Thrust Estimation for Aero-engine Based on Deep Convolution Neural Network

Binbin Wang, Jianguo Xu*, Xingyi Liu and Qiangang Zheng
Nanjing University of Aeronautics and Astronautics

* xujianzuo@nuaa.edu.cn

Abstract. Aero-engines' main task is to provide thrust for military or civil aircraft, so direct thrust control technology becomes one of the important research directions in the aero-engine control field. Thrust estimator design is an important part of direct thrust control. To overcome the shortcomings of traditional thrust estimation methods, such as low accuracy and poor robustness, a deep convolution neural network (DCNN) was introduced to design thrust estimator for aero-engine. Taking a turbofan engine component-level model as the objective, 100000 data samples were generated through simulation for training and testing the estimator. Twelve measurable parameters, such as fuel flowrate ($W_f$), combustion chamber exit temperature ($T_4$), high pressure rotor speed ($HPS_{N_{\text{mech}}}$), low pressure rotor speed ($LPS_{N_{\text{mech}}}$), nozzle exit temperature ($T_8$), etc. were selected as the input features of the estimator, and engine thrust was used as the output label. The effects of training data and test data partition ratio, convolution and pooling layer number, convolution kernel size, pooling window size and channel number of convolution/pooling layers on the performance of the thrust estimator were compared and analysed. Furthermore, the performance of the DCNN based thrust estimator was compared with that of traditional thrust estimators based on BP neural network and support vector machine (SVM). The validation results show that the DCNN based thrust estimator proposed in this paper has excellent thrust estimation performance, and its average accuracy can reach as high as 0.11%.

1. Introduction
With the increasing attention paid to the safety, economy and environmental friendliness of aircraft, the requirements for aero-engine control system are getting higher and higher. The traditional aero-engine control methods are based on the closed-loop control of the measurable parameters such as rotor speed or pressure. But for turbojet or turbofan engines, the engine thrust is the parameter that people really care about when the safety is guaranteed [1-6]. Because it is very difficult to directly measure the thrust of an engine, the usual methods are to indirectly reflect the thrust using other parameters such as rotor speed, pressure and temperature. However, the relationship between these measurable parameters and the thrust is complex, and tends to change when the engine state changes, resulting in the problem that the engine thrust cannot be accurately controlled.

Scholars have done much relevant research on thrust estimator design technology. Zhou Jun et al. proposed a EW-ELM algorithm for thrust estimation, which was designed based on extreme learning machine (ELM) and wavelet theory [7]. Yao Yanlong et al. presented an adaptive genetic neural network algorithm (AGNNA), which has good global convergence property and rapid local searching capability, and can predict aero-engine thrust in the full flight envelope [8]. Maggiore et al. designed a
thrust estimator by using the non-linear mapping ability of neural network [9]. However, the structure of shallow neural network is very simple, and the initial weight matrix is usually generated randomly, so it is prone to fall into local optimal solution. Deep convolution neural network (DCNN) has excellent performance in feature extraction and modelling and has been more and more studied and applied in academia and industry in recent years. In this paper, DCNN was studied and used to design the thrust estimator for aero-engine.

2. DCNN Model

2.1. DCNN

DCNN is constructed by imitating the biological visual perception mechanism, and can be used for supervised learning and unsupervised learning. The sharing of convolution kernel parameters and sparsity of interconnection between layers in hidden layers enable DCNN to extract features with small computation. The basic structure of DCNN is composed of an input layer, several convolution layers, several pooling layers, a full connection layer and an output layer. The convolution and pooling layers are alternately arranged and connected. The last pooling layer is reshaped as the full connection layer, and the output layer is determined according to the type of outputs [10].

There are two important concepts in DCNN convolution layer: local receptive fields and shared weights. Assuming that the input is shown in Figure 1, the red-line area in the figure is the local receptive field. The neurons in the hidden layer are only connected with 3*3 neurons in the input layer. In other words, each neuron in the hidden layer has a fixed-size visual field to observe the features of the previous layer.

![Figure 1. Local receptive field of a hidden neuron](image)

Each local receptive field has a weight matrix, convolution kernel, which is the same size as the field. If each convolution kernel is different from each other, the computation will be very large in the training process. So a common practice is that different local receptive fields share the same convolution kernel.

The pooling layer is used to reduce the dimension of the feature maps outputted from the convolution layer. Similarly, the pooling layer also has a pooling window to scan the feature map. And there are two main types of pooling: maximum pooling and average pooling.

2.2. Model Construction

As shown in Table 1, twelve typical measurable parameters of turbofan engine were selected as the input features of the thrust estimator, including fuel flowrate ($W_f$), combustion chamber exit temperature ($T_4$), high pressure rotor speed ($HPS_{N_{mech}}$), low pressure rotor speed ($LPS_{N_{mech}}$), nozzle exit temperature ($T_8$), etc. In addition, engine thrust ($F_g$) is taken as the output label.

| Symbol   | Parameter                        |
|----------|----------------------------------|
| $W_f$    | fuel flowrate                    |
| $HPC-P_r$| high pressure compressor pressure ratio |
| $LPC-P_r$| low pressure compressor pressure ratio |
DCNN usually uses square matrices as input. Wei Zhang, etc. used one-dimensional tensors as input data, and the network training efficiency was improved [11]. We used the same method to input the feature parameters. The input data first passes through the convolution layer, which scans the input data with a certain stride. The output of the convolution layer acts as the input of the next pooling layer, and the pooling layer scans the input data with the pooling window in a fixed stride. There may be several convolution layers and pooling layers. The last pooling layer is reshaped as the full connection layer, and the output layer produces engine thrust prediction. The structure of the DCNN based thrust estimator is shown in Figure 2.

![Figure 2. Structure of DCNN based thrust estimator](image)

### 3. Performance analysis

The dataset used for training thrust estimator in this paper were all generated through simulation with a component-level model of a turbofan engine [12]. The dataset has 100,000 data samples which can be chosen as either training samples or test samples.

Because the order of magnitude of each input feature is quite different, some training problems such as gradient disappearance may occur during the training process and the model accuracy will be greatly affected. So it is necessary to normalize all the input features. We adopt the min-max normalization method as shown in formula (1).

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

In the formula (1), $x_{\min}$ and $x_{\max}$ denote the minimum and the maximum of the input feature $x$, respectively, and $x^*$ denotes the normalized value of $x$. Thus the feature $x$ is re-scaled within the range [0, 1].
3.1. training data and test data partition ratio
To demonstrate and illustrate the performance of the DCNN thrust estimator model, it was compared with two other AI approaches, backpropagation neural network (BP) and support vector machine (SVM) [13]. BP neural network adopts the structure of single hidden layer with 6 neurons. SVM uses Gauss kernel function, the regularization factor C is 100, the width of Gauss kernel γ is 3, and the error coefficient is 0.01. The learning rate for the three methods is set to be 0.01, and the training iteration number is 100.

The original dataset was divided into training sets and test sets according to the ratio of 8:2, 6:4 and 5:5. And the thrust estimation accuracy for the three methods, which is represented by mean square error (MSE), is shown in Figure 3.

![Figure 3](image-url) **Figure 3.** Performance comparison for DCNN, BP and SVR models with different training data and test data partition ratio

As seen in Figure 3, the DCNN model is more accurate than the other two models, and when the ratio of training set to test set is 8:2, the performance of the three models all become best. So the training data and test data partition ratio is set at 8:2 in this paper. Figure 4 shows some of comparison results between DCNN prediction values and thrust labels.

3.2. Number of convolution/pooling layers
For DCNN, there are two different types of hidden layers: convolution layer and pooling layer. Different DCNNs can be constructed with different number of convolution layers and pooling layers. Although the increase of hidden layers enhances the ability to extract features, more hidden layers does not mean better performance for specific applications. In addition, because of the different effects of convolution layer and pooling layer, the performance of different DCNN structures is usually not same. Suppose Conv represents convolution layer, Pool represents pooling layer and Full represents full connection layer. The performance of four different network structures, Conv-Full, Conv-Pool-Full, Conv-Conv-Full and Conv-Pool-Conv-Pool-Full, were compared with the channel number set to 16. The comparison results are shown in Table 2.

| Table 2. MSE of different DCNN Structures |
|-----------------------------------------|
| **CNN Structure** | **MSE** | **Max MSE** | **Min MSE** |
|-------------------|---------|-------------|-------------|
| Conv-Full         | 1.82*10^4 | 1.5*10^{-2} | 1.79*10^{-5} |
| Conv-Pool-Full    | 8.36*10^{-6} | 6.39*10^{-4} | 1.19*10^{-6} |
| Conv-Conv-Full    | 4.42*10^{-5} | 3.31*10^{-3} | 2.03*10^{-6} |
| Conv-Pool-Conv-Pool-Full | 2.07*10^{-5} | 1.76*10^{-3} | 1.53*10^{-6} |
Table 2 shows that when the network structure is Conv-Pool-Full, the thrust estimation accuracy is the highest, and the average MSE is $8.36 \times 10^{-6}$. And as the number of hidden layers increases, the MSE increases accordingly. Although the MSE of the last network structure also tends to decrease, the second network structure, Conv-Pool-Full, was chosen to design the thrust estimator for its simplicity.

### 3.3. Channel number

Channel number is usually determined based on experience. Academia has provided many types of DCNNs, such as LeNet-5, AlexNet, VGGNet and so on. Because LeNet model has excellent performance and is widely used, we choose the channel number referring to LeNet-5 model. We constructed 4 DCNN models which channel numbers are 6, 16, 32 and 64 respectively, and the training results of these models are shown in Table 3.

| Type          | Conv6+Pool6 | Conv16+Pool16 | Conv32+Pool32 | Conv64+Pool64 |
|---------------|-------------|---------------|---------------|---------------|
| MSE           | $2.8 \times 10^{-5}$ | $8.36 \times 10^{-6}$ | $4.37 \times 10^{-5}$ | $8.8 \times 10^{-5}$ |

As seen in Table 3, the training effects are not very good if the channel numbers are too small or too large. When the channel number is 16, which is close to the number of input features, the estimation performance of DCNN is the best. Figure 5 shows the performance of the DCNN models change during the training process. As can be seen, although Conv64+Pool64 model has better performance in the initial training stage, Conv16+Pool16 model becomes best in the middle and later stages and has the best average performance.

### 3.4. Convolution kernel and pooling window size

The sizes of convolution kernel and pooling window also have influence on DCNN performance and they are usually set as 2, 3 or 5 based on experience. In this paper, the DCNN models were constructed with a convolution layer and a pooling layer. So we trained and tested the DCNN models with 1*3 and 1*5 convolution kernel, and 1*2, 1*3 and 1*5 pooling window. As can be seen in Table 4, when the convolution kernel size is 1*5 and the pooling window size is 1*5, the DCNN model has the highest accuracy. The MSE of the DCNN model is $6.32 \times 10^{-6}$ and the average accuracy is 0.11%.
| Conv Kernel Size | Pooling Window Size | MSE     |
|------------------|---------------------|---------|
| 1*3              | 1*2                 | 8.36*10^6 |
| 1*3              | 1*3                 | 5.80*10^5 |
| 1*3              | 1*5                 | 6.69*10^5 |
| 1*5              | 1*2                 | 1.54*10^4 |
| 1*5              | 1*3                 | 2.07*10^4 |
| 1*5              | 1*5                 | 6.32*10^6 |

4. Conclusions
In this paper, an aero-engine thrust estimator was designed based on DCNN. The training and test datasets for the estimator were generated through simulation with a turbofan engine component level model. The effects of training data and test data partition ratio, convolution and pooling layer number, convolution kernel and pooling window size, channel number, etc. on the performance of the thrust estimator were studied. The validation test results show that the DCNN based thrust estimator has excellent performance and is better than the traditional thrust estimators based on BP neural network and SVM.

References
[1] J. X. Mu, David Rees, G. P. Liu. Advanced controller design for aircraft gas turbine engines[J]. Control Engineering Practice, 2004, 13(8).
[2] Timothy R. Conners, Robert L. Sims. Full flight envelope direct thrust measurement on a supersonic aircraft[R]. NASA/TM-1998-206560.
[3] Rupesh Kumar, Hyunkyoung Shin. Thrust estimation of a flapping foil attached to an elastic plate using multiple regression analysis[J]. International Journal of Naval Architecture and Ocean Engineering, 2019, 11(2).
[4] Maggiore M., Ordonez R., Passino K., et al. Estimator design in jet engine applications[C]. Proceedings of the 38th Conference on Decision and Control . 1999.
[5] Q. Jing, C. Wu. Beidou satellite maneuver thrust force estimation for precise orbit determination[J]. GPS Solutions, 2018, 22(2).
[6] O. N. Korsun, B. K. Poplavsky, S. Ju. Priholdko. Intelligent Support for Aircraft Flight Test Data Processing in Problem of Engine Thrust Estimation[J]. Procedia Computer Science, 2017, 82-87.
[7] J. Zhou, T. H. Zhang. Aero-engine Thrust Estimation Based on Ensemble of Improved Wavelet Extreme Learning Machine[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2018, 35(02): 290-299.
[8] Y. L. Yao, J. G. Sun. Application of adaptive genetic neural network algorithm in design of thrust estimator[J]. Journal of Aerospace Power, 2007, 22(10): 1748-1753.
[9] Maggiore M., Ordonez R., Passino K. M., et al. Estimator design in jet engine applications[J]. Engineering Applications of Artificial Intelligence, 2007, 16(7): 579-593.
[10] J. Gan, W. Q. Wang, K. Lu. A new perspective: Recognizing online handwritten Chinese characters via 1-dimensional CNN[J]. Information Sciences, 2019, 375-390.
[11] W. Zhang, G. L. Peng, C. H. Li, Y. H. Chen, Z. J. Zhang. A New Deep Learning Model for Fault Diagnosis with Good Anti-Noise and Domain Adaptation Ability on Raw Vibration Signals [J]. Sensors (Basel, Switzerland), 2017, 17(2).
[12] J. W. Chapman, R. D. May, T. M. Lavelle, et al. Toolbox for the Modeling and Analysis of Thermodynamic Systems (T-MATS) User's Guide[J]. 2014.
[13] Y. P. Zhao. Support vector regression and their applications to parameter estimation for intelligent aero-engines [D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2009. (in Chinese)