NN), support vector machine(SVM)[2], random forest(RF)[3] , Adaboost integrated model[4], etc. Since 2012, the rapid development of deep learning methods has made people pay more attention. Among them, the method represented by the recurrent neural network (RNN)can be specifically used to deal with sequence problems. Although RNN can effectively deal with nonlinear time series, there are still the following problems: 1) Due to gradient disappearance and explosion, RNN cannot handle time series data with long delay; 2) Training RNN model needs to determine the delay window length in advance, but in reality, it is difficult to automatically obtain the optimal value of this parameter. As a result, there appeared the long-short term memory (LSTM) model, which replaces the hidden layer of RNN cells with LSTM cells to obtain long-short term memory ability. It maybe can effectively predict RUL of products.

So, in this paper we tried to use LSTM model to predict RULs of some typical products. Some data of turbine engines were found and then selected from NASA public data website. (https://nasa.github.io/data-nasa-gov-frontpage/)

2. Data and Preprocessing

NASA used software to simulate large commercial turbine engines under different operating conditions. The data file contains a total of 218 engine data. We used the first 180 engine data to train, and the remaining for test. For example, the data of No. 3 engine is 26 dimensions, which are the engine number, life cycle, 3 working condition parameters and 21 operating parameters. The data are m pieces and n dimensional.

The method of principal component analysis (PCA) is used for feature extraction, and the steps are as follows:

1) Form matrix X composed of n rows and m columns;
2) Subtract the average of each row of X, that is zero-averaged;
3) Obtain the covariance matrix C, $C = 1/m\, XX^T$;
4) Find the eigenvalues and corresponding eigenvectors of the covariance matrix C;
5) Arrange the feature vectors in rows from top to bottom according to the size of the corresponding eigenvalues, take the first k rows to form a matrix P;
6) $Y = PX$ is the data after k-dimension reduction.

The decentralization step is to subtract the average value of each column:

$$\mu = \frac{\sum_{i=1}^{n} x_i}{n} \tag{1}$$

$$x_i = x_i - \mu \tag{2}$$

Where x_i is the original data value; μ is the average value of each column; n is the number of the data.

The normalized way is linear Max-Min:

$$x_i^{'} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{3}$$

Where $x_{\min}$ is the minimum value of each column; $x_{\max}$ is he maximum value of each column; x is the original data value; x_i' is the normalized value.

3. RUL Prediction Based on LSTM Model

3.1. Model Architecture

Figure 1 shows the structure of a LSTM model.

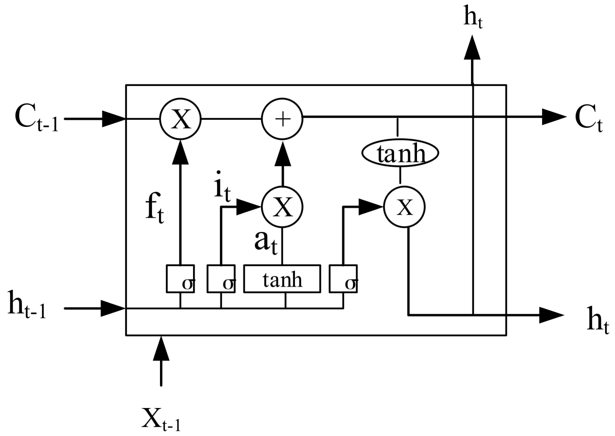


Figure 1. LSTM structure

Hidden layer h_{t-1} and feature vector x_t are input, $C(t)$ is cell state, $f(t)$ is forget gate and $i(t)$ is input gate.

The forward calculation method can be expressed as follows:

$$i(t) = \sigma(W_i h(t-1) + U_i x(t) + b_i) \tag{3}$$

$$f(t) = \sigma(W_f h(t-1) + U_f x(t) + b_f) \tag{4}$$

$$C(t) = C(t-1) \odot f(t) + i(t) \odot a(t) \tag{5}$$

$$\begin{aligned} o(t) &= \sigma(W_o h(t-1) + U_o x(t) + b_o) \\ h(t) &= o(t) \odot tanh(C(t)) \end{aligned} \tag{6}$$

Where $o(t)$ is output gate; W, U, b are corresponding weight and bias; σ is *sigmoid* activation function; $\odot$ is Hadamard product.

3.2. Predicting RUL

With the help of Keras, a deep learning framework, the network shape was set of activation function, loss function, regular terms, bias terms, constraints, and dropout[5].

A 4-layer sequential model was built, 3-layer LSTM to train, and 1-layer Dense to output. The first three layers are composed of LSTM nodes, of which the number is 48, 36, and 24. The structure has shown in figure 2 to train the data set.

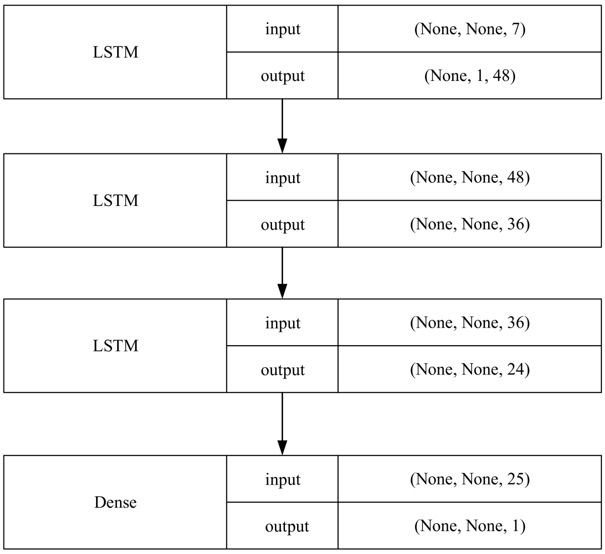


Figure 2. The network structure

Set the batch size as the total number of a single engine operating data, the learning rate is 0.001 with Adam optimizer. And train the network for epoch=100 times on the training set.

The error change during the training process is shown in Figure 3. The horizontal axis represents epoch, the vertical axis represents the errors MSE and MAE, respectively.

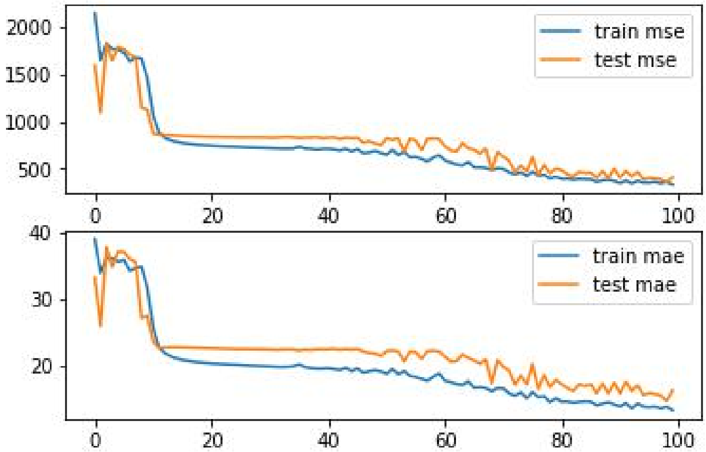
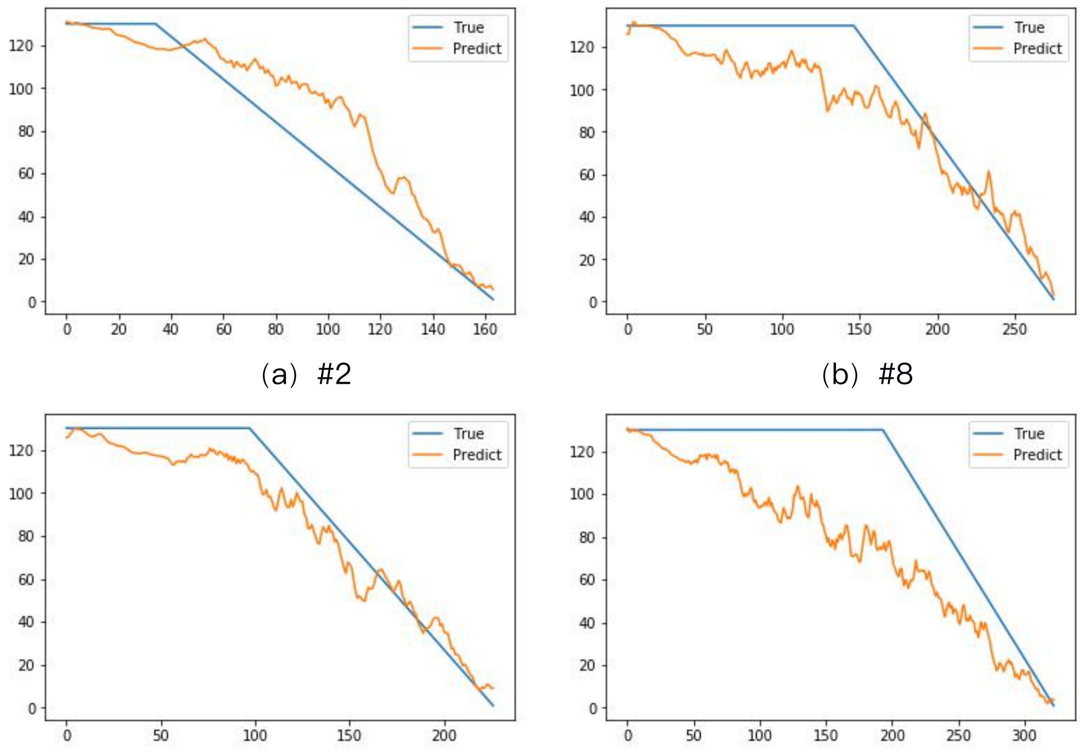


Figure 3. The changes in errors

The model predicts well on samples #2, #8, #111 and #147, which are shown in Figure 5.



(a) #2 (b) #8

(c) #111 (d) #147

Figure 4. The prediction of LSTM

The mean absolute error (MAE) on the training set is 13.34, and on the testing set is 16.30.

4. Parameter Optimization based on Bayesian Theory

There are many parameters directly affecting the performance of the model, such as the hidden parameters, learning rate, batch size, number of layers, dropout, and regularization coefficients. Setting appropriate hyperparameters is critical to the model predictive ability.

As the term suggests, the Bayesian optimization method uses Bayesian thought theory. The next hyperparameters is selected based on the existing results. Bayesian method tracks past evaluation results, and uses these results to form a probabilistic model to map these to the objective function's scoring probability:

$$P(score\,|\,hyperparameters) \tag{7}$$

4.1. Algorithm Implementation of Bayesian Optimization Method

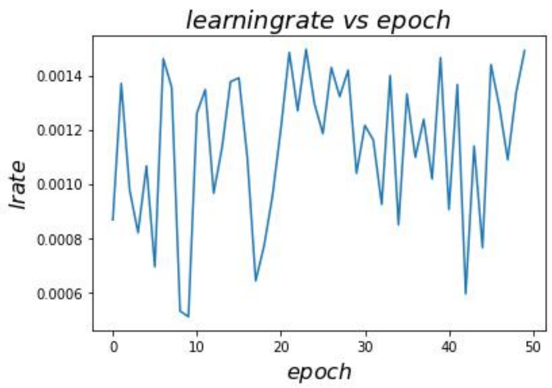
The Bayesian optimization method includes four parts: objective function, search space, optimization algorithm and visualization.

Objective function: The objective function is the LSTM in the previous section, and the return value is the MAE of the training set.

Search space: The search space is the input value we want to search. The search space is set as the learning rate, which is continuously and uniformly distributed from 0.0005 to 0.0015.
Search algorithm: Search algorithm is the core algorithm of the method. There are two options: random search and Tree of Parzen Estimators (TPE), the latter of which is selected to use.

4.2. Optimization Result

Bayesian optimization method is used to automatically adjust the learning rate of the neural network. Set epoch as 50 to minimize MAE. The learning rate of each epoch is shown in Figure 5.



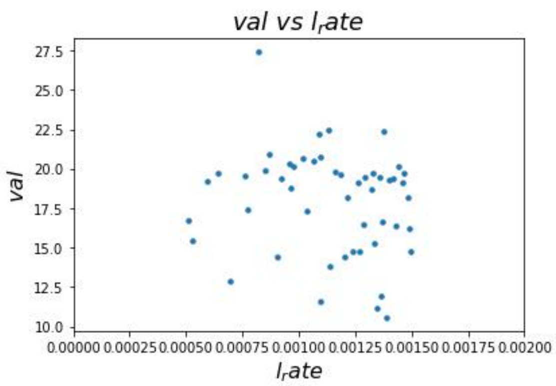


Figure 5. The learning rate for each epoch **Figure 6.** Learning rate and MAE scatter plot

With the increase of the number of epoch time, the learning rate gradually approaches the optimal solution, and the optimal solution of the learning rate is 0.001226 at the epoch is 37.
The following figure is scatter plot of learning rate and MAE.
The result shows that the MAE is the smallest when the learning rate is 0.0012260442375009789, and the minimum value is 10.55162912133929.

4.3. Comparison

Set the learning rate as 0.0012260442375009789 and retrain the model, the error change during the training process is shown in Figure 7.

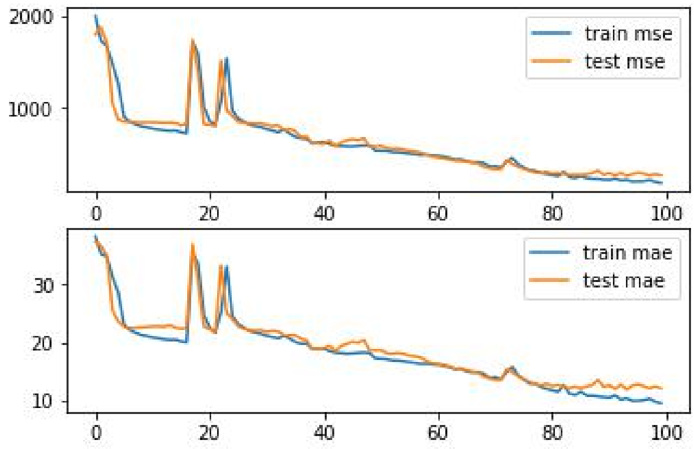
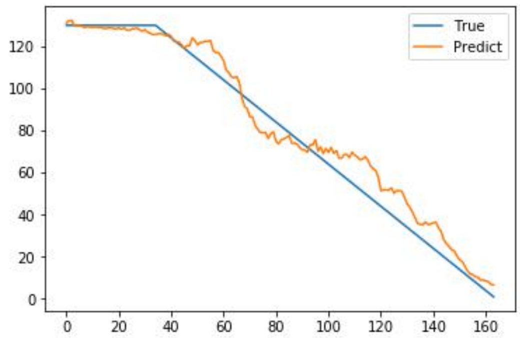
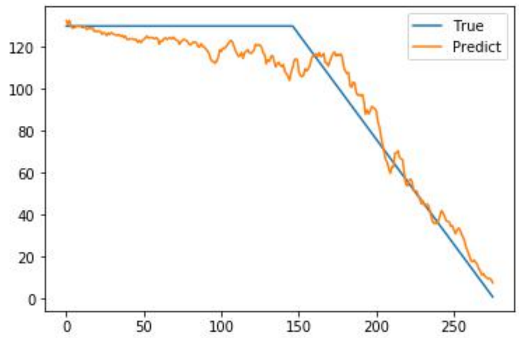


Figure 7. The error change of new model

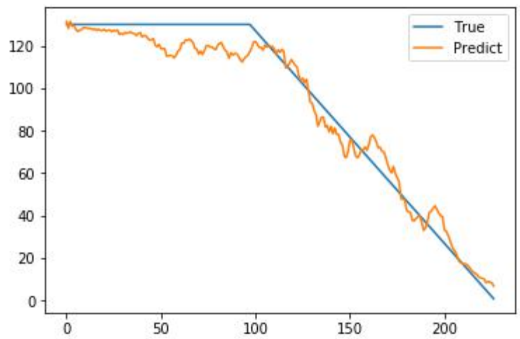
The prediction results on same samples numbered are shown in Figure 8.



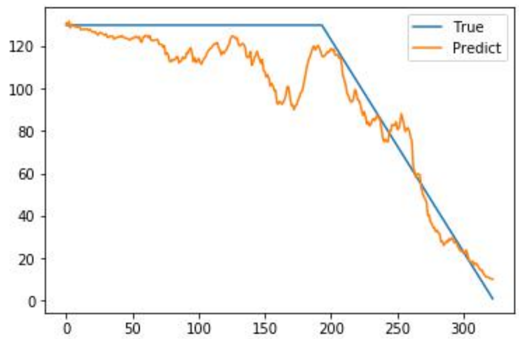
(a) #2



(b) #8



(c) #111



(d) #147

Figure 8. The prediction of the new model

The mean absolute error (MAE) on the training set is 10.55. Comparison shows that the accuracy of the optimized LSTM model is significantly improved.

Table 1. the MAE in new model and the previous

	Previous model	Optimized model

MAE on the training set	13.34	10.55

5. Conclusion and Future Work

In this paper, the PCA algorithm was used to extract data features, and a LSTM model was trained to predict the RUL of some turbine engines. Then, a Bayesian parameter optimization method was studied, and the better hyperparameters were obtained, which means more accurate RUL prediction model can be obtained. In the future, how to add an algorithm to update the model automatically may be studied. That is, to optimize hyperparameters online and adjust model dynamically.

6. References

[1] Li J K, Duan F F, et al. Reliability Evaluation and Life Prediction of Aircraft Integral Drive Generator[J]. Chinese Journal of Construction Machinery, 2018,16(05):399-403

[2] Strapp J W, Leaitch W R, Liu P S K. Hydrated and Dried Aerosol-Size-Distribution Measurements from the Particle Measuring Systems FSSP-300 Probe and the Deiced PCASP-100X Probe[J]. Journal of Atmospheric & Oceanic Technology, 1992,9(05):548-555

[3] Chennubhotla C, Jepson A. Sparse PCA Extracting Multi-scale Structure from Data[J]. Eprint Arxiv, 2001

[4] Desale R P, Verma S V. Study and analysis of PCA, DCT & DWT based image fusion techniques: Signal Processing Image Processing & Pattern Recognition (ICSIPR), 2013 International Conference on, 2013[C]

[5] Ma X L, Tao Z M and Wang, Y H. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data[J]. Transportation Research Part C, 54:187-197