

Aircraft Shape Design Using Artificial Neural Network

Der-Chen Huang,¹ Yu-Fu Lin,¹ Lee-Jang Yang,² and Wei-Ming Chen^{3*}

¹Department of Computer Science and Engineering, National Chung Hsing University, Taichung 402, Taiwan, R.O.C.

²Aeronautical Systems Research Division, National Chung-Shan Institute of Science & Technology,
Taichung 407, Taiwan, R.O.C.

³Department of Information Management, National Dong Hwa University, Hualien 974, Taiwan, R.O.C.

(Received February 28, 2020; accepted June 30, 2020)

Keywords: aerodynamic coefficient, computational fluid dynamics, wind tunnel experiments, artificial neural network

To date, the aerodynamic coefficient of an aircraft has been obtained by computational fluid dynamics (CFD) or wind tunnel experiments, which have a high cost. To reduce the cost and period of analysis, we adopt big data analysis and AI techniques to build an artificial neural network (ANN) and perform learning and training based on historical flight and wind tunnel experiment parameters, so as to predict the aerodynamic coefficient of aircraft. Experimental results show that the values obtained by the proposed method are close to those obtained by wind tunnel experiments. Consequently, the proposed method can effectively reduce the amount of simulation analysis by CFD and wind tunnel experiments.

1. Introduction

In general, the aircraft design procedure includes the design of the configuration, internal arrangement, structure, control law, and each system function, which are closely bound up with each other. The optimum design point to meet the required performance for missions is found after an iterative trade-off.⁽¹⁾ The configuration design is the first part of the overall aircraft design process, and the quality of aerodynamic analysis is also a key factor in the configuration design. Once the first version of the configuration is obtained, the configuration is studied and changed continuously to match the subsequent design requirements, aerodynamic analysis is performed repeatedly, and a massive amount of aerodynamic data is calculated.

The complete aerodynamic data for an aircraft can be obtained by wind tunnel experiments and computational fluid dynamics (CFD).^(2,3) However, wind tunnel experiments are mainly used to verify the design in the final phase of the aircraft design and are inapplicable to design iteration because the hardware facilities of wind tunnel experiments are expensive, the testing and analysis take a long time, and the overall cost is very high. CFD has a complex execution process and many variation factors, such as different mesh-generating tools, e.g., POINTWISE or Hypermesh business software, or different types of CFD software for analysis and calculation, e.g., the SU2 open software of Stanford University, Fluent, and the CFX simulation software of ANSYS. Different results might be obtained using different tools. Moreover, high-

*Corresponding author: e-mail: wmchen88@gms.ndhu.edu.tw
<https://doi.org/10.18494/SAM.2020.2845>

speed hardware is required for the large amount of calculation, and calculations take a long time.

In 1993, Rokhsaz and Steck⁽⁴⁾ used the angle of attack (AOA) and its derivative as the input layer neurons of a back-propagation neural network (BPN) to predict the lift coefficient (CL), drag coefficient (CD), and pitch moment (CM) of wings, so as to provide a reference frame for high-AOA flight. In addition, they applied an artificial neural network (ANN) to perform learning and training continuously based on the correlation between four parameters, namely, pitch angle, pitch angle ratio, velocity, and thrust, to predict the thrust deflection angle. Nevertheless, they only used the AOA and its derivative as eigenvalues and the prediction accuracy was low.

Because the aerodynamic coefficient prediction model based on a complex interpolation method requires a large amount of computer storage and a long computing time, in 2001, Li *et al.*⁽⁵⁾ proposed a fuzzy neural network and used the Mach number and AOA as the input values of the network, and the lift coefficient was the expected output value. The network can approximate any nonlinear function rapidly because of its nonlinear processing capacity and fault tolerance. However, the method might result in local optimization due to data characteristics.

In 2002, Rajkumar and Bardina⁽⁶⁾ adopted the AOA and Mach number as ANN eigenvalues to predict the lift coefficient. The method can predict the lift coefficients of different Mach numbers and AOAs effectively, but it cannot predict other aerodynamic coefficients. Thus, in 2003, they used the speed brake deflection and side slip angles as ANN input values in addition to the AOA and Mach number to predict the lift coefficient, drag coefficient, pitch moment, roll moment, side force, and yawing moment.

In recent years, with the advances in computational capability, the combination of an ANN and different algorithms has increased the aerodynamic prediction accuracy.^(7,8) In 2014, Tao *et al.*⁽⁹⁾ combined a BPN with a hybrid optimization algorithm, wherein the hybrid optimization algorithm comprises the genetic algorithm (GA) and a support vector machine (SVM). The former selected a better ANN weight set for crossover to generate a better network weight set and used mutation to avoid the weight falling into a local optimum. The latter classified the ANN weight set into two groups to validate the correctness of the network weight. If the correctness was lower than a threshold, then the correct network weight set was substituted in the GA again to calculate an optimal weight set (global optimization).

Differing from the method of Tao *et al.*, Gomec and Canibek⁽¹⁰⁾ combined an ANN and the GA to predict six aerodynamic coefficients: the drag coefficient, roll moment, left aileron hinge moment coefficient (CHMAIL), lift coefficient, pitch moment coefficient, and left elevator hinge moment coefficient (CHMEL). The method uses the GA to adjust ANN parameters, i.e., the number of neurons in the hidden layer, the upper and lower limits of the initial hidden layer weight vector, the upper and lower limits of the initial input layer weight vector, the upper limit of the difference, and the initial learning rate. After the GA adjusts these parameters, a more accurate ANN prediction model is obtained.

The method of Tao *et al.* can avoid the ANN weight falling into a local optimum, but it does not have normalized parameters, so a parameter with a large value has a greater effect than that

with a small value, making the learning and training procedures more difficult. For example, most AOAs are larger than the Mach number, so the Mach number becomes an unimportant input value and the AOA determines the lift coefficient. To avoid this problem, here we propose a normalization method and an appropriate learning termination condition based on experiments to further improve the prediction accuracy. Moreover, in contrast to Gomec and Canibek, who only used multiple aerodynamic data for training and prediction without considering changes in the aircraft mock-up, we have considered the modification of the mock-up design to derive more accurate lift and drag coefficients in the iterative design of the aircraft development process.

2. Related Literature

In the 1980s, Searle divided artificial intelligence (AI) into two classes: strong AI and weak AI. In the former, machines and humans have the same complete cognitive ability. In the latter, machines do not have the cognitive ability equivalent to that of the human brain and only have self-learning ability.^(11,12)

AI aims to give computers the ability to think and learn similarly to humans. Machine learning involves the substitution of mass data in algorithms for training and learning to generate a model. It can be the basis of new data prediction and be divided into three classes, i.e., supervised learning, unsupervised learning, and reinforcement learning, as shown in Fig. 1. In supervised learning, data features and labels are substituted in the algorithm for training and learning to build a prediction model. Any new data can be substituted in the model for prediction.

Different from supervised learning, unsupervised learning performs learning and training without labels. An example of unsupervised learning is clustering analysis, which performs clustering according to the correlation coefficient or dissimilarity of data, so that data in the same cluster have a higher correlation and the correlation between clusters is low.

In 1975, Holland proposed the GA,^(13,14) which simulates the evolution rule of the survival of the fittest, where superior species are selected from the current iteration. Its genetic characteristic is exchanged randomly to generate more superior species. The gene information of the optimal species can be obtained after several iterations.

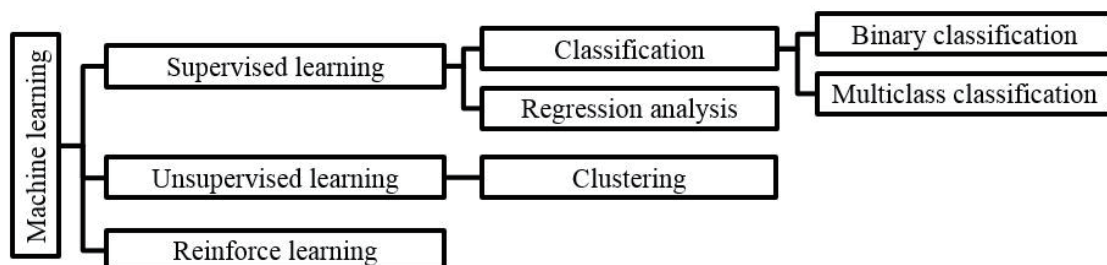


Fig. 1. Machine learning classifications.

A convolutional neural network comprises convolution, pooling, rectified linear unit, and fully connected layers, as shown in Fig. 2.^(15–17) There are two convolution layers, two rectified linear unit layers, one pooling layer, and one fully connected layer. Generally, the number and order of layers are determined by data features; there are more layers for more complex nonlinear problems. The convolution layers are used to extract eigenvalues. As the number of convolution layers increases, high-level features can be extracted from low-level features.

In the sky, an aircraft is subjected to four forces,⁽¹⁸⁾ i.e., lift force, drag force, gravity, and thrust, as shown in Fig. 3. The following explains the four forces. The airflow is separated, where the speed of airflow on the wing's upper surface is high, thus the pressure is low. In contrast, the speed of airflow on the wing's lower surface is low, thus the pressure is high. This phenomenon induces lift force. Drag force is the reverse aerodynamic of the forward direction of an aircraft and depends on factors including friction, shape, induced drag, and air wave. Gravity depends only on the aircraft weight and atmosphere. A low gravity means that the aircraft is light, and it can enhance the performance of aircraft. Thrust is determined by the engine and increases with the performance of the engine.

These forces depend on the environment and the state of the aircraft, determined by the Mach number, Reynolds number, AOA, and angle of slide, where the Mach number is the ratio of the speed of the aircraft (V) to the acoustic velocity (a), i.e., $Ma = V/a$. The Reynolds number is determined by the flight altitude. When the Reynolds number is small, the flow is stable. It is the ratio of the inertia of the flow to the viscosity. A low Reynolds number is suitable for flight. The AOA is the angle between the relative wind and the chord of the wing as shown in

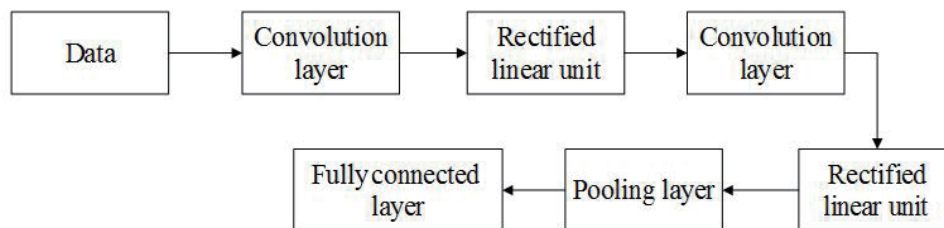


Fig. 2. Convolutional neural network architecture.

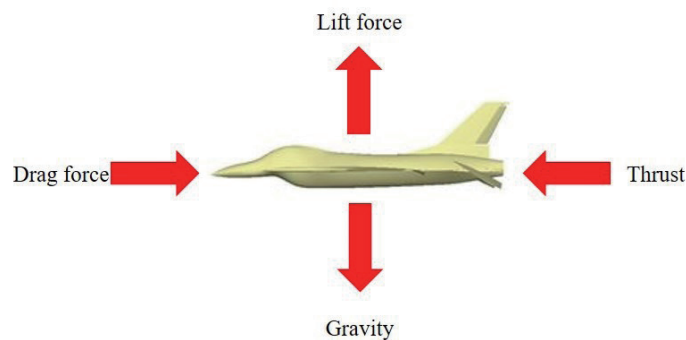


Fig. 3. (Color online) Four forces.

Fig. 4. In general, the lift coefficient increases with the AOA. However, if the AOA is higher than the stall AOA, then the lift coefficient decreases. Therefore, finding the stall AOA is important. In Fig. 5, the angle of side slide is the angle between the longitudinal axis of the body and the forward direction, thereby changing the flight situation.

In aircraft shape design, different calculation tools and tests give different results with different costs and accuracies. In general, considerable amounts of labor and resources are needed to obtain design parameters. Thus, the use of AI technology in aircraft shape design to reduce costs and increase accuracy is the subject of our research.

3. Proposed Method

3.1 System architecture

Figure 6 shows a block diagram of the proposed method. In accordance with the performance and mission specifications, the aircraft size is calculated by considering the wing area and aerodynamic coefficients. The obtained results can be used to generate the outer model line. To obtain the aerodynamic coefficients of an aircraft, three analysis techniques can be used: panel code simulation, CFD simulation, and wind tunnel experiments. Different techniques give different simulation results and have different computational costs. For example, the panel code simulation can efficiently but inaccurately derive the aerodynamic coefficients. In contrast, CFD simulation and wind tunnel experiments can be used to find aerodynamic coefficients with high accuracy. Furthermore, these aerodynamic coefficients can be used as the training parameters for the AI model in our proposed method. The coefficients can also be used as decision information in the design of a flight controller. However, wind tunnel experiments require expensive equipment and CFD simulation requires large amounts of computational resources. To overcome these shortcomings, we used the parameters and results mentioned in Sect. 2 to establish an ANN model for predicting the aerodynamic coefficients to obtain more accurate and useful parameters with less labor. Such data can also be used as decision information during the design of a flight controller.

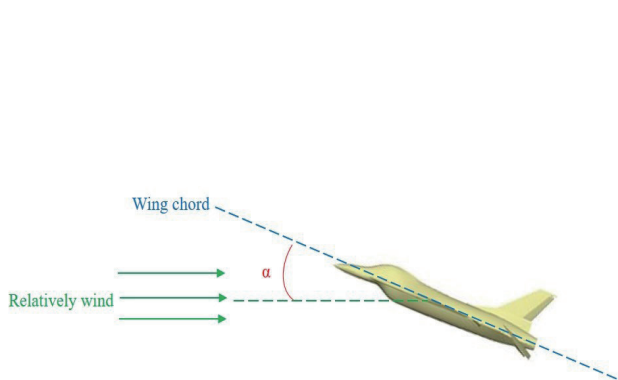


Fig. 4. (Color online) Illustration of AOA.

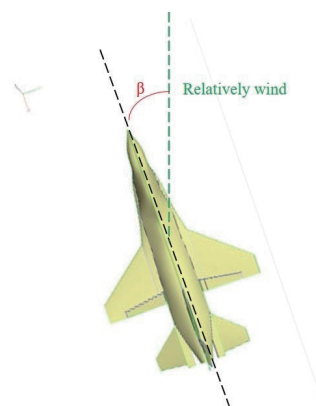


Fig. 5. (Color online) Illustration of angle of side slide.

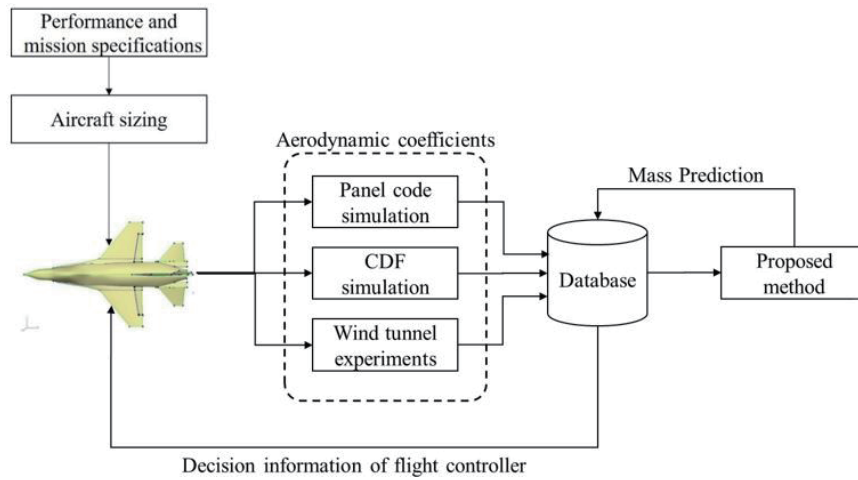


Fig. 6. (Color online) Block diagram of the proposed application.

3.2 Multilayer neural network establishment phase

Figure 7 shows a flowchart of the BPN. Firstly, the war-plane configuration parameters and target values (aerodynamic coefficients) are substituted in the BPN for training and learning. The war-plane configuration parameters include the aspect ratio (AR), AOA, Mach number (Ma), angle of slide (β), Reynolds number, size of tank, taper ratio, and sweepback. The target values of our method are the lift and drag coefficients. After these values are substituted, the weight is adjusted continuously to reduce the prediction error.

Figure 8 shows that the BPN comprises three layers, i.e., input, hidden, and output layers, wherein the input layer comprises eight neurons, i.e., the war-plane configuration parameters; the numbers of hidden layers and neurons determine the BPN weight. A more complex problem requires more layers and neurons. The output layer neuron is the target value, i.e., aerodynamic coefficients, such as the lift and drag coefficients.

The BPN training stage is divided into forward propagation and back-propagation. The former calculates the sum of the product of each eigenvalue and its corresponding weight to obtain the prediction value. The latter adjusts the weight according to the prediction output so as to increase the forecast accuracy. The computing equation is expressed as

$$\mu_{q,j} = \begin{cases} \sum_i x_i w_{i,j} + b, & \text{if } q = 1, \\ \sum_j z_{k,j-1} w_{j,n} + b, & \text{otherwise,} \end{cases} \quad (1)$$

where x_i is the i th neuron of the input layer and $w_{i,j}$ is the connection weight of the i th neuron of the input layer when q is equal to 1. For the j th neuron of the hidden layer, $Z_{k,j-1}$ is the output of the k th neuron in the previous layer, $w_{j,n}$ is the connection weight between the j th neuron of the input layer and the n th neuron of the hidden layer, and b is the bias. The higher the value of $\mu_{q,j}$,

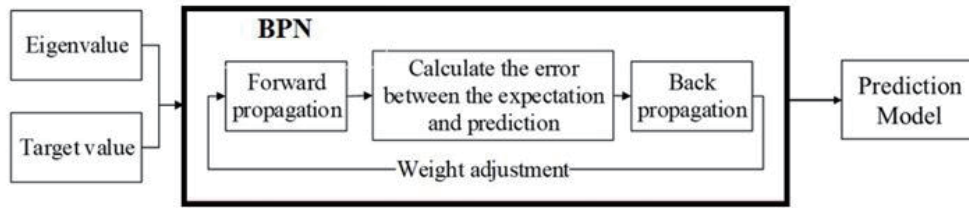


Fig. 7. BPN flowchart.

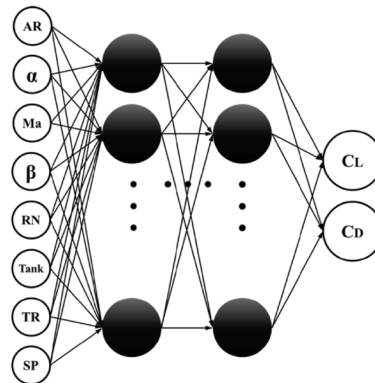


Fig. 8. BPN architecture.

the easier to be activated by a neuron. $Y_{q,j}$ is the j th neuron of the q th layer with $q = 1$ or $q > 1$. Then, $Y_{q,j}$ is substituted in the activation function so that the BPN can solve a complex linear or nonlinear problem, i.e.,

$$Y'_{q,j} = f(Y_{q,j}), \tag{2}$$

where $f(\cdot)$ is the activation function, which may be a linear, hyperbolic tangent sigmoid, or sigmoid function, as shown in Fig. 9, whose equations are respectively expressed as

$$f(Y_{q,j}) = Y_{q,j}, \tag{3}$$

$$f(Y_{q,j}) = \frac{1}{1 + e^{-Y_{q,j}}}, \tag{4}$$

$$f(Y_{q,j}) = \frac{2}{(1 + e^{-2Y_{q,j}}) - 1}. \tag{5}$$

Finally, the products of $Y'_{q,j}$ and the connection weight are summed up to obtain the prediction value

$$O = f\left(\sum_n Y'_{q,n} w_{n,m} + b\right). \tag{6}$$

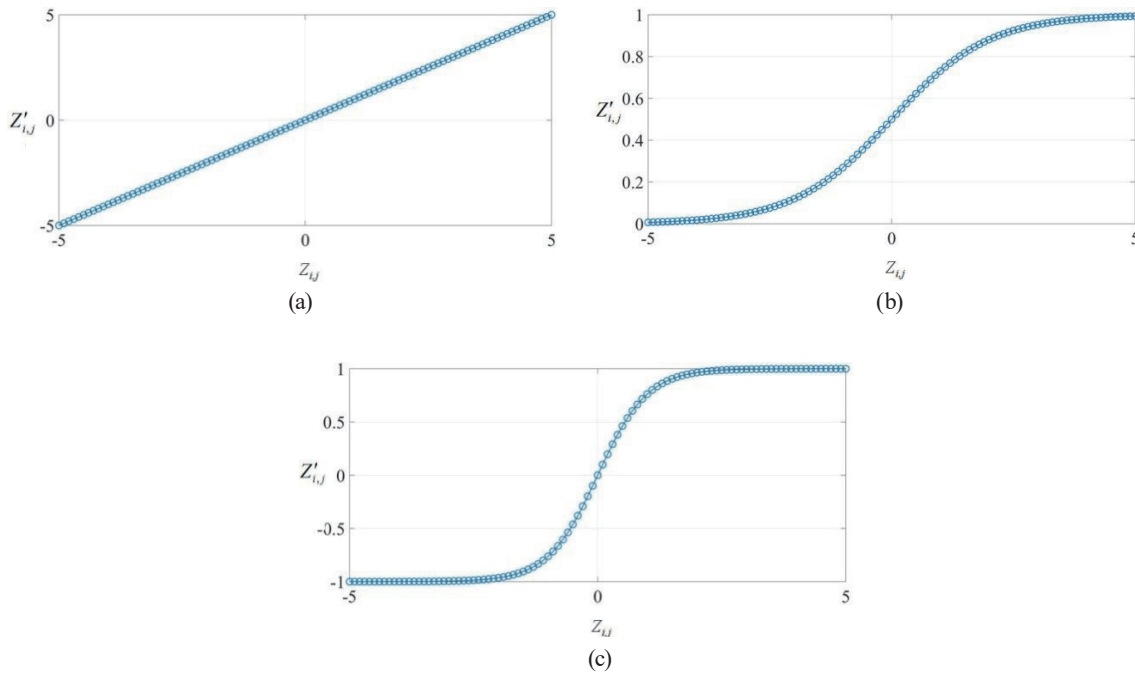


Fig. 9. (Color online) Activation function. (a) Linear function. (b) Sigmoid function. (c) Hyperbolic tangent sigmoid function.

After the forward propagation process is completed, the phase II back-propagation procedure is performed and the weights are adjusted to increase the precision of the prediction value. Finally, the mean square error (MSE) between the output value O and target value G is calculated as

$$MSE = \frac{1}{N} \sum_{i=1}^N (O_i - G_i)^2, \quad (7)$$

where N is the number of engine data.

If the MSE is smaller than a predetermined threshold, the learning and training processes are stopped because there is no significant difference between the prediction and target values. The BPN contains many neuron weights and biases. Here, we assume two weights for simplicity. We adopt the gradient steepest descent method to adjust the weights to reduce the MSE, as shown in Fig. 10, where $w_{1,1}$ and $w_{1,2}$ are the two weights. To find the minimum value, the local optimization information must be used along with the gradient of the descent toward the global optimal solution.

The adjustment method calculates the error vector $\delta_{q,j}$ of the hidden layer and the error vector δ_1 of the output layer as

$$\delta_{j,n} = \sum_j \delta_1 w_{j,n} \times Y'_{q,j} \times (1 - Y'_{q,j}), \quad (8)$$

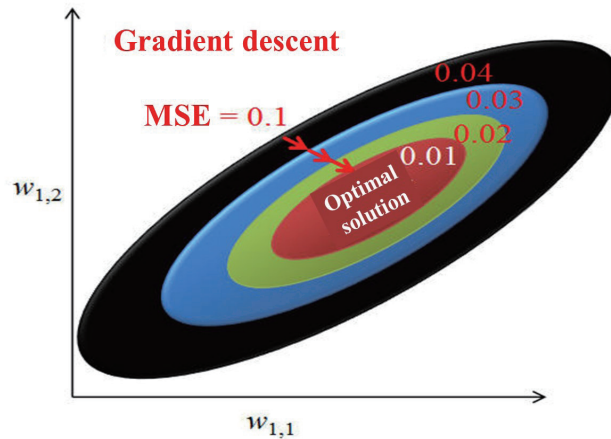


Fig. 10. (Color online) Schematic diagram of gradient descent method.

$$\delta_1 = (R - O) \times O(1 - O). \quad (9)$$

The error vector is multiplied by the predetermined learning rate (η), and the connection weight correction vector is obtained as follows:

$$\Delta w_{n,m} = \eta \times \delta_1 \times Y'_{q,j}, \quad (10)$$

$$\Delta w_{q,j} = \begin{cases} \eta \times \delta_{j,n} \times x_i^i, & \text{if } q = 1, \\ \eta \times \delta_{j,n} \times Y_{q,j-1}, & \text{if } q > 1. \end{cases} \quad (11)$$

Then, the connection weight is updated using the connection weight correction vector, i.e.,

$$w'_{n,m} = w_{n,m} + \Delta w_{n,m}, \quad (12)$$

$$w'_{j,n} = w_{j,n} + \Delta w_{j,n}. \quad (13)$$

These steps are repeated until the MSE is smaller than the threshold so as to complete the construction of the BPN to obtain a high-accuracy forecast.

3.3 BPN prediction phase

After the BPN is constructed, the war-plane configuration parameter eigenvalues are imported into the BPN to predict the aerodynamic coefficient, as shown in Fig. 11. Figure 12 shows an example of aerodynamic coefficient prediction, where two eigenvalues $\{0.3, 0.5\}$ are multiplied by the corresponding weights $\{0.3, 0.2\}$, i.e., $Y_{1,1} = 0.3 \times 0.3 + 0.5 \times 0.2 + 0.2 = 0.39$, and substituted in the activation function $f(Y_{1,1}) = 1 / (1 + e^{-0.39}) = 0.59628$. Then, the second neuron of the first hidden layer is calculated in the same way, i.e., $Z_{1,2} = 0.3 \times 0.3 + 0.5 \times 0.1$

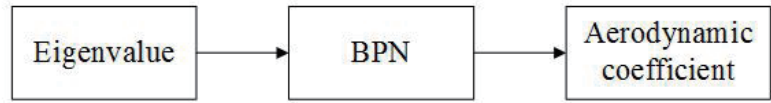


Fig. 11. Flowchart of aerodynamic coefficient prediction.

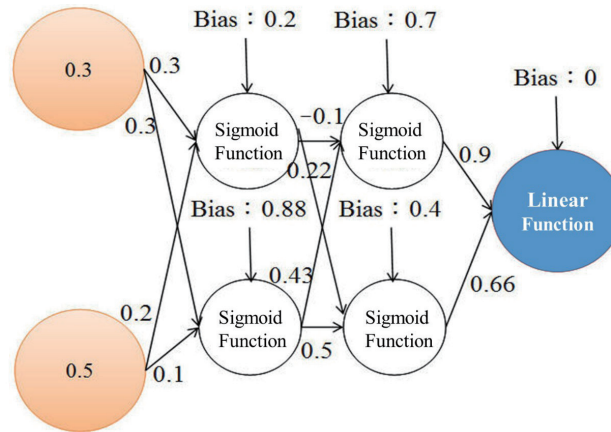


Fig. 12. (Color online) Example of aerodynamic coefficient prediction.

+ 0.43 = 0.57, and substituted in the activation function, i.e., $f(Y_{1,2}) = 1 / (1 + e^{-0.57}) = 0.6387$.

The two activated values (0.59628, 0.6387) are substituted in the first neuron of the second hidden layer to compute the sum of the products, i.e., $Y_{2,1} = 0.59628 \times -0.1 + 0.6387 \times 0.43 + 0.7 = 0.915$, and substituted in the activation function for the calculation, i.e., $f(Y_{2,1}) = 1 / (1 + e^{-0.915}) = 0.714$. Then, the second neuron of the second hidden layer is $Y_{2,2} = 0.59628 \times 0.22 + 0.6387 \times 0.5 + 0.4 = 0.8505$, which is substituted in the activation function to obtain $f(Y_{2,2}) = 1 / (1 + e^{-0.8505}) = 0.7006$. These values (0.714, 0.7006) are substituted in the output layer neuron for the calculation, i.e., $O = 0.714 \times 0.9 + 0.7006 \times 0.66 + 0 = 1.104$. As shown above, the BPN prediction method is implemented in our research. As reported in the next section, we use wind tunnel data for training and compare the results of different functions to find the best solution.

4. Experimental Results

To measure the efficiency effectively, we adopt the mean absolute error (MAE) for evaluation, whose equation is

$$MAE = \frac{1}{N} \sum_{i=1}^N |A_i - A'_i|, \quad (14)$$

where N is the number of test data, A_i is the aerodynamic coefficient, e.g., lift or drag coefficient, and A'_i is the prediction value of the proposed method.

Table 1 shows the high-speed dynamometer check matrix of an F1 aircraft wind tunnel test, which has many variables, such as AOA (α), angle of slide (β), and control surface (LEF , TEF , DH , DR , SB). For each Mach number, the test includes a set of data, and every test is named with a different number such as 97, 98, 99, ... etc. For example, round 99 is the test data including the Mach number 0.9, AOA (α) of -4 to 15° , and nose flap and rear flap angles of 10° . We use round 99 as testing data, and data of other rounds are used as training data to verify the proposed method.

In the first experiment, we compare the lift coefficient forecast accuracies of different activation functions: a linear function, a sigmoid function, and a hyperbolic tangent function. The number of hidden layers is fixed to 1 and the number of neurons is set as 5 to compare the forecast accuracies. Figure 13 shows the prediction value obtained by the proposed method and the aerodynamic coefficient from the wind tunnel experiment. The lift coefficient predicted by the linear function in our method is significantly different from that obtained by the traditional wind tunnel experiment. Because the prediction of the aerodynamic coefficient is a highly nonlinear problem, the linear activation function is not suitable for aerodynamic coefficient prediction. In addition, the numerical error of the hyperbolic tangent function is larger than that of the sigmoid function, so we choose the sigmoid function as the activation function for the next phase.

Table 1
(Color online) F1 aircraft high-speed dynamometer check matrix.

CONF.	α	B	LEF	TEF	DH	DR	SB	MACH number						
								0.6	0.8	0.9	0.95	1.05	1.2	
F1	A2	0	0	0	0	0	0	97	98	99	100	100	101	
									114				113	
								256	257	258			259	
									117	191	118	119		
									189					
									200	235				
									236					
	A3	10	15	12.5	20	12.5	20	12.5			192			
											231			
											195			
											232			

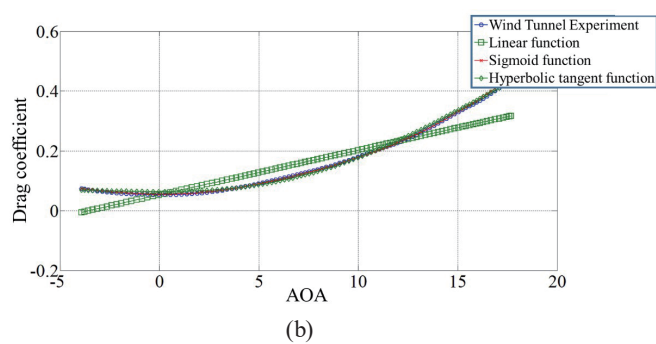
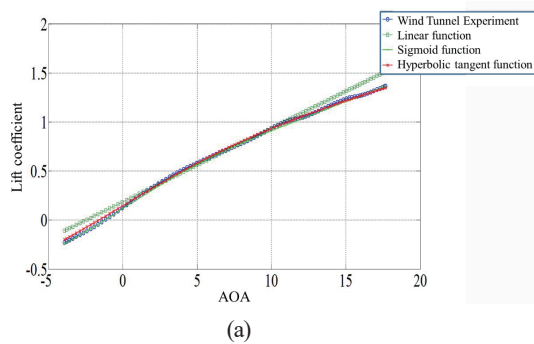


Fig. 13. (Color online) Aerodynamic coefficients with different activation functions and AOAs. (a) Lift coefficient under different AOAs. (b) Drag coefficient under different AOAs.

In the second experiment, we compare the forecast accuracies obtained for different numbers of neurons in the hidden layer. As shown in Figs. 14(a) and 15(b), we use four, five, and six neurons in the hidden layer to obtain the lift and drag coefficients for different AOAs, respectively. The figure shows that the lift and drag coefficients are proportional to the AOA. By adjusting the number of neurons to evaluate the effect of neuron number on the MAE, it is shown that the MAE of the lift coefficient is significantly greater than that of the drag coefficient; thus, the MAE of the lift coefficient can be used as a basis to determine the number of neurons.

As shown in Fig. 14(a), when the AOA is larger than 13° , there is a significant difference between the lift coefficient obtained with the sigmoid function with five or six neurons and that obtained in the wind tunnel experiment. A similar problem for the drag coefficient with four, five, and six neurons is shown in Fig. 14(b). As shown in Fig. 15, the MAEs are 0.02 for four neurons, 0.0238 for five neurons, and 0.94 for six neurons, indicating that the MAE is roughly proportional to the number of neurons. This is because more neurons may result in overfitting. The number of neurons in the hidden layer should be set as four such that the training and prediction times can be shortened.

In the third experiment, we compare the lift and drag coefficients for different numbers of neurons in the hidden layers. Figure 16 shows that if the number of hidden layers is increased, then the error between the expected value obtained by our method and that obtained in the wind

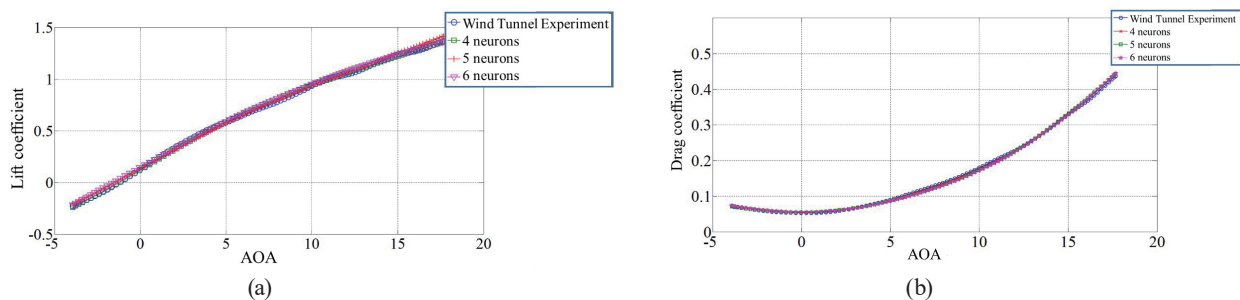


Fig. 14. (Color online) Aerodynamic coefficients for different numbers of neurons and AOAs. (a) Lift coefficient under different AOAs. (b) Drag coefficient under different AOAs.

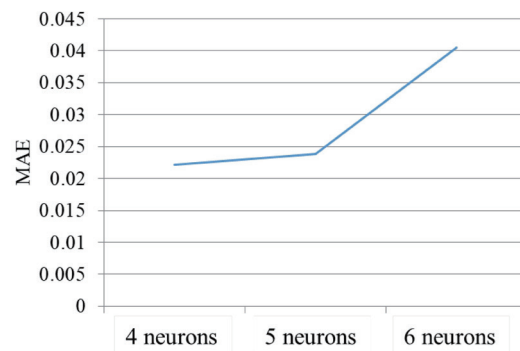


Fig. 15. (Color online) MAEs of lift coefficient for different numbers of neurons.

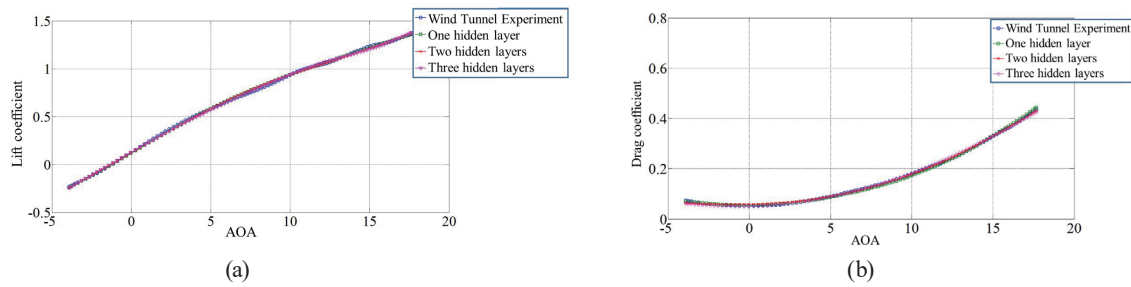


Fig. 16. (Color online) Aerodynamic coefficients with different numbers of hidden layers and AOAs. (a) Lift coefficient under different AOAs. (b) Drag coefficient under different AOAs.

Table 2
(Color online) Dynamometer check matrix for configuration optimization of F1 aircraft with tank.

CONF.	α	B	LEF	TEF	DH	DR	SB	MACH Number									
								0.6	0.8	0.9	0.95	1.05	1.2				
F1+CTK-1	A2	0	0	0													
		10															
F1+CTK-4		0															
		10															
F1+CTK-5		-10															

Table 3
Tank dimensions.

Form fit tank code	Height × Width × Length (inch)
CTK-1	1.643 × 1.908 × 14.401
CTK-4	1.667 × 1.908 × 13.879
CTK-5	1.631 × 1.961 × 13.698

tunnel experiment is also increased. The main cause of this is data overfitting. The MAE for one hidden layer is 0.02, and the MAE for three hidden layers is increased to 0.03. Thus, the number of hidden layers is set to one in this research.

Table 2 shows the dynamometer check matrix for the configuration optimization of F1 aircraft with different tank sizes. Similarly to Table 1, the check matrix uses many variables and different Mach numbers to obtain a set of test data. Our fourth experiment adopts the optimal training parameters in the former three experiments to predict the aerodynamic data for different tank sizes (different configurations); F1 aircraft have several types of tanks with different mount points, and the configuration affects the accuracy of aerodynamic coefficient prediction. The tank size is used as an input parameter in this experiment, as shown in Table 3. Tanks CTK-1 and CTK-4 are taken as training data. Round 120 CTK-5 is used as the test data.

Figure 17 shows the experimental results. The mean error of the lift coefficient is 0.0046 and the maximum error is 0.0128. These values are smaller than the tolerable error of 0.03. The mean error of the drag coefficient is 0.0028 and the maximum error is 0.0055, indicating that the proposed method has a reasonably high prediction accuracy and the error can be controlled in all cases.

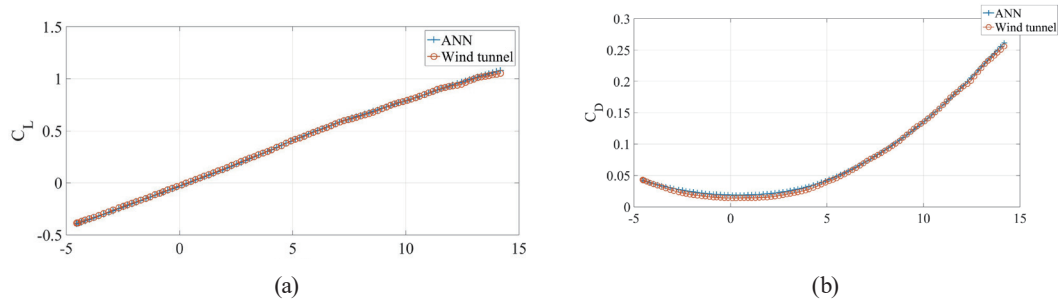


Fig. 17. (Color online) Predicted aerodynamic coefficients for different gas tanks. (a) Lift coefficient under different AOAs. (b) Drag coefficient under different AOAs.

5. Conclusions

The proposed method predicts the lift coefficient more accurately as the number of neurons in the hidden layer is increased. The experimental results also show that the proposed method has a smaller error if the learning rate is increased. The proposed method can be used as an alternative to CFD software and the wind tunnel test to reduce the experimental cost and shorten the design period of the aerial vehicle configuration. In addition, the parameters are first normalized before the lift coefficient is predicted. Our method only predicts the lift coefficient instead of predicting the lift and drag coefficients simultaneously since the lift coefficient is more important than the drag coefficient. In addition, many important features such as the Mach number, full AOA, pitch angle, and Reynolds number are selected as the input of the proposed ANN model, increasing the accuracy of the estimated lift coefficients.

This method is applicable to aerodynamic analysis for different aerial vehicles. The proposed prediction method is established on the basis of big data analysis and AI technology to reduce the development cost, labor, and design period.

In the future, we will add more shape parameters as inputs for training such as the wing area, airfoil, and size of fuselage. We hope that the shape parameters can be used to describe the shape of aircraft more clearly. Moreover, we expect that the training results can specifically reflect the effects of shape changes and accelerate the design of aircraft.

References

- 1 P. R. Daniel: Aircraft Design: A Conceptual Approach, 6th ed. American Institute of Aeronautics and Astronautics Inc (1992) 515–552. <https://doi.org/10.2514/4.104909>
- 2 A. A. de Paula, F. D. Magalhães Porto, and M. S. Sousa: AIAA Modeling and Simulation Technologies Conf. (2016) 4133. <https://doi.org/10.2514/6.2016-4133>
- 3 F. Nicolosi, D. Ciliberti, P. Della Vecchia, and S. Corcione: 2018 Applied Aerodynamics Conf. (2018) 2855. <https://doi.org/10.2514/6.2018-2855>
- 4 K. Rokhsaz and J. E. Steck: J. Guidance Control Dynam. **16** (1993) 934. <https://doi.org/10.2514/3.21104>
- 5 S. L. Li, F. T. Zhang, Z. Li, and K. H. Wei: Acta Simulata Systematica Sinica **4** (2001) 24.
- 6 T. Rajkumar and J. E. Bardina: FLAIRS Conf. (2002) 242–246.
- 7 S. I. Horikawa, T. Furuhashi, and Y. Uchikawa: IEEE Trans. Neural Netw. **3** (1992) 801.
- 8 C. T. Lin and C. G. Lee: Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems (Prentice Hall, 1996).

- 9 C. Tao, D. Chen, W. Songyan, and Y. Ming: Proc. 11th World Congr. Intelligent Control and Automation (2014) 4961–4964.
- 10 F. S. Gomec and M. Canibek: Proc. 7th European Conf. Aeronautics and Space Sciences (2017).
- 11 S. J. Russell and P. Norvig: Artificial Intelligence: a Modern Approach (Malaysia, Pearson Education Limited, 2016).
- 12 D. K. Lin: Practical Application of TensorFlow+Keras Deep Learning AI (DrMaster Press, 2017).
- 13 L. Davis: Handbook of Genetic Algorithms (Van Nostrand Reinhold, 1991).
- 14 M. T. Su, and H. T. Chang: Machine Learning: Artificial Neural Network, Fuzzy System and Genetic Algorithm (Chuan Hwa Books, 2016) 4th ed.
- 15 J. Long, E. Shelhamer, and T. Darrell: IEEE Conf. Computer Vision and Pattern Recognition (2015) 3431–3440.
- 16 Y. Kim: Convolutional neural networks for sentence classification. arXiv preprint arXiv (2014) 1408.5882.
- 17 S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back: IEEE Trans. Neural Netw. **8** (1997) 98.
- 18 A. Sóbester and A. I. Forrester: Aircraft Aerodynamic Design: Geometry and Optimization (John Wiley & Sons, 2014).

About the Authors



Der-Chen Huang received his B.S. degree in electronic engineering from Fung Chia University, Taiwan, in 1983, his M.S. degree in computer engineering from Florida Institute of Technology, U.S.A., in 1991, and his Ph.D. degree in computer engineering from the Department of Computer Science and Information Engineering, Chung-Cheng University, Chiayi, Taiwan, R.O.C. in 2000. From 1983 to 1989, he worked as a design engineer with the Computer Communication Lab (CCL)/Industrial Technology Research Institute (ITRI) and National Chung-Shan Institute of Science and Technology (CSIST) when he was assigned to a partnership project at General Dynamics, Fort Worth, Texas, U.S.A. He was an associate professor with the Department of Electronic Engineering, National Chinyi Institute of Technology, Taichung, Taiwan, R.O.C. from 1991 to 2004. In 2004, he joined the Department of Computer Science and Engineering, National Chung Hsing University, Taichung, Taiwan, R.O.C. He was a director of the Computer and Information Center of Chung Hsing University from 2007 to 2011. Currently, he is a professor of Chung Hsing University. Dr. Huang served as a reviewer for various technical journals and conferences and a member of the editorial board of Journal of Internet Technology. He received the Best Paper Award from the 5th International Conference on Future Information Technology, Korea, in 2010. His research interests include VLSI design for testability and diagnosis, VLSI digital signal process, communication, image, and artificial intelligence. (huangdc@nchu.edu.tw)



Yu-Fu Lin received his M.S. degree in 2012 from the Institute of Computer and Communication Engineering, National Cheng Kung University, Tainan, Taiwan. He is currently pursuing his Ph.D. degree in Computer Science and Engineering at National Chung Hsing University, Taichung, Taiwan. He has been working at National Chung-Shan Institute of Science & Technology (NCSIST) in the area of intelligent computing technologies since 2012. He is currently the leader of the Unit of Data Application at the Aeronautical Systems Research Division of NCSIST. His current research is aimed at aircraft design and military application in artificial intelligence. (yufu35@gmail.com)



Lee-Jang Yang received his M.S. degree from the Institute of Applied Mechanics at National Taiwan University (NTU) of Taiwan, in 1991, and his Ph.D. degree from the Department of Computer Science and Engineering at National Chung-Hsing University (NCHU) of Taiwan, in 2015. He has been working at the National Chung-Shan Institute of Science & Technology (NCSIST) in the area of intelligent computing technologies since 1991. He is currently the chief of the Department of Information & Network Technology at the Aeronautical Systems Research Division of NCSIST. His current research is aimed at the creation and study of Semantic Web technology for Linked Data, military information fusion technology, and military application in artificial intelligence. (calsasrd@hotmail.com)



Wei-Ming Chen is currently a professor in the Department of Information Management at National Dong Hwa University. He was a professor in the Department of Computer Science and Information Engineering and the Chairman of Computer Science and Information Engineering at National Ilan University. He received his M.E. and Ph.D. degrees in Computer Science and Information Engineering from National Chung Cheng University. His current research interests include image processing, computer networks, multimedia systems, and artificial intelligence. (wmchen88@gms.ndhu.edu.tw)