CLC ____________ Number____________

UDC____________ Available for reference ☐Yes ☐No

Southern University of Science and Technology

Undergraduate Thesis

Thesis Title: **Airfoil Design and Optimization Based on Extreme Learning Machine**

Student Name: **XU Boqing**

Student ID: **11812910**

Department: **Computer Science and Engineering**

Program: **Computer Science and Technology**

Thesis Advisor: **Dr. Ran Cheng**

Date: April 24th, 2022

COMMITMENT OF HONESTY

1. I solemnly promise that the paper presented comes from my independent research work under my supervisor's supervision. All statistics and images are real and reliable.
2. Except for the annotated reference, the paper contents no other published work or achievement by person or group. All people making important contributions to the study of the paper have been indicated clearly in the paper.
3. I promise that I did not plagiarize other people's research achievement or forge related data in the process of designing topic and research content.
4. If there is violation of any intellectual property right, I will take legal responsibility myself.

Signature:

Date:

Airfoil Design and Optimization Based on Extreme Learning Machine

XU Boqing

(Computer Science and Engineering, Advisor: Dr. Ran Cheng)

[摘要]： 摘要内容字体为四号宋体，行间距为固定值 25 磅。摘要应简明扼要的概括出论文的主要内容，字数应为 300-500 字。

[关键词]： 关键词 1; 关键词 2; 关键词 3; （逗号隔开，关键词不少于 3 个，不多于 5 个。）

[ABSTRACT]: In the area of civil aircraft airfoil design, aerodynamic optimization technology has been widely used, but traditional design and optimization depends on complicated aerodynamic optimization technology. Thus, scientists nowadays are using a few deep learning models to help aerodynamic optimization technology's application, and this research has gained some achievements. However, Extreme Learning Machine (ELM) remains to be verify its application. This study aims at using the ELM to do the airfoil design and optimization, and in the end design a website to display the results of this study. （英文摘要字数 250-400 个实词，注意使用英文标点符号。）

[Keywords]: Airfoil Design; Extreme Learning Machine; Airfoil Optimization （关键词不少于 3 个，不多于 5 个。中英文关键词要严格对译。）

CONTENTS

1 Introduction

1.1 Aerodynamic optimization technology in airfoil design

In recent years, aerodynamic optimization technology has been widely used in civil aircraft airfoil design to improve performance and reduce aircraft cost.

The traditional civil aircraft airfoil design costs a lot. An airfoil requires an engineer's large amount of time to be designed. After that, it requires a series of simulations and experiments to verify its performance (e.g., the computational fluid dynamics simulation and wind tunnel experiment). The computational fluid dynamics costs a few days to simulate an airfoil's flow field, and the wind tunnel experiment requires a qauntity of money to build the laboratory.

1.2 The application of deep learning

However, in the application of aerodynamic optimization technology, the design variables are usually geometric parameters, and the design objective is concerned aerodynamic performance, which could be regarded as a typical black box expensive optimization problem. Therefore, more and more studies are inspired by deep learning, hoping to automatically learn the required model for airfoil design from massive data and obtain a systematic method from massive data to automatic airfoil design. Right now, modern deep learning models show stronger generalization ability and data utilization ability.

As far as the latest achievements of deep learning and airfoil design are concerned, the main method is to extract abstract features from airfoil/flow field images and build relationship models between these features and airfoil characteristics. In essence, these methods mine data features from the airfoil itself to match airfoil performance. However, without repeated use of flow field information, they may not really reflect the complex mapping from airfoil to flow field. Therefore, research on this aspect needs to be strengthened.

1.3 Extreme Learning Machine

In addition to deep learning, there are few researches on airfoil design using Extreme Learning Machine (ELM). ELM is a kind of machine learning system or method based on Feedforward neural Network, which is applicable to supervised learning and unsupervised learning.

ELM was proposed by Guang-Bin Huang, Qin-Yu Zhu and Chee-Kheong Siew of Nanyang Technological University in 2004, and presented at the IEEE International Joint Conference that year. In 2006, the original author of ELM further evaluated the algorithm and published the conclusion to Neurocomputing, which attracted attention.

There are few researches on airfoil design using Extreme Learning Machine ELM. While when compared with other shallow learning systems, ELM is thought to have a possible advantage in learning speed and generalization. Therefore, there may be some progress in airfoil design by extreme learning machine.

2 Research status at home and abroad

Current airfoil design mainly uses deep learning models.

According to the tasks solved by deep learning models, existed deep learning models can be divided into four categories: classification model, generative model, text processing model and reinforcement learning model. The research content of this project is mainly related to the classification model and generation model, and the convolutional neural network in the classification model is the main one.

The purpose is to build a data fitting model from the perspective of airfoil, experimental condition setting and experimental performance evaluation. As far as the latest achievements of deep learning and airfoil design are concerned, the main method is to extract abstract features from airfoil/flow field images and build relationship models between these features and airfoil characteristics.

2.1 Research status of airfoil design abroad

In 2017, Emre Yılmaz and Brian J. German used convolutional neural networks to train the deep learning model for airfoil performance prediction to replace the traditional computational fluid dynamics（CFD）simulation, and optimized the airfoil using the agent model Fuzhou optimization method, achieving 80% prediction accuracy on the test set [1]. The basic principle is shown in Figure 1.

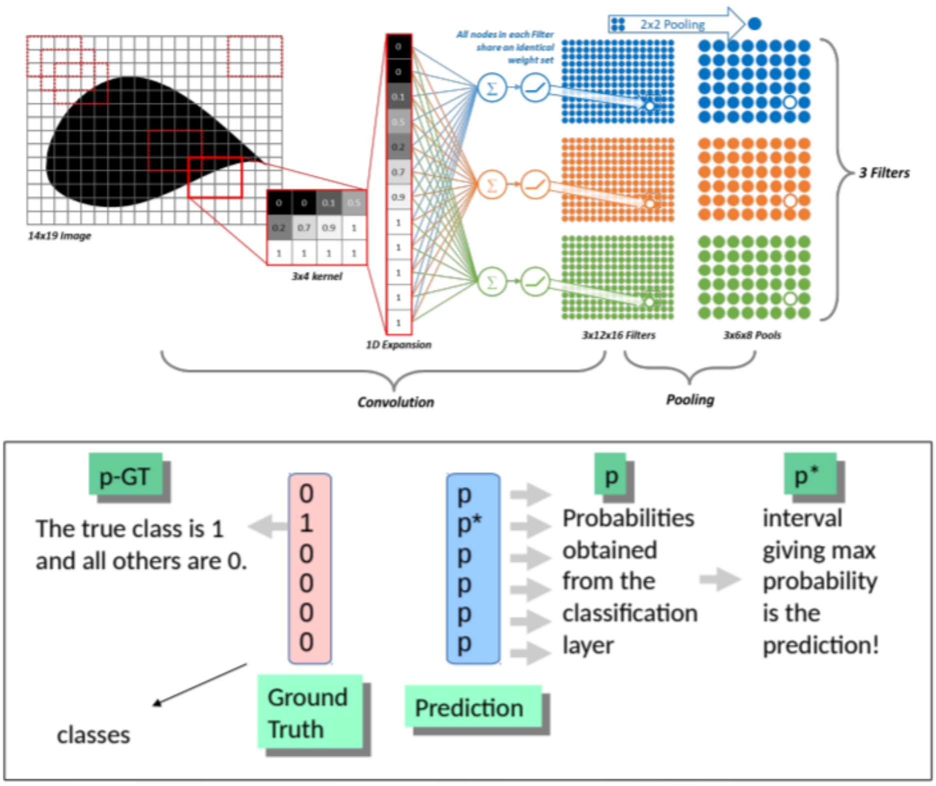

Figure 1. Schematic diagram of basic principles of airfoil performance prediction using convolutional neural networks

Subsequently, a large number of similar techniques have been used to predict wing lift coefficients [2] [3], inverse design airfoils [4], rapid pressure distribution prediction [5] and Reynolds mean Navier-Stokes simulation of airfoils flow [6]. In essence, these methods all mine data features from airfoil itself for matching airfoil performance, but without repeated use of flow field information, they may not really reflect the complex mapping from airfoil to flow field, and research in this area needs to be strengthened urgently.

For example, S. Ashwin Renganathan et al. used the method of graphic reconstruction to realize the flow field data utilization, and finally achieved better airfoil design effect than using airfoil data alone [5].

In order to solve the airfoil design and optimization problem from the perspective of generation, some researchers use generative adversarial network for airfoil design and optimization. Related work basically originated from 2019. For example, Wei Chen et al. used Bezier Generative Adversarial Networks to parameterize and optimize airfoil design[7]. The basic principle is shown in Figure 2.

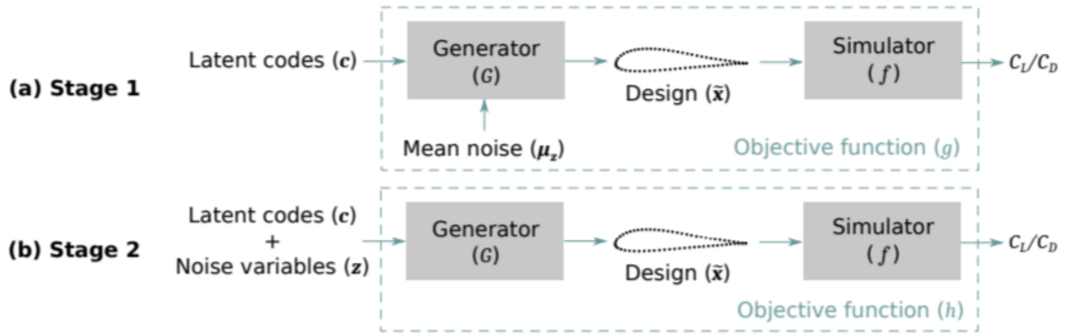

Figure 2. Parameterization and optimization method of two-stage airfoil design using GAN model

Later, Wei Chen et al. also used generative adversarial network for aerodynamic design optimization and shape exploration, and achieved good results [8]. Inspired by these ideas, Gabriel Achour et al. proposed the use of conditional generative adversance network for airfoil shape optimization [9]. By introducing the design requirements into the GAN generation process, the desired airfoil optimization effect can be obtained. Although this method makes full use of conditional information, it does not use flow field information and cannot fully reflect the mapping relationship between performance and airfoil. Therefore, the method of using conditions to generate adversarial network, airfoil shape and flow field data at the same time still needs to be explored.

Later, Wei Chen et al. also used generative adversarial network for aerodynamic design optimization and shape exploration, and achieved good results [8]. Inspired by these ideas, Gabriel Achour et al. proposed the use of conditional generative adversance network for airfoil shape optimization [9]. By introducing the design requirements into the GAN generation process, the desired airfoil optimization effect can be obtained. Although this method makes full use of conditional information, it does not use flow field information and cannot fully reflect the mapping relationship between performance and airfoil. Therefore, the method of using conditions to generate adversarial network, airfoil shape and flow field data at the same time still needs to be explored.

Emre Yı Lmaz et al. [10] studied the inverse design of airfoil through deep learning, mainly focusing on directly expressing the relationship between airfoil geometry and pressure distribution in a data-driven way, and avoiding the method of geometric parameterization. The results showed

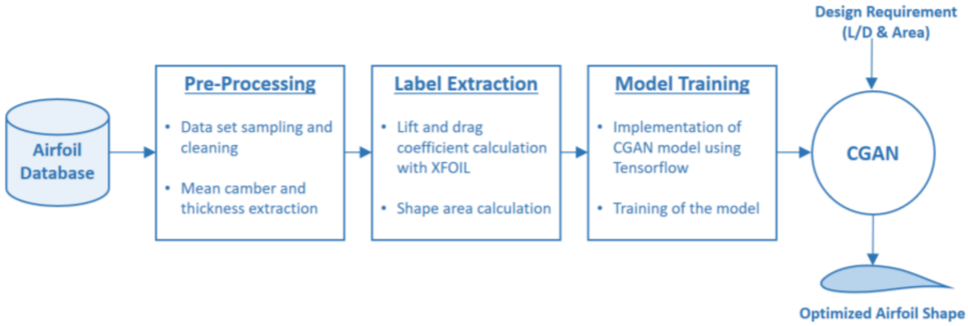

Figure 3. Principle of airfoil design method using conditional generative adversarial networks

that the prediction accuracy of this inverse design method was 70

Nils Thuerey et al. [11] predicted the $RANS$ analysis results of airfoil flow field by using the method of deep learning, and the results showed that the prediction error of velocity and pressure distribution was less than 3

Chintan S.[12] et al., taking the optimization of centrifugal pressurized airfoil as the research object, adopted the artificial neural network agent model to predict the computational results of CFD and optimize the airfoil, and compared the method with the traditional genetic algorithm. It is more advantageous to adopt deep neural network optimization method in both optimization effect and performance.

Cristina White et al. [13] proposed a method for rapid neural network prediction from constrained aerodynamic data sets, which is very suitable for highly nonlinear computational fluid dynamics problems. Compared with the current best ROM downscaling model prediction method, the amount of computation is reduced by an order of magnitude while keeping the accuracy unchanged.

Jichao Li et al. [14] used deep product generative adversarial neural network to sample and design aerodynamic agent model samples of airfoil and airfoil, so as to screen out geometric abnormal samples. Through this method, the generation efficiency of agent model can be more than doubled.

Xiaosong Du et al. [15] proposed an adversative generation network based on Bspline to parameterize the geometric shape of the generated airfoils. This method can predict that the design space maintains sufficient diversity of sample shapes and construct multiple neural network proxy models for aerodynamic characteristics, thus greatly improving optimization efficiency.

2.2 Research status of airfoil design in China

In the process of using deep learning models to develop airfoil design and optimization, a series of methods using deep learning models to assist airfoil design and optimization are proposed in China. Compared with foreign countries, deep learning models in China mainly used were classification models, which were used to predict airfoil aerodynamic coefficients, leading edge pressure distribution and so on.

Chen Hai et al. [16] from China Aerodynamics Research and Development Center proposed a method to fit airfoil aerodynamic coefficients using convolutional neural network (basic principle is shown in fig.4), which achieved far better prediction results than traditional machine learning methods and reduced root mean square error from 10^{-3} to 10^{-4} orders of magnitude. Later, Chen Bingyan et al. [17] of The China Academy of Aerospace Aerodynamics used the deep residual network to conduct relevant research on the prediction of the waverider's aerodynamic performance, which showed better prediction performance than the traditional artificial neural network. Bai Junqiang et al. [18] from Northwestern Polytechnical University also adopted the method of adding a fitter to the generated admixture network to achieve unsteady periodic flow prediction, and achieved the effect of saving more computing resources than CFD simulation. In addition, Kai Li et al. from Northwestern Polytechnical University creatively used short and long time memory neural network to model unsteady aerodynamic forces.

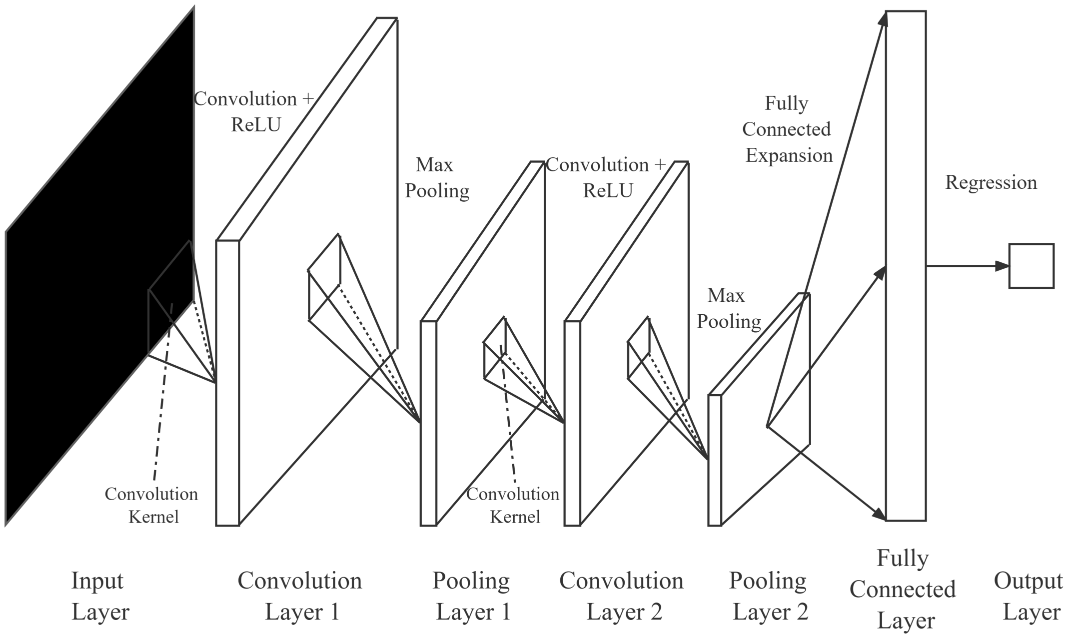

Figure 4. The specially constructed CNN used to predict the starting coefficient

In addition, Wang Yiwei et al. from the institute of mechanics, Chinese Academy of Sciences [19], adopted automatic coding machine as an order reduction method on the dimensionality of feature extraction flow field data after the encoded data is associated with the flow field characteristics, established the code flow field by regression cylindrical surface pressure coefficient of the neural network, and explored the application of data after the dimension reduction. This method provides the feasibility for the relationship between feature mining and airfoil chattering.

Chen Haixin et al. [20] [21] from Tsinghua University put forward the use of machine learning technology to simulate reasonable human behavior and mechanism of action in the process of optimization, so as to make deep use of information and knowledge to improve the practicality and efficiency of optimization. The development of machine learning technology in aerodynamic optimization is summarized, and the typical application of machine learning in optimization design is introduced based on practical work. The reinforcement learning method is applied to the aerodynamic modification strategy of supercritical airfoils, and the action strategy is learned by simulating the learning process of human and interacting with the environment. The results show that reasonable pre-training can effectively improve the efficiency of reinforcement learning and the robustness of the final strategy.

He Lei et al.[22]used machine learning methods in the application of aerodynamic characteristics model. They set the parameters of flow condition to forms vectors, which are used to map images, and these images and profile images together form the ”composited image”. They established an airfoil-aerodynamic-characteristics deep learning neural network model, based on airfoil geometry images,incoming Mach number, and the angle of attack, and obtained a not bad prediction effect, The application scope of deep learning modeling method for aerodynamic characteristics is extended. Based on the deep reinforcement learning strategy.

Wen Nuan et al. [23] studied the autonomous optimization of the shape of a class of variant aircraft. They combined deep learning and deterministic strategy gradient reinforcement learning, designed deep deterministic politic gradient ($DDPG$) learning steps to make the aircraft have higher autonomy and environmental adaptability after training and learning.

Liao Peng et al. [24] proposed a mixed airfoil frontal pressure distribution prediction method based on a deep learning model. They realized the airfoil geometry feature extraction and the parameterization of the pressure distribution curve, ultimately established the convolutional neural network model (CNN), and used the calculation results as the training sample through CFD to realize the mixed airfoil frontal pressure distribution prediction.

Compared with the deep learning model widely used in foreign countries for feature extraction and automatic parameter design, machine learning model is mainly used in China to mine laws related to specific physical quantities. For example, classification and regression tree, radial basis network, support vector machine and $Gaussian$ process models are used to model aerodynamic characteristics. Such methods do not further use a large amount of data to associate airfoils and flow fields, and the use of data is in a relatively preliminary stage. And domestic auxiliary airfoil design depth study methods are mainly concentrated in the use of classification model (i.e. regression model) to the fitting of airfoils and performance evaluation, there is no learning from generating divergent Angle to explore other unconventional airfoil design possibilities, so based on the airfoil design and optimization of the deep learning needs to be further exploration and research.

3 Extreme learning machine (ELM)

3.1 Introduction to the Extreme Learning Machine

ELM usually can be regarded as a special class of FNN or an improvement on FNN and its back propagation algorithm. Its feature is that the weight of nodes in the hidden layer is given randomly or artificially and does not need to be updated. Only the output weight is calculated in the learning process[25].

Traditional ELM has a single implicit layer. When compared with other shallow learning systems, such as single Layer Perceptron and Support Vector Machine (SVM), ELM is thought to have a possible advantage in learning speed and generalization[26].

In simple terms, the network structure of ELM is similar with single hidden layer feedforward neural network ($SLFN$), just in the stage of training to test ELM no longer used the traditional neural network based on gradient algorithm (propagation), while use the random weights and deviations, and the weights of input layer and the output layer weights are calculated by generalized inverse matrix theory[27]. After the weights and deviations on all network nodes are obtained, the training of ELM is completed. Then, when the test data comes, the network output can be calculated using the weights of the output layer just obtained to complete the prediction of the data.

3.2 Principle of the Extreme Learning Machine

Let's assume that we have a training set

$$\left\{\mathrm{x}_i, \mathrm{t}_i | \mathrm{x}_i \in \mathrm{R}^D, \mathrm{t}_i \in \mathrm{R}^m, i = 1, 2, \ldots, N\right\} \tag{3.1}$$

In symbolic expression, x_i represents the i th data example, t_i represents the corresponding mark of the ith data example, and the set generation refers to all training data[25], the node number of hidden layer of extreme learning machine is L. The network structure of extreme learning machine is the same as that of single hidden layer feedforward neural network, as shown in the figure 5.

For a neural network, we can completely regard it as a "function", only from the input and output looks much simpler. It is clear from left to right in the figure above that the input to the neural network is the training sample set x. There is a hidden layer in the middle, which is fully connected from the input layer to the hidden layer. Write the output of the hidden layer as $H(\mathrm{x})$, then the calculation formula of hidden layer output $H(\mathrm{x})$ is as follows

$$H(\mathrm{x}) = [h_1(\mathrm{x}), \ldots, h_L(\mathrm{x})] \tag{3.2}$$

The output of the hidden layer is the input multiplied by the corresponding weight plus the deviation, and the sum of the results of all nodes of a nonlinear function. $H(\mathrm{x}) = [h_1(\mathrm{x}), \ldots, h_L(\mathrm{x})]$ Is the ELM nonlinear mapping (hidden layer output matrix).

$h_i(\mathrm{x})$ is the output of the ith hidden layer node. The output functions of hidden layer nodes are not unique, and different output functions can be used for different hidden layer neurons. Usually, in practice, $h_i(\mathrm{x})$ is denoted as follows

$$h_i(\mathrm{x}) = g(\mathrm{w}_i, \mathrm{b}_i, \mathrm{x}) = g(\mathrm{w}_i\mathrm{x} + \mathrm{b}_i), \mathrm{w}_i \in R^D, \mathrm{b}_i \in R \tag{3.3}$$

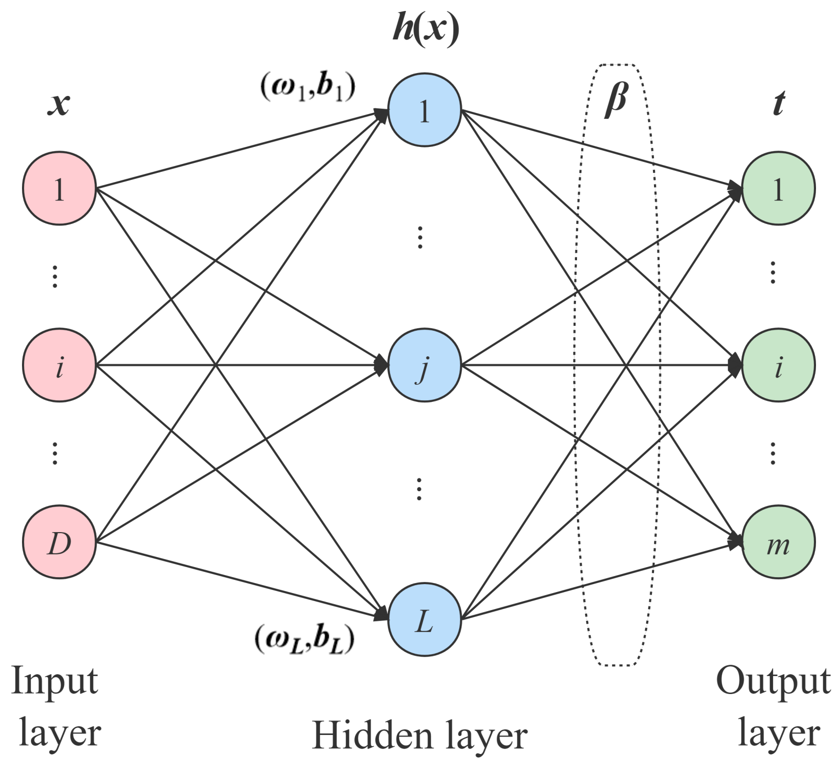

Figure 5. ELM network structure

$g(\mathrm{w}_i, \mathrm{b}_i, \mathrm{x})$($\mathrm{w}_i$ and b_i are the parameters of the activation function) is the activation function, which is a nonlinear piecewise continuous function satisfying the ELM general approximation capability theorem. The $Sigmoid$ Function and $Gaussian$ Function are commonly used. For example, we can use the $Sigmoid$ Function, then the $g(\mathrm{w}_i, \mathrm{b}_i, \mathrm{x})$ is

$$g(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} \tag{3.4}$$

(substitute the $\mathrm{w}_i\mathrm{x} + b_i$ into x). The graph of this $Sigmoid$ Function is shown in Figure 6.

After passing through the hidden layer, it enters the output layer. According to the above diagram and formula, the output of the single hidden layer for "generalized" feedforward neural network ELM is

$$f_L(\mathrm{x}) = \sum_{i=1}^{L} {}_i h_i(\mathrm{x}) = H(\mathrm{x})\boldsymbol{\beta}$$

$\boldsymbol{\beta} = [{}_1, \ldots, {}_L]^T$ is the weights between the hidden layer (with L nodes) and the input layer (with m nodes, $m \geq 1$). So far, the operation of ELM neural network from input to output is the calculation process of the above formula.

It should be noted that the unknown quantities in the above formula so far are w, b, $\boldsymbol{\beta}$ which are the weights, deviation and output weights on the nodes of the hidden layer respectively. We know that the process of neural network learning (or training) is to adjust the weights and biases between neurons according to the training data, but in fact what is learned is contained in the connection weights and biases. Then we will use ELM mechanism to solve these three values (ELM training process).

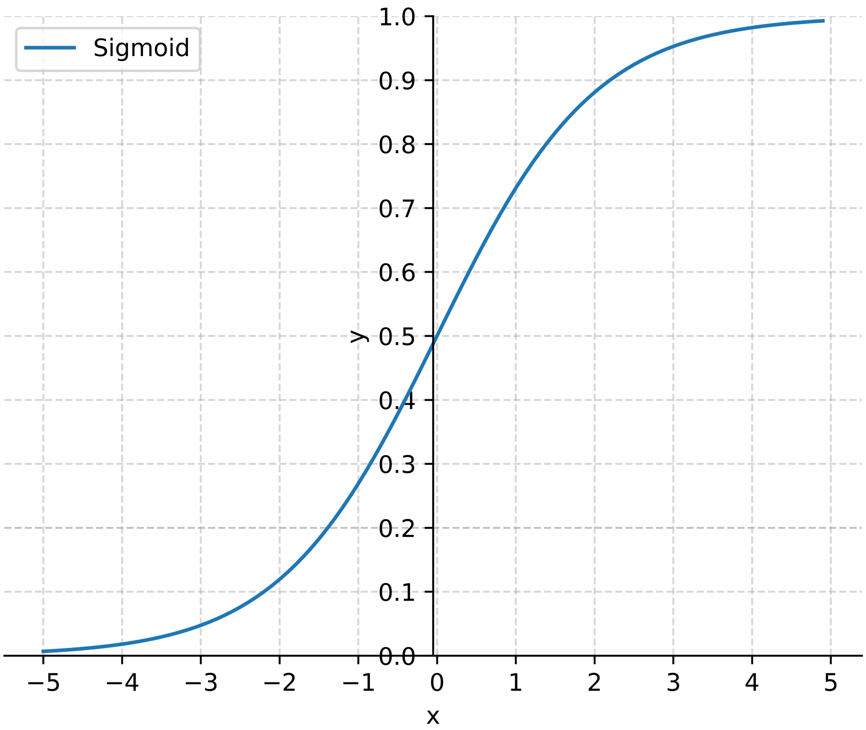

Figure 6. Image of $Sigmoid$ Function

Basically, ELM training $SLFN$ is divided into two main stages: (1) random feature mapping. (2) linear parameter solving[28].

In the first stage, the hidden layer parameters are initialized randomly, and then some nonlinear mappings are used as activation functions to map the input data to a new feature space (called ELM feature space). To put it simply, the weights and deviations on ELM hidden layer nodes are randomly generated. The random feature mapping stage is different from many existing learning algorithms (for example, SVM uses kernel function for feature mapping, deep neural network uses restricted Boltzmann machine (RBM), feature learning uses autoencoder/autodecoder). The nonlinear mapping function in ELM can be any nonlinear piecewise continuous function. In ELM, hidden layer node parameters (w and b) are randomly generated according to arbitrary continuous probability distribution (unrelated to training data) rather than determined by training, which results in a great advantage in efficiency compared with traditional BP neural network[29].

After the first stage w, b has been determined, thus the hidden layer output H can be calculated according to formulas. In the second stage of ELM learning, we only need to solve the weight of the output layer ($\boldsymbol{\beta}$). In order to obtain $\boldsymbol{\beta}$ with good effect on the training sample set, it is necessary to ensure the minimum training error, we can use $H\boldsymbol{\beta}$ and sample label T to minimize the square deviation as the evaluation of training error (objective function). The solution that minimizes the objective function is the optimal solution. That is, the weight $\boldsymbol{\beta}$ connecting the hidden layer and the output layer is solved by minimizing the approximate square deviation. The objective function is as follows:

$$min||H\boldsymbol{\beta} - T||^2, \beta \in R^{L\times m} \tag{3.5}$$

Where H is the output matrix of the hidden layer, T is the target matrix of training data:

$$H = [h(x_1), ..., h(x_N)]^T = \begin{bmatrix} h_1(x_1)...h_1(x_N) \\ ... \\ h_L(x_1)...h_L(x_N) \end{bmatrix}, T = \begin{bmatrix} t_1^T \\ ... \\ t_N^T \end{bmatrix} \tag{3.6}$$

Based on the knowledge of line algebra and matrix theory, the optimal solution of formula can be deduced as

$$\boldsymbol{\beta} = H^{\dagger}T \tag{3.7}$$

Where $H^{\dagger}$ is the $Moore–Penrose$ generalized inverse matrix of the H.

At this time, the problem is transformed into a $Moore - Penrose$ generalized inverse matrix for computing matrix H. The main methods of this problem are orthogonal projection method, orthogonalization method, iterative method and singular value decomposition method (SVD). When H^TH is a non-singular (reversible) matrix, the orthogonal projection method can be used, and the calculation result can be as follows:

$$H^{\dagger} = (H^TH)^{-1}H^T \tag{3.8}$$

And sometimes H^TH is the singular (irreversible), so the orthogonal projection method does not work well for all cases. The orthogonalization and iteration methods have limitations due to the use of search and iteration. The SVD can always be used to compute the $Moore - Penrose$ generalized inverse matrix of H, and is therefore used in most implementations of ELM.

4 Results

4.1 Implementation of Extreme Learning Machine

4.1.1 Algorithm of ELM

In summary, the ELM algorithm is summarized as follows:

Algorithm 1 $ELM\ Algorithm$

Input:
dataset: $\{\mathrm{x}_i, \mathrm{t}_i | \mathrm{x}_i \in \mathrm{R}^D, \mathrm{t}_i \in \mathrm{R}^m, i = 1, 2, \ldots, N\}$
Number of hidden layer neurons: L
Activation function: $g(x)$

Output:
Weights: $\boldsymbol{\beta}$

Steps:
1.Randomly generated input weights w and hidden layer b
2.Calculate the output of hidden layers H
3.Calculate the weight of the output layer β

4.1.2 Implementation of ELM

Implementation language: $Python$, and the version is 3.7. Mainly used project Interpreters are $Sklearn$, $Matplotlib$ and $Numpy$.

4.2 Airfoil generation through ELM

Since the ELM has been implented, we can input airfoil data to the ELM model to generate the output.

4.2.1 Input

Here is an example of one of the airfoil data shown in Figure 7.

As shown in this figure, the data of one airfoil from the training set is a mount of coordinate points. During the training period, the whole input is a large quantity of this points. Every point in each airfoil data will be converted into vector (through the system of polar coordinates).

4.2.2 Output

Through inputting a large amount of airfoil data to train this model, we can get the results now, which is shown in Figure 8.

In this figure, we can see there are nine airfoils, each of which is generated by the ELM model after trainning. In fact, the original output are the vectors, and this figure is drawned by another program according to these vectors.

0.012500	0.019300	0.012500	-0.005000
0.025000	0.031700	0.025000	-0.004200
0.050000	0.051300	0.050000	-0.001000
0.075000	0.066400	0.075000	0.002800
0.100000	0.078000	0.100000	0.006800
0.150000	0.093400	0.150000	0.014500
0.200000	0.101300	0.200000	0.021700
0.250000	0.104400	0.250000	0.028200
0.300000	0.104800	0.300000	0.033300
0.400000	0.100200	0.400000	0.038500
0.500000	0.090500	0.500000	0.038600
0.600000	0.077100	0.600000	0.035000
0.700000	0.061000	0.700000	0.028600
0.800000	0.042800	0.800000	0.020200
0.900000	0.022900	0.900000	0.010000
0.950000	0.012400	0.950000	0.004400
1.000000	0.001600	1.000000	-0.001600

Figure 7. An example of input data

4.2.3 Analysis

The output

4.3 The website of this research

4.3.1 Fundamental Introduction

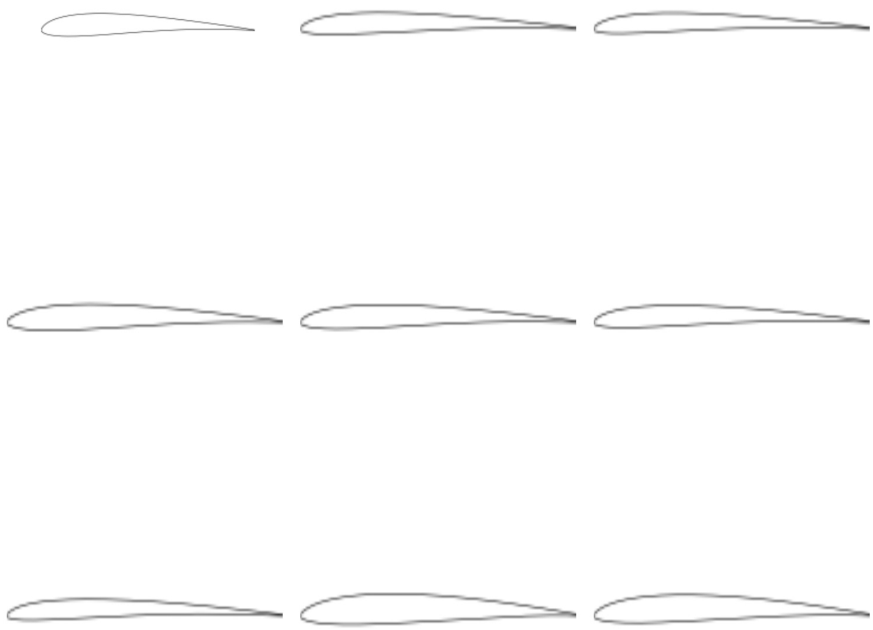

Figure 8. An example of input data

5 Conclusion

正文内容格式：中文为宋体，英文为 Times New Roman，均为小四号字，段落首行缩进 2 字符，行距 1.5 倍，下同。）

6 References

[1] Yilmaz E, German B. A convolutional neural network approach to training predictors for airfoil performance[C]//18th AIAA/ISSMO multidisciplinary analysis and optimization conference. 2017: 3660.

[2] Zhang Y, Sung W J, Mavris D N. Application of convolutional neural network to predict airfoil lift coefficient[C]//2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference. 2018: 1903.

[3] Yuan Z, Wang Y, Qiu Y, et al. Aerodynamic Coefficient Prediction of Airfoils with Convolutional Neural Network[C]//Asia-Pacific International Symposium on Aerospace Technology. Springer, Singapore, 2018: 34-46.

[4] Renganathan S A, Maulik R, Ahuja J. Enhanced data efficiency using deep neural networks and Gaussian processes for aerodynamic design optimization[J]. arXiv preprint arXiv:2008.06731, 2020.

[5] Sekar V, Zhang M, Shu C, et al. Inverse design of airfoil using a deep convolutional neural network[J]. AIAA Journal, 2019, 57(3): 993-1003.

[6] Hui X, Bai J, Wang H, et al. Fast pressure distribution prediction of airfoils using deep learning[J]. Aerospace Science and Technology, 2020: 105949.

[7] Thuerey N, Weißenow K, Prantl L, et al. Deep learning methods for Reynolds-averaged Navier–Stokes simulations of airfoil flows[J]. AIAA Journal, 2020, 58(1): 25-36.

[8] Chen W, Chiu K, Fuge M. Aerodynamic design optimization and shape exploration using generative adversarial networks[C]//AIAA Scitech 2019 Forum. 2019: 2351.

[9] Achour G, Sung W J, Pinon-Fischer O J, et al. Development of a Conditional Generative Adversarial Network for Airfoil Shape Optimization[C]//AIAA Scitech 2020 Forum. 2020: 2261.

[10] Emre Yılmaz and Brian J. German. A Deep Learning Approach to an Airfoil Inverse Design Problem[C]. AIAA Aviation Forum. June 25-29, 2018,Atlanta, Georgia.

[11] Nils Thuerey, Konstantin Weißenow, Lukas Prantl, and Xiangyu Hu[J]. AIAA Journal, Sep,2019.

[12] Chintan S. Patel，Chad Iverson.Parametric Design Optimization in Computational Aerodynamics using Artificial Deep Neural Networks[C]. AIAA Propulsion and Energy Forum. August 24-28, 2020, VIRTUAL EVENT

[13] Cristina White, Daniela Ushizima, Charbel Farhat‡. Fast Neural Network Predictions from Constrained Aerodynamics Datasets[C]. AIAA SciTech Forum. 6-10 January 2020, Orlando, FL.

[14] Jichao Li, Mengqi Zhang, Joaquim R. R. A. Martins.Efficient Aerodynamic Shape Optimization with Deep-Learning-Based Geometric Filtering[J].AIAA Journal, 10 June 2020.

[15] Xiaosong Du, Ping He, and Joaquim R. R. A. Martins.A B-Spline-based Generative Adversarial Network Model for Fast Interactive Airfoil Aerodynamic Optimization[C]. AIAA SciTech Forum. 6-10 January 2020, Orlando, FL.

[16] 陈海, 钱炜祺, 何磊. 基于深度学习的翼型气动系数预测 [J]. 空气动力学学报, 2018, 36(2): 294-299.

[17] 陈冰雁, 刘传振, 白鹏, 等. 使用深度残差网络的乘波体气动性能预测 [J]. 空气动力学学报, 2019, 37(3): 505-509.

[18] 惠心雨, 袁泽龙, 白俊强, 等. 基于深度学习的非定常周期性流动预测方法 [J]. 空气动力学学报, 2019, 37(3): 462-469.

[19] 叶舒然, 张珍, 宋旭东, 等. 自动编码器在流场降阶中的应用 [J]. 空气动力学学报, 2019, 37(3): 498-504.

[20] 李润泽，张宇飞，陈海昕. 针对超临界翼型气动修型策略的强化学习研究 [J]. 航空学报.2020,1.

[21] 陈海昕，邓凯文，李润泽. 机器学习技术在气动优化中的应用 [J]. 航空学报. 2019，1. Vol.40, No.1。

[22] 何磊，钱炜祺，汪清等. 机器学习方法在气动特性建模中的应用 [J]. 空气动力学学报.Vol.37, No.3,2019,06.

[23] 温暖, 刘正华. 深度强化学习在变体飞行器自主外形优化中的应用 [J]. 宇航学报, 2017, 11. Vol38, No.11.

[24] 廖鹏，姚磊江，白国栋. 基于深度学习的混合翼型前缘压力分布预测 [J]. 航空动力学报，2019,8,Vol34.No.8.

[25] Huang, G. B., Zhu, Q. Y., Siew, C. K. Extreme learning machine: theory and applications. Neurocomputing, 2006, 70, 489-501.

[26] Huang, G. B., Chen, L., Siew, C. K. Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans. Neural Networks, 2006, 17(4), 879-892.

[27] Huang, G. B., Chen, L. Convex incremental extreme learning machine. Neurocomputing,2007, 70(16-18), 3056-3062.

[28] Huang, G. B., Chen, L. Enhanced random search based incremental extreme learning machine. Neurocomputing, 2008, 71(16-18), 3460-3468.

[29] Huang, G. B., Ding, X., Zhou, H. Optimization method based extreme learning machine for classification. Neurocomputing, 2010, 74(1-3), 155-163.

7 Appendix

7.1 Symbol descriptions

Symbol	Description
x_i	the i th data example
t_i	the corresponding mark of the ith data example
m	the node number of input layer
L	the node number of hidden layer
$H(\mathrm{x})$	the calculation formula of hidden layer
$h_i(\mathrm{x})$	the output of the ith hidden layer node
w_i, b_i	the parameters of the activation function, $\mathrm{w}_i \in R^D$, $\mathrm{b}_i \in R$
$g(\mathrm{w}_i, \mathrm{b}_i, \mathrm{x})$	the activation function, $g(\mathrm{w}_i, \mathrm{b}_i, \mathrm{x}) = g(\mathrm{w}_i\mathrm{x} + \mathrm{b}_i)$
$f_L(\mathrm{x})$	the output of the single hidden layer
$\boldsymbol{\beta}$	the weights between the hidden layer and the input layer
H	the output of hidden layer
T	the target matrix of training data

Table 1. Symbol Descriptions

7.2 Abbreviation descriptions

Abbreviation	Description
ELM	Extreme Learning Machine
CFD	computational fluid dynamics
$RANS$	Reynolds-averaged Navier-Stokes equations
ROM	read only memory
CNN	convolutional neural network
$DDPG$	deep deterministic politic gradient
FNN	Fuzzy Neural Network
SVM	Support Vector Machine
$SLFN$	single hidden layer feedforward neural network
SVD	singular value decomposition method

Table 2. Abbreviation descriptions

7.3 The main code of the ELM

Implementation language: $Python$, and the version is 3.7. Mainly used project Interpreters are $Sklearn$, $Matplotlib$ and $Numpy$.

```
import numpy as np
from sklearn.preprocessing import OneHotEncoder,LabelEncoder
import numpy as np
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn import metrics
from sklearn.model_selection import cross_validate
from sklearn import metrics
```

```
class HiddenLayer:
    def __init__(self,x,num):
        row = x.shape[0]
        columns = x.shape[1]
        rnd = np.random.RandomState(4444)
        self.w = rnd.uniform(-1,1,(columns,num))
        self.b = np.zeros([row,num],dtype=float)
        for i in range(num):
            rand_b = rnd.uniform(-0.4,0.4)
            for j in range(row):
                self.b[j,i] = rand_b
        h = self.sigmoid(np.dot(x,self.w)+self.b)
        self.H_ = np.linalg.pinv(h)

    def sigmoid(self,x):
        return 1.0 / (1 + np.exp(-x))

    def regressor_train(self,T):
        T = T.reshape(-1,1)
        self.beta = np.dot(self.H_,T)
        return self.beta

    def classifisor_train(self,T):
          en_one = OneHotEncoder()
        T = en_one.fit_transform(T.reshape(-1,1)).toarray()
        print(self.H_.shape)
        print(T.shape)
        self.beta = np.dot(self.H_,T)
        print(self.beta.shape)
        return self.beta

    def regressor_test(self,test_x):
        b_row = test_x.shape[0]
        h = self.sigmoid(np.dot(test_x,self.w)+self.b[:b_row,:])
        result = np.dot(h,self.beta)
        return result

    def classifisor_test(self,test_x):
        b_row = test_x.shape[0]
        h = self.sigmoid(np.dot(test_x,self.w)+self.b[:b_row,:])
        result = np.dot(h,self.beta)
        result = [item.tolist().index(max(item.tolist())) for item in result]
        return result
```

8 Acknowledgements

Three years ago, with the strongest confidende, I chose the Computer Science and Technology without hesitation. However, my imagination is different from the reality. The course is difficult for me, and what a pity that until I have entered this program did I realize that I am not gifted in CS. It is a heavy setback for me. Fortunately, I am not alone, and I do not want to give up.

I am gateful for my advisor, Dr. Ran Cheng. I know I am almost his most shameful student, but he, just the opposite, disagree. He once told me that everyone has his or her own way, there is no necessary to judge every one in the same standard, and he thought I am also a successful man, though not in science research. I hope in the future I can do everything better, to confirm that he is right: I am not a loser, and I will not be, either.

I want to thank my another teacher, Cheng He. In the past time, he was Dr. Ran Cheng's RAP and helped me a lot. He was strict, but I know he did this for us. I remember he corrected all the wrongly written words for me, and emphasized the importance of those details over and over again. He now is about to leave SUSTech, and I will miss him with my best wishes.

Thank my companions in CS. I, to be honest, is not that guy that can often carry a project. I am grateful that my partners never give me up. Here are some of their names: Yuchen Wang, Yifan Zhao, Haotian Gao, Yueqian Hu, Xinyu Chen, Yusong Cui, Ruilong Liu, Yongzheng Luo, etc. Thank you guys, wish you all the best in the future, and go for your dreams!

I am grateful for my friends in Shuren College. My university counselors, and the deans of the Shuren College, and schoolmates, gived me many encouragement and stand my all kinds of faults. It is Shuren College that helped me grow a lot and kept a warm heart for life. I also spent many time in Student Union of Shuren, Shuren Peer Mentor and other student groups, and harvest a lot. Thank you, I will always memorize those happy time.

Thank the Career Center of Office of Stuendents Affairs. They offerred me a chance to continue my study in SUSTech, and they admit my talents in administrative work. I am very grateful for that. If not them, I may encounter many troble for finding my next station.

I want to thank my parents. They are not that kind of traditional Chinese parents. In fact they are kind, always ready to listen, willing to help me, and often encourage me. I myself is just a normal person, but I am still fortunate enough to have such parents. I wish they are healthy forever.

Thank you, thank all persons who helped me in the past four years. Forgive me that I may not mention all of you, but I will not forget what you have done.

In the end, I want to thank myself. The only thing I can take control is my whole body, and I am proud that I did not waste it at all. I, though, not able to become the person I imagined, I turned out to be the best guy I can be. Thus, move on boy. There is still a long way wating for me, but I believe that I will go on and on and on.

"The past, is everything we were, don't make us who we are. So I'll dream, until I make it real, and all I see is stars It's not until you fall that you fly. When your dreams come alive you're unstoppable Take a shot, chase the sun, find the beautiful. We will glow in the dark turning dust to gold. And we'll dream it possible possible And we'll dream it possible."