pp 223–243. © The Author(s), 2020. Published by Cambridge University Press on behalf of Royal Aeronautical Society.
doi:10.1017/aer.2020.83

Flow field reconstruction method based on array neural network

W. Yuqi
Renmin University of China
Beijing 100872
China

Y. Wu, **L. Shan and Z. Jian**
yuanwu@sccas.cn
Computer Network Information Center
Chinese Academy of Sciences
Beijing 100190
China

R. Huiying and Y. Tiechui
Computer Network Information Center
Chinese Academy of Sciences
Beijing 100190
China

University of Chinese Academy of Sciences
Beijing 100049
China

K. Menghai
China Unicom Software Research Institute
Beijing 100176
China

ABSTRACT

Multi-dimensional aerodynamic database technology is widely used, but its model often has the curse of dimensionality. In order to solve this problem, we need projection to reduce the dimension. In addition, due to the lack of traditional method, we have improved the traditional flow field reconstruction method based on artificial neural networks, and we proposed an array neural network method.

In this paper, a set of flow field data for the target problem of the fixed Mach number is obtained by the existing CFD method. Then we arrange all the sampled flow field data into a matrix and use proper orthogonal decomposition (POD) to reduce the dimension, whose size is determined by the first few modals of energy. Therefore, significantly reduced data

Received 11 December 2019; revised 24 June 2020; accepted 20 August 2020.

https://doi.org/10.1017/aer.2020.83 Published online by Cambridge University Press

are obtained. Then we use an arrayed neural network to map the flow field data of simplified target problem and the flow field characteristics. Finally, the unknown flow field data can be effectively predicted through the flow field characteristic and the trained array neural network.

At the end of this paper, the effectiveness of the method is verified by airfoil flow fields. The calculation results show that the array neural network can reconstruct the flow field of the target problem more accurately than the traditional method, and its convergence speed is significantly faster. In addition, for the case of high angle flow field, the array neural network also performs well. There are no obvious jumps, and huge errors are found in results. In general, the proposed method is better than the traditional method.

Keywords: neural network; flow field reconstruction; proper orthogonal decomposition; machine learning

NOMENCLATURE

ANN	artificial neural networks
POD	projection orthogonal decomposition method
SVD	singular value decomposition
CFD	computational fluid dynamics
A	flow field data matrix
M_h	the flow field data dimension
M^s	the number of samples
U,V	an orthogonal matrix
podbasic	the number of modal
$\hat{U}$	podbasic columns of U
$\mathbb{R}$	real domain
X	input of ANN
Y	output of ANN
$\tilde{Y}$	the approximation of Y
W	the weight of the output of ANN
B	the bias of ANN
S_i	the solution vector of flow field data
E	function of the squared error

Greek Symbol

Σ	singular value matrix
$\hat{\Sigma}$	podbasic lines of $\hat{\Sigma}$
Λ	equals ΣV^T
$\hat{\Lambda}$	podbasic lines of Λ
Υ	energy function
θ	the activation function
Ξ	parameter set
Phi	the mapping

https://doi.org/10.1017/aer.2020.83 Published online by Cambridge University Press

1.0 INTRODUCTION

Nowadays, multi-dimensional aerodynamic databases have been widely used in design, optimisation, control and other related research fields, which have high research value and practical significance. Among them, multi-dimension represents multiple parameters, such as flying height, Mach number, angle of attack, etc. However, the amount of data required often exponentially increases with the dimension, which we called the curse of dimensionality[(1)]. Thus, the multi-dimensional nature will make the calculation amount very large. In order to reduce the computational overhead, data dimension reduction is particularly important. The projection-based dimension reduction method is a relatively common projection method[(2)]. The most attractive point of this method is that it can save the model structure and ensure the universality of the method. It is especially important for data-based models because the model can be improved in time and applied to hidden conditions that cannot be observed during the model changes. There are two main methods based on projection, including orthogonal decomposition method (POD)[(3)] and greedy algorithm[(4)]. The POD method requires a sample in a parameter space, then it needs singular value decomposition, and constructs a space that requires less modality, while the greedy algorithm selects samples in an adaptive and minimising error manner[(10)]. Therefore, the greedy algorithm should be more efficient than the POD algorithm; however, the criteria for sample selection in practical applications are not easy to determine, which leads to greedy algorithms that are not always feasible.

In addition, since the new space constructed based on the projection model uses only a few modals, this method also has significant advantages in terms of computational efficiency. Furthermore, due to the differences in operating platforms and the affine dependence of the right-hand rule, the parameters of the projection model are generally not suitable for general nonlinear problems. Hesthaven et al.[(4)] and Sha et al.[(5)] proposed an improved POD method, which approximates the nonlinear properties of POD bases and new parameters. This method is not described in detail in this paper. The detailed study on this model can be found in Ref. (6).

And in other references, Per Christian Hansen[(37)] proposed that the truncated singular value decomposition (SVD) can be considered as a method for regularisation of ill-posed linear least squares problems. And he introduced some basic theory of SVD. Shnayderman, A., Gusev, A. and Eskicioglu, A.M.[(38)] introduced the application of the SVD method in image processing. In fact, image processing also means processing the data matrix. Drma and Veseli[(39)] introduced how to break the barrier between numerical accuracy and run-time efficiency in computing the fundamental decomposition of numerical linear algebra — the singular value decomposition (SVD) of general dense matrices. G. Berkooz, P. Holmes and J.L. Lumley[(40)] introduced the basic theory of the POD method in detail. CW. Rowley[(41)] compared three different methods of model reduction: proper orthogonal decomposition (POD), balanced truncation and a method called balanced POD. K. Kunisch and S. Volkwein[(42)] proved error estimates for Galerkin proper orthogonal decomposition (POD) methods for nonlinear parabolic systems arising in fluid dynamics. Y.C. Liang and other authors[(43)] gave a summary of the POD methods and showed the connections among these methods.

In recent years, data-based models have been widely favoured by scholars in various fields. Data-based intelligent learning algorithm (also known as machine learning) has developed rapidly, such as Refs 11 and 12. Its basic idea is to collect sample data and directly analyse its underlying mechanism. Machine learning methods are widely used in the fields of image processing, speech recognition, drug research, etc., which show great potential for scientific computing in various fields. It is well known that artificial neural network (ANN) is one of the

https://doi.org/10.1017/aer.2020.83 Published online by Cambridge University Press

machine learning algorithms. The concept of artificial neural network originates from biology. Neural network consists of simple but interrelated elements called neurons[23]. ANNs are usually non-linear and are suitable for a variety of classification and regression problems, and can approximate any linear and nonlinear functions with different complexity[24,25]. The role of ANN is to form different functional relationships between vectors. As an important non-parametric data-driven model, ANN can make full use of the data and determine the structure and parameters of the model from the data without any restrictions and assumptions. Thus, it can obtain good prediction results.

Computational fluid dynamics, as an important branch of scientific computing, incorporates many machine learning strategies. The field of fluid field remodelling in computational fluid dynamics (CFD) includes turbulence models[13–16], transition models[17], flow control[18], unsteady aerodynamic prediction[19], etc. This paper mainly considers the flow field reconstruction problem (that is to say, reconstructs the flow field from the collected high-precision data samples). Bui-Thanh et al. also did related research work[20]. In addition, Luo Jie et al.[7] and Qiu Yasong et al.[8] also did related research on eigen-to-orthogonal decomposition and discussed in detail the criteria for modal selection of energy and modal numbers in flow places. In these studies, a flow field prediction correlation method based on intrinsic orthogonal decomposition was proposed. Hu Zhi et al.[9] analysed the dynamic modal decomposition applied to different degrees of dynamic stall flow field, and some scholars proposed a method combining interpolation with improved orthogonal decomposition, etc. The improved orthogonal decomposition is called gappy POD[21] (gappy POD method generates high-precision samples multiple times at the beginning and then interpolates for the parameters sought). After that, gappy POD extended to unsteady flow reconstruction problems[22] and so on.

In this paper, an array neural network method is proposed based on the flow field reconstruction method, which uses the artificial neural network[10]. We obtain a set of high-precision samples through CFD calculation, which we called the target problem. Then we simplify the problem and reduce the amount of data by POD method. At the same time, we seek to approach the target problem with a certain degree of accuracy. The simplified problem corresponds to the sample of the target problem. Then we hope to use the array neural network to establish a mapping between the flow field features of simplified target problem and the flow field data. We first train the neural network with the known samples. For the unknown flow field data, the trained neural network is used to predict the simplified problem. Finally, we use the reverse POD process to restore the simplified samples to the high-precision samples.

In this paper, we describe the calculation methods in Sections 1 and 2 at first, including POD methods and ANN methods. In the third section of this paper, the flow field reconstruction method based on array neural network is proposed. Then in the fourth section, several typical airflow flow field examples are given, and the figures of calculated reconstruction flow field are shown. What's more, we analyse the energy of modals. Finally, we give some conclusions in the fifth section.

2.0 POD METHOD

In this section, we describe the POD method. The POD method is actually divided into two parts, singular value decomposition and projection transformation. First, we perform singular value decomposition (SVD) on the sample matrix; then we select the appropriate number of modals for projection. Finally, we obtain the simplified sample data.

https://doi.org/10.1017/aer.2020.83 Published online by Cambridge University Press

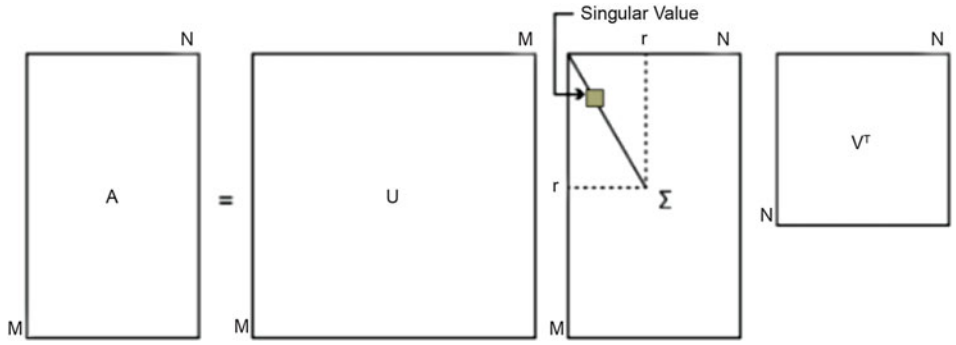


Figure 1. SVD.

2.1 SVD matrix decomposition

Singular value decomposition is an important matrix decomposition in linear algebra and has important applications in signal processing and statistics[37]. Singular value decomposition is similar to the diagonalisation of symmetry matrices or Hermit matrices based on eigenvectors in some respects. However, although these two matrix decompositions are related, they are significantly different. The basis of symmetric vector decomposition is spectral analysis, while singular value decomposition is the generalisation of spectral analysis theory on arbitrary matrices[38].

SVD is essentially the decomposition of matrix, which is different from eigendecomposition. SVD doesn't require that the matrix must be a square matrix[38]. Suppose A is an $M_h \times M^s$ matrix, where M_h is the flow field data dimension, M^s is the number of samples, and all of the elements in A belong to the domain K which is the real or complex domain. Thus, there is a decomposition that makes

$$A = U\Sigma V^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

where U is an $M_h \times M_h$ orthogonal matrix, $\Sigma = diag(\sigma_1, \cdots, \sigma_r)$ is a matrix of $M_h \times M^s$, its elements are all zero, except the elements on the main diagonal. On the main diagonal of Σ each element is a singular value ($r \leq min(M_h, M^s)$), and V is an $M^s \times M^s$ orthogonal matrix.

Figure 1 visually shows the definition of SVD above.

In addition, both U and V are unitary matrix, which satisfied

$$U^T U = I, \quad V^T V = I$$

Among them, the columns of V constitute a series of orthogonal "input" base vectors, which are the feature vectors of $A^T A$. We obtain these vectors through matrix operation. First, we multiply A^T and A to obtain a square matrix $A^T A$ of $M^s \times M^s$. Since $A^T A$ is a square matrix. It can perform eigenvalue decomposition, whose feature values and feature vectors satisfy

$$\left(A^T A\right) v_i = \lambda_i v_i$$

Thus, M^s eigenvalues and the corresponding M^s eigenvectors v_i of $A^T A$ are obtained. Then we form all the feature vectors of $A^T A$ into an $M^s \times M^s$ matrix V. Thus, matrix V is obtained.

https://doi.org/10.1017/aer.2020.83 Published online by Cambridge University Press

Similarly, the columns of U form a set of orthogonal "output" base vectors, which are eigen vectors of AA^T. The method of getting U is similar to V.

The elements on the diagonal of Σ are singular values that can be thought of as scalar "expansion controls" between input and output[39]. These elements are the non-negative square roots of the eigenvalues of A^TA and AA^T, which correspond to the row vectors of U and V. We obtain these elements though the method as following

$$\begin{aligned} A &= U\Sigma V^T \\ \Rightarrow AV &= U\Sigma V^T V \Rightarrow AV = U\Sigma \\ \Rightarrow Av_i &= \sigma_i u_i \Rightarrow \sigma_i = Av_i/u_i \end{aligned}$$

Then we can obtain the singular value matrix Σ.

2.2 Projection transformation

For the convenience of calculation, we multiply Σ and V^T, which are among the three matrices obtained by SVD decomposition[10]:

$$A = U \times \Sigma V^T = U\Lambda$$

where ΣV^T is a matrix of $M_h \times M^s$.

Then we consider the projection transformation[28]. Suppose that the number of modal (base) we choose is *podbasic*, that is, we select the *podbasic* lines of Σ (recorded as $\widehat{\Sigma}$) and multiply it by V^T, which in order to get the matrix $\widehat{\Sigma}V^T$ of *podbasic* $\times M^s$(recorded as $\hat{\Lambda}$). However, for the convenience of the program operation, we choose its equivalent form. We select the first *podbasic* columns of U (denoted as $\widehat{U}$), and left multiply its transposition by matrix A. Here, we choose the transposition is because U is an orthogonal matrix. There is a theorem can prove that the transposition of an orthogonal matrix is equal to the inverse of an orthogonal matrix[40–43]. Therefore, taking the transposition is equal to taking the inverse matrix. Finally, we can obtain the equivalent form $\widehat{U}^T A$ of $\widehat{\Sigma}V^T$. That is

$$U^TA = \Sigma V^T \Rightarrow \widehat{U}^TA = \widehat{\Sigma}V^T$$

As shown in the following Fig. 2.

In fact, the value of *podbasic* is determined by the metric ε_{POD}

$$\Upsilon(podbasic) = \frac{\sum_{m=1}^{podbasic} \sigma_m^2}{\sum_{m=1}^{M^s} \sigma_m^2} \geq 1 - \varepsilon_{POD}^2$$

where $\Upsilon(podbasic)$ represents the percentage of energy retained under the modal (base), the number of modal is *podbasic*, we called it the generalised "energy"[10]. If $\hat{A}$ meets the formula above, then we have

$$\frac{\|\hat{A} - A\|_F}{\|A\|_F} \leq \varepsilon_{POD}$$

where $\|\cdot\|_F$ represents the Frobenius norm.

However, in fact, we only calculate the coefficient matrix $\hat{\Lambda} = \widehat{U}^TA$ when we train the ANN.

https://doi.org/10.1017/aer.2020.83 Published online by Cambridge University Press

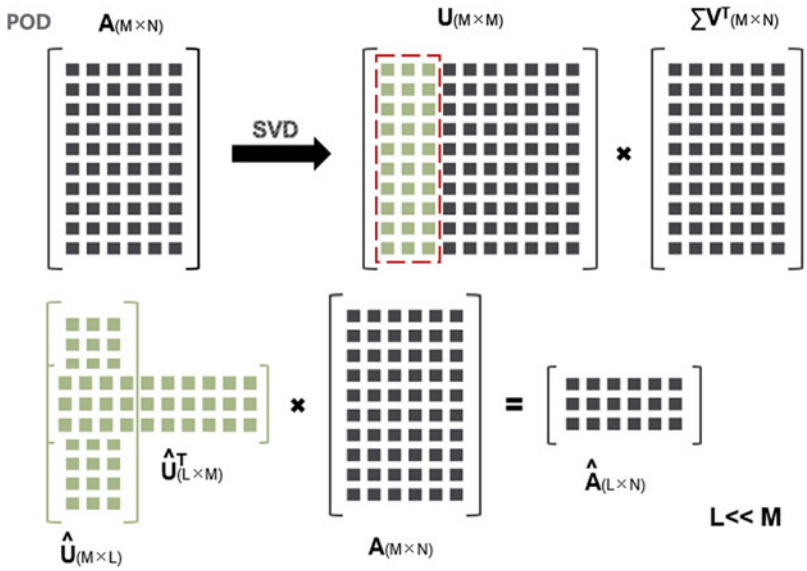


Figure 2. POD.

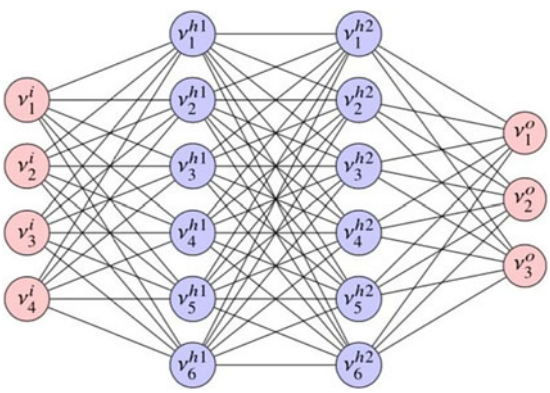


Figure 3. Multi-layer feedforward neural network[10].

3.0 ANN METHOD

We choose TensorFlow[29,31] as the framework for ANN. Then we briefly introduce the multi-layer feed-forward neural network[30] used in this paper, which is most widely used in ANNs. A typical multilayer feed-forward network is shown in Fig. 3. It consists of three types of layers (one input layer, two hidden layers, and one output layer), each layer consists of one or more neurons. Feed-forward means that the information in the neural network flows only from input layer to output layer, instead of reverse (cyclic) propagation. Although it has been demonstrated in article[27] that a neural network containing two hidden layers and a differentiable function is sufficient to approximate any function, there is no definitive rule for determining the number of neurons required for each layer.

https://doi.org/10.1017/aer.2020.83 Published online by Cambridge University Press

Suppose that we know two vectors $X \in \mathbb{R}^{N^i}$ and $Y \in \mathbb{R}^{N^o}$, we seek to establish a mapping between the two vectors: $\emptyset$: $\mathbb{R}^{N^i} \mapsto \mathbb{R}^{N^o}$, namely

$$\tilde{Y} = \emptyset(X) \qquad \ldots (2)$$

where $\tilde{Y}$ represents the approximation of Y.

If we use the neural network in the Figure 3, the transmission of its internal information follows the following equations.

$$\begin{cases} Y_n^i = X_n, & n = 1, \cdots, N^i \\ Y_n^{h_1} = \theta^{h_1}\left(\sum_{m=1}^{N^i} W_{n,m}^{h_1} Y_m^i + B_n^{h_1}\right), & n = 1, \cdots, N^{h_1} \\ Y_n^{h_2} = \theta^{h_2}\left(\sum_{m=1}^{N^{h_1}} W_{n,m}^{h_2} Y_m^{h_1} + B_n^{h_2}\right), & n = 1, \cdots, N^{h_2} \\ \cdots & \cdots \\ Y_n^{h_k} = \theta^{h_k}\left(\sum_{m=1}^{N^{h_{k-1}}} W_{n,m}^{h_k} Y_m^{h_{k-1}} + B_n^{h_k}\right), & n = 1, \cdots, N^{h_k} \\ Y_n^o = Y_n^{h_k}, & n = 1, \cdots, N^o \end{cases}$$

Where X_n is the input, which called featue information (abbreviated as feature); $W_{n,m}^{h_i}, i = 1, \ldots, k$ are the weights of the output information of the m neurons among all neurons in the upper layer received by the n neurons in the first hidden layer to the kth hidden layer; $B_n^{h_i}, i = 1, \ldots, k$ are the offsets of information received by the neurons from the first hidden layer to the kth hidden layer; $N^i, N^{h_1}, N^{h_2}, \cdots, N^{h_k}, N^o$ are the number of neurons in the input layer, the first hidden layer, the second hidden layer, . . ., the kth hidden layer and the output layer; $Y_n^i, Y_n^{h_1}, \cdots, Y_n^{h_k}, Y_n^o$ are the information output by the input layer, the first hidden layer, the second hidden layer, . . . the kth hidden layer and the output layer, where Y_n^o is the information that needs to be predicted (we call them labels). Take $Y_2^{h_1}$ as an example, $Y_2^{h_1}$ represents the information output by the second neuron of the first hidden layer. $\theta^{h_i}, i = 1, \ldots, k$ is the activation functions of the first hidden layer to the kth hidden layer, which are the key factors for the ability of the ANN to approximate the nonlinear function. The selection of different activation functions can be found in the literature[32]. In this paper, we choose the relu activation function. What's more, the activation function of the two hidden layers is called the rectifying linear unit function[33].

$$\theta^{h_1}(a) = \theta^{h_2}(a) = \max(a, 0),\ \theta^{h_k}(a) = a$$

After defining the structure of the neural network, we use an optimisation method with the following second-order loss function to train the network.

$$C = \frac{1}{M^S N^\circ} \sum_{m=1}^{M^S} \sum_{n=1}^{N^\circ} (\tilde{Y}_n^m - Y_n^m)^2 + P$$

https://doi.org/10.1017/aer.2020.83 Published online by Cambridge University Press

Here M^s is the number of samples in the training set, and P is a regularisation term used to mitigate overfitting, whose definition as shown below

$$P = \gamma\left(\frac{1}{N^{h_1}N^i}\sum_{n=1}^{N^{h_1}}\sum_{m=1}^{N^i}\left(W_{n,m}^{h_1}\right)^2 + \frac{1}{N^{h_2}N^{h_1}}\sum_{n=1}^{N^{h_2}}\sum_{m=1}^{N^{h_1}}\left(W_{n,m}^{h_2}\right)^2 + \frac{1}{N^oN^{h_2}}\sum_{n=1}^{N^o}\sum_{m=1}^{N^{h_2}}\left(W_{n,m}^{o}\right)^2\right)$$

where γ is a positive coefficient, which is determined by the specific problem. After determining the specific weights and deviations, all the parameters of the ANN have been fully defined. The application process of the specific ANN will be elaborated in the next section.

4.0 THE FLOW FIELD RECONTRUCTION METHOD BASED ON ARRAY NEURAL NETWORK

4.1 Traditional flow field reconstruction method

In this paper, according to the flow field reconstruction method using artificial neural network[10], we describe our method process and the calculation process, which are described below.

At first, given the parameter set $\Xi = \left\{\xi^1, \cdots, \xi^p\right\}$, we have a corresponding set of high-precision samples corresponding to this set of parameters, which can be expressed as

$$A = [S_1 \mid \cdots \mid S_{M^s}] \in \mathbb{R}^{M_h \times M^s}$$

Here, S_{M^s} represents the flow field data solution vector of the M^s sample, M^s is the number of samples, and M_h is the dimension of the solution vector of each sample of the flow field data. Take the one-dimensional problem as an example, the parameters are

$$\Xi = \begin{bmatrix} Ma \\ Angle \end{bmatrix}$$

the solution vector is

$$S_{M^s} = \begin{bmatrix} \rho_{M^s} \\ u_{M^s} \\ p_{M^s} \end{bmatrix}$$

where $Ma, Angle$ represent Mach number and the angle of attack, respectively, $\rho_{M^s}, u_{M^s}, p_{M^s}$ are the solution vectors of density, velocity, pressure and so on. It should be noted that the number of parameters and solution vectors can be only one or not the first three (namely, one or more vectors may be removed, or one or more vectors may be added as needed). The parameters selected in this paper are Mach number and the angle of attack, and the solution vector is the pressure value.

In order to use the neural network for training, we use POD projection on flow field data matrix A. This is because all samples are generated by CFD code whose algorithm is finite volume method. The CFD grid used by the code is very large, while the dimension of the solution vector is proportional to the size of the grid. Therefore, the dimension of the sample is usually very large. If we use neural network training directly, each layer will need a lot of

https://doi.org/10.1017/aer.2020.83 Published online by Cambridge University Press

hidden neurons. Furthermore, if we want to determine the parameters of these neurons, we will need more samples to train, which will make the training process more difficult. Thus, we must carry out the POD process to simplify the problem. As mentioned above, POD includes eigenvalue decomposition and projection transformation, and A can be approximated as

$$\hat{A} = \widehat{U}\hat{A} = \sum_{i=1}^{k} \sigma_i u_i v_i^T \in \mathbb{R}^{M_h \times M^s},\ k \le r$$

where

$$\widehat{U} = [u_1 \mid \cdots \mid u_k] \in \mathbb{R}^{M_h \times k}$$

$$\hat{\Lambda} = [\sigma_1 v_1 \mid \cdots \mid \sigma_k v_k]^T \in \mathbb{R}^{k \times M^s}$$

Here k is the *podbasic* in the POD that we have mentioned before. It should be noted that $r \ll M_h$, thus $k \ll M_h$. Therefore, the dimension of the solution vector dimension is significantly reduced, which means that the amount of calculation is significantly reduced.

The next step is to build the ANN (namely, we need to determine the number of hidden layers and the number of neurons per layer). Unless otherwise stated, the ANN framework of ANN used in this article is as follows

(1) Set the number of neurons in the hidden layer to an intermediate value between the number of neurons of the input and output layers. In the practical application of the example in this paper, we gradually increase the number of hidden layer neurons and perform various training. Finally, we choose the number of neurons that can minimise the model error in the final model. It should be noted that the selection of the number of neurons in the hidden layer is not strictly limited, and the number of neurons defined by other options may also be a feasible solution.

(2) Set the number of hidden layers of the ANN to no more than three hidden layers. In the choice of hidden layers, for each case, we compare different number of hidden layers and choose the number of hidden layers which can minimise the model error. In the practical example, we determine the optimal number of neurons first for one, two, and three hidden layers based on the method of selecting the number of neurons in Step (1). Then, we select the number of hidden layers in the network which can minimise the error of the testing set as the number of hidden layers of the final model.

For the choice of optimiser in artificial neural networks, we use Adam (adaptive moment estimation) algorithm[34], which calculates the exponential moving average of the gradient and the square of the gradient, where the moving average represents the gradient estimated value of the first moment and the second moment which changes with time. Given the initial value (fixed to 10^{-4} in this paper), the Adam algorithm adaptively calculates the learning rate of each network parameter based on the estimated gradient values at different times during the training process. Adam is widely used, and it is not as effective as other stochastic optimisation methods. It is assumed that Z^n and $\widehat{Z}_n$ represent the n sample in the validation set and its corresponding predicted solution vectors. Then the squared error E between the predicted and exact values on the validation set is calculated as

$$E = \frac{1}{M^s N^o} \sum_{m=1}^{M^s} \sum_{n=1}^{N^o} \left(\hat{Z}_n^m - Z_n^m\right)^2$$

https://doi.org/10.1017/aer.2020.83 Published online by Cambridge University Press

In order to determine the time to terminate the training, we store the squared error of the continuous time step training in a series, where E_k represents the error after the kth training step. In all the examples given in this paper, we specify that if $E_k > E_{k-500}$, the training process will stop, which is in order to prevent over-fitting.

After constructing the ANN, we first introduce the traditional method for the convenience of comparison. The traditional method use classical artificial neural networks to train coefficient matrices, which only train the coefficient matrix $\hat{\Lambda}$.

$$\hat{\Lambda} = \widehat{U}^T A$$

The whole of $\hat{\Lambda}$ is used as the direct output of the neural network, but we find that the prediction of traditional method has a large error. Therefore, the neural network has been improved in this paper, we proposed the array neural network, which is used for training in this paper. The specific details are described in the next section.

The traditional method only establishes the following mapping through artificial neural network.

$$\hat{\Lambda} = \emptyset(\Xi)$$

that is

$$\hat{\Lambda}^{train} = \emptyset\left(\Xi^{train}\right)$$

$$\hat{\Lambda}^{validation} = \emptyset\left(\Xi^{validation}\right)$$

$$\hat{\Lambda}^{test} = \emptyset\left(\Xi^{test}\right)$$

where the superscript train, validation, and test respectively represent the parameters and the solution vectors after the POD of the corresponding training set, validation set and testing set. The output is the whole $\hat{\Lambda}$, which reflects that the traditional method doesn't consider the linear independence between modes and the fit of the neural network, which is one of the most causes of error. The detailed description of the traditional method can be found in the Ref. 10.

4.2 The improvement of traditional methods

The proposed method that we build the ANN is the same as the traditional method, we use the array neural network to train the coefficient matrix. As mentioned above, we only train the coefficient matrix $\hat{\Lambda}$.

$$\hat{\Lambda} = \widehat{U}^T A$$

The reason why it is called array neural network is that we have improved the neural network based on the traditional method. As mentioned above, when the whole $\hat{\Lambda}$ is used as the direct output of the neural network, we find that the prediction has a large error, so we adopt the array neural network for training in our paper. Array neural network means that the whole matrix $\hat{\Lambda}$ is not obtained directly, but we train the parameter set to obtain each row of the matrix $\hat{\Lambda}$, which equivalent to training each modal of the samples. This is because each row

https://doi.org/10.1017/aer.2020.83 Published online by Cambridge University Press

of the matrix $\hat{\Lambda}$ represents one modal (base) of all samples. What's more, the simultaneous training of all modals (base) will result in inaccurate prediction due to the difference in energy occupied by the modals. Therefore, we adopt the method which we predict after training by array neural network.

We consider establishing the following mapping through array neural network

$$\hat{\Lambda} = \emptyset(\Xi)$$

namely

$$\hat{\Lambda}^{train} = \emptyset\left(\Xi^{train}\right)$$

$$\hat{\Lambda}^{validation} = \emptyset\left(\Xi^{validation}\right)$$

$$\hat{\Lambda}^{test} = \emptyset\left(\Xi^{test}\right)$$

where the superscript train, validation, and test respectively still represent the parameters and the solution vectors after the POD of the corresponding training set, validation set and testing set, which can be written into the structure of an array neural network, that is

$$\begin{cases} \hat{\Lambda}_1 = \emptyset(\Xi) \\ \hat{\Lambda}_2 = \emptyset(\Xi) \\ \cdots \cdots \\ \hat{\Lambda}_{podbasic} = \emptyset(\Xi) \end{cases}$$

where $\hat{\Lambda}_i, i = 1, \cdots,$ *podbasic* represent each row of $\hat{\Lambda}$, the training set, the validation set are input into the artificial neural network through the manner above to obtain the trained network, i.e. the mapping $\emptyset$. Then we use the trained network $\emptyset$ to obtain the $\hat{\Lambda}$ of testing set through the known parameter set Ξ of the testing set.

After that we restore the $\hat{\Lambda}$ in testing set to A by inverse POD process, namely

$$A = \widehat{U}\hat{\Lambda} = \widehat{U}\widehat{U}^T A = A$$

where A is the testing set we obtained through training.

Finally, the predicted data of testing set is written back to the original high-precision data of the samples of the testing set, which is in order to achieve the purpose of flow field reconstruction.

In summary, the basic idea of the array neural network method in this article can be shown as follows:

Step 1: We use the POD method to decompose a set of flow fields into a set of basic modals, so that each flow field can be expressed by the basic modals and a weighted sum of coefficients. The basic modals are common to this group of flow fields, while the coefficients are unique to each flow field.

Step 2: We use these coefficients for neural network training. Each group of coefficients corresponds to a flow field and also corresponds to the features of this flow field. These features are the Mach number and angle of attack.

https://doi.org/10.1017/aer.2020.83 Published online by Cambridge University Press

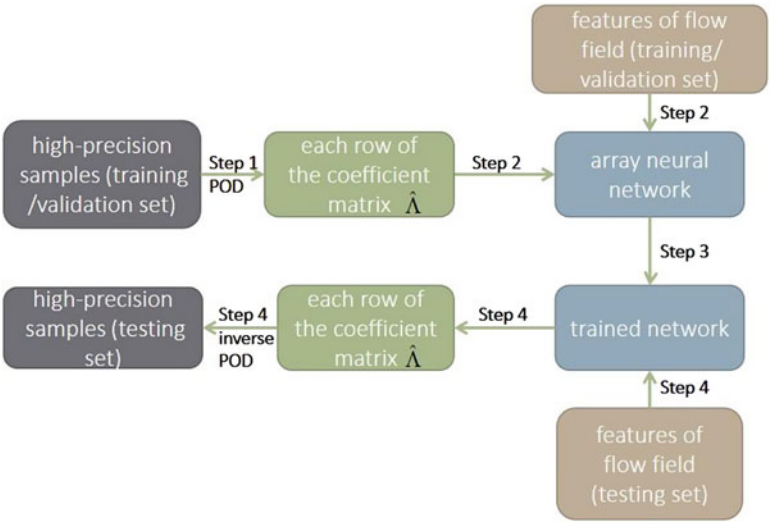


Figure 4. The flow chart of array neural network method.

Step 3: Through such training, we can get a trained network that can predict the modal coefficients at different Mach numbers and angles of attack.

Step 4: Through the trained network, we can predict the modal coefficients and the modals, and then we combine them. After that we can reconstruct or predict the flow field through the inverse POD process.

Figure 4 shows the process in detail.

5.0 RESULTS AND DISCUSSION

5.1 Results

In this section we demonstrate the flow field reconstruction under typical conditions, such as wing flow. Before drawing conclusions, it should be noted that it is necessary to incorporate as much knowledge as possible when we construct array neural networks. Because ANN is only used for the construction of unknown relationships, it should be supplemented by perfect theories or attributes. In order to reduce the number of samples required, the number of neurons of input layer and output layer should be minimised. In addition, due to the theoretical basis of ANN, the solution vectors of the ANN output must be determined by the input parameters. That is to say, the given samples must satisfy the mapping between the input and the output.

All examples in our paper were performed on the Intel CORE i5 7th Gen platform. The running time averages about 0.1 seconds for a training round. Because the training time varies with factors such as the number of neurons, training samples, and so on. The maximum number of training steps set by all the examples is 300,000 times. However, the difference between the samples and the rows of $\hat{\Lambda}$ may result in over-fitting. Thus the number of the stop training step is different. Therefore, not each branch network will reach the maximum number of training step.

https://doi.org/10.1017/aer.2020.83 Published online by Cambridge University Press

Table 1
Sample distribution

	Ma	*Angle*	Samples	Modals
Training set	0.3	[−10.0, 10.0]	88	10
Validation set	0.3	[−10.0, 10.0]	11	10
Test set	0.3	[−9.0, 9.0]	10	10

Table 2
The comparison of error between array neural network method and artificial neural network method

	Array neural network method (taking the first modal as an example)	Artificial neural network method
l2 error	0.0036778792	0.31334883
linear error	0.005562663	0.7747089

In this paper, we choose the flow field data of the NACA0012 airfoil[26] in the subsonic and wide range of angle of attack to study, which is under the calculation condition of Mach number 0.3, Reynolds number 6.98e6, and the range of angle of attack is −10°~10°, whose interval is 0.1°. Table 1 shows the distribution of samples. All examples are the airfoil flow fields we described above. The dimension of all the original flow field data is 7564, which is same as the size of the grid. The solution vector selected by all the examples is the pressure, and the features of flow field are the Mach number and the angle of attack[35,36].

Figure 5 shows the relative error of the array neural network method and the traditional neural network method on the same testing set. The abscissa represents the serial number of the testing samples, and the ordinate represents the relative error between the high-precision samples and the high-dimensional prediction samples. This figure reflects the relative error of each sample between the traditional method and the array neural network method.

As can be seen from this figure, the relative error of the array neural network method in this paper is relatively stable, and it doesn't change when the angle of attack changes, which shows that our method has good universality for samples with different conditions. The reason why the relative errors of the two methods are not so different is that this figure reflects the relative error of the entire flow field, instead of the relative error at one point of the flow field.

The error in Table 2 is the comparison between modal coefficients of the array neural network and the traditional method. It can be seen that the l2 errors and linearity error of the array neural network method are significantly lower than the artificial neural network method, and the degree of reduction is greater than two orders of magnitude.

Figures 6 and 7 show the convergence rate and order of the training error and the cost error (the error of the loss function) of the array neural network (the number of modal is ten) and the traditional method. For the convenience of observation, we take the logarithm of base 10. The abscissa represents the number of iterations, and the ordinate represents the training error and the cost error. Figure 6 reflects the trend of training error with the number of iterations. Figure 7 reflects the trend of cost error with the number of iterations.

https://doi.org/10.1017/aer.2020.83 Published online by Cambridge University Press

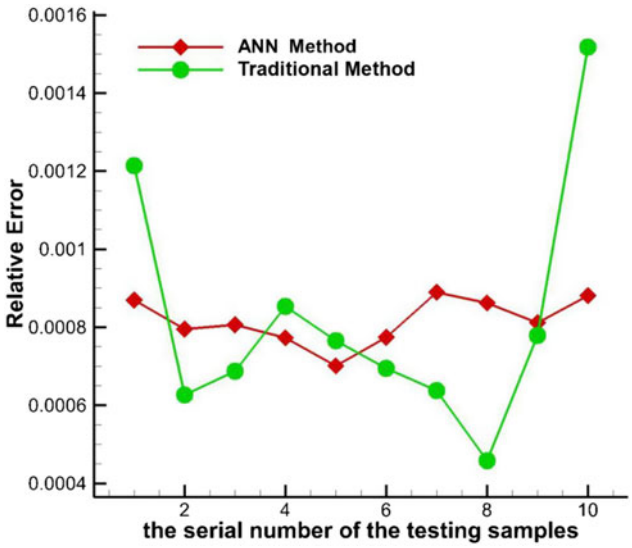


Figure 5. The comparison of relative error of flow field reconstruction between the array neural network and the traditional method.

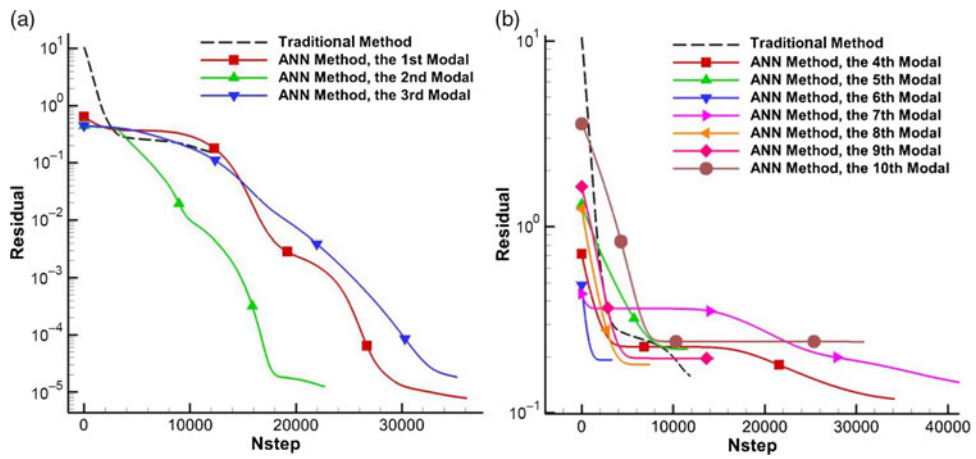


Figure 6. Training error of the array neural network method and tradition method. (a) modal 1-modal 3 (b) modal 4-modal 10.

We have drawn the first three modals and the other modals separately. It can be found that the convergence of the first three modals is very good, and their error can converge to the power of -4 or even -5 of 10, while the traditional method only converges to the power of -1. That is to say, the training error and loss function error of the array neural network are significantly lower than the traditional neural network method. The reason why the phenomenon in the figure appears is that the energy of the first three modals accounts for a large percentage of the overall energy. Actually, each modal represents different information of the flow field. The larger the percentage of the energy the modal has, the more information it represents. The convergence error of the first three modals is very small, which also explains

https://doi.org/10.1017/aer.2020.83 Published online by Cambridge University Press

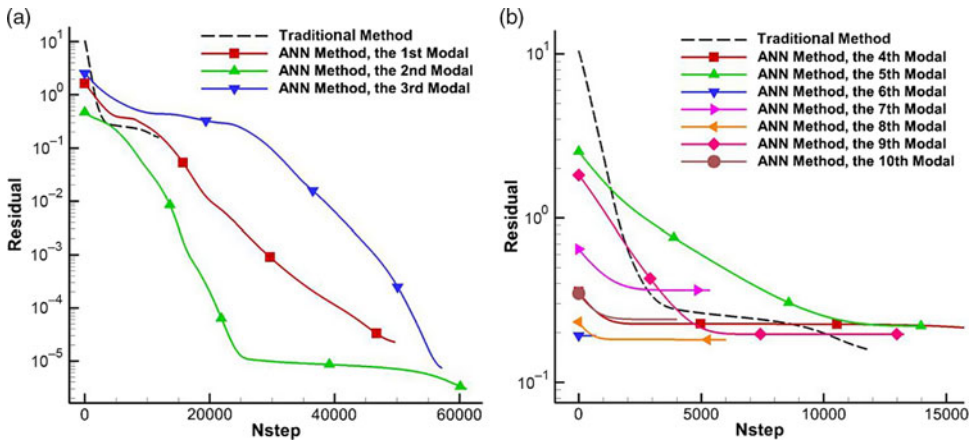


Figure 7. Loss function error of array neural network method and tradition method. (a) modal 1-modal 3 (b) modal 4-modal 10.

that the array neural network method can reconstruct the flow field well. Therefore, the first few modals of the array neural network have almost determined the convergence rate and convergence order of the error of the whole array neural network, which is one of the reasons why we choose the first few modals instead of all the modals. Another reason will be analysed in detail in the energy analysis subsection of this section.

It can also be seen from Figs 6 and 7 that the error of the traditional method converges in less than 20,000 steps although it is less than that of our method, which means that the traditional method converges quickly. This is because the array neural network method is equivalent to training the information contained in each modal of the flow field, and the neural network has different learning efficiency for each information. However, the traditional method is equivalent to training the information in the flow field together, which is equivalent to mixing the information of all the flow field samples. Since the neural network has different learning efficiency of mixed information and single information, it is reasonable that the traditional method converges faster than the array neural network method in our paper. In addition, in order to prevent overfitting, when the error after iteration 500 steps is larger than the current error, the calculation will automatically stop. This is why the traditional method and the array neural network method have different training convergence iteration steps.

Some of the result of flowfield reconstruction by array neural network method are shown in the Fig. 8. It can be seen that the contour of the reconstructed flow field is smooth, and the flow field structure is correct, which are in agreement with the real flow field (CFD flow field) well.

5.2 The analysis of flow field modal energy

As described in the second section of this paper, the original flow field data matrix $A = [S_1 | \cdots | S_{M^s}] \in \mathbb{R}^{M_h \times M^s}$ has been decomposed into the product of three matrices U, Σ and V^T after SVD, where U and V are orthogonal matrices, and Σ is an eigenvalue matrix. It can be known from the basic knowledge of advanced algebra that the result of multiplication of the orthogonal matrix and the diagonal matrix is still an orthogonal matrix. Therefore, the matrix $\hat{\Lambda}$ obtained by POD is also an orthogonal matrix. What's more, the orthogonal vector group must be linearly independent, so each row of the orthogonal matrix $\hat{\Lambda}$ must be linearly

https://doi.org/10.1017/aer.2020.83 Published online by Cambridge University Press

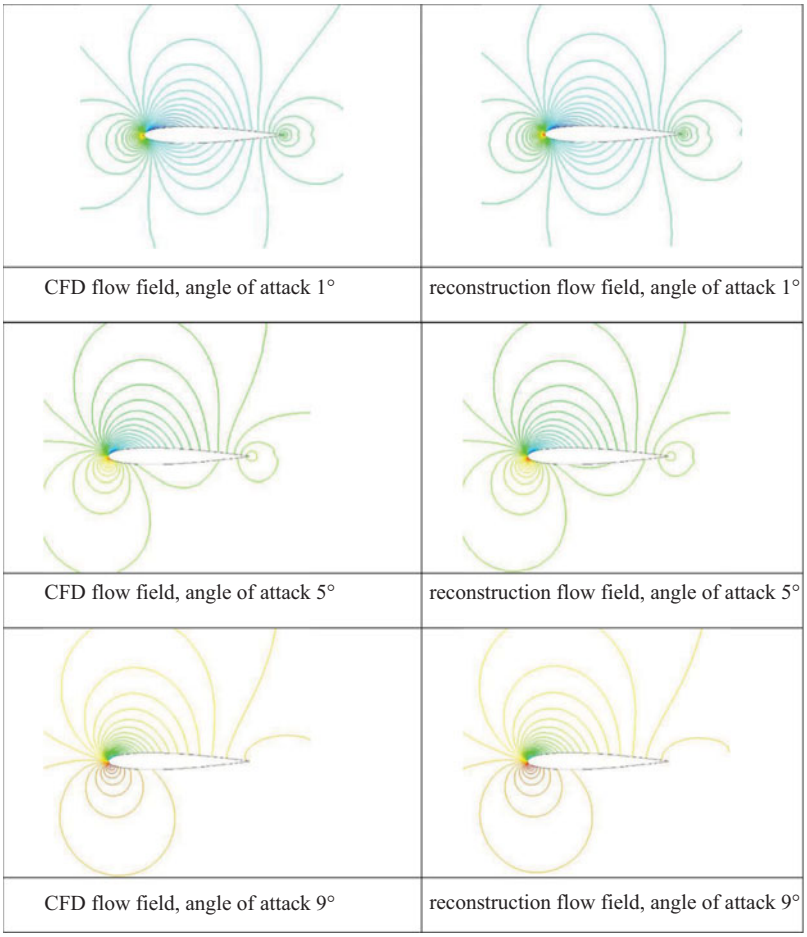


Figure 8. The comparison between the effects of different angles of attack in CFD flow field and reconstructed flow field, pressure contour, Mach number 0.3.

independent. According to the linear independence of $\hat{\Lambda}$ and the energy analysis above, the flow field information and energy contained in each row of $\hat{\Lambda}$ are different, which is also not related to each other. Therefore, using a single artificial neural network to predict all the lines of $\hat{\Lambda}$ is not appropriate. This is also the reason why we choose BP neural network algorithm in artificial neural network instead of circulation or convolution neural network algorithm.

Σ is a diagonal matrix, and the eigenvalues on the diagonal are distributed from large to small. The essence of POD is to find a set of optimal standard orthogonal modals φ of the flow field data matrix A, so that the projection of the elements of the flow field data information $\{S_t\}|_{t=1}^{M^s}$ on this group of modals can be the largest[8], namely

$$\begin{cases} \max\limits_{\varphi} \dfrac{1}{M^s} \sum\limits_{t=1}^{M^s} |(S_t, \varphi_i)|^2 \\ (\varphi, \varphi) = 1 \end{cases}$$

https://doi.org/10.1017/aer.2020.83 Published online by Cambridge University Press

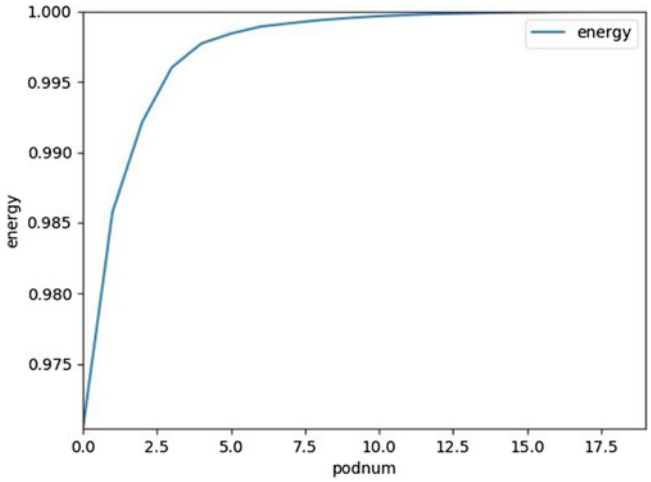


Figure 9. The proportion of energy in modal quantity.

where $(*, *)$ represents the inner product operation. Theoretically any inner product form can be used. However, in terms of the physical meaning of our paper, the inner product must reflect the physical region solved by the governing equation. In this paper, we use the inner product defined on the square integrable space (L2 space), and $|*|$ is the norm induced by the inner product.

From the perspective of flow field energy, the larger the eigenvalue, the larger the flow field energy. That is to say, the flow field contains more information of data matrix[7]. Thus, we must maximise the generalised "energy" function $\Upsilon(podbasic)$ when we ensure that the amount of computation is as small as possible. Figure 9 shows the trend of Υ increasing with the number of modals.

It can be seen from the Fig. 9 that when the number of modals reaches about 10, the energy ratio has been already higher than 99.95%. Therefore, selecting a larger number of modals can't significantly improve the accuracy of the reconstructed flow field. Instead, it will increase the calculation amount and calculation time; so we choose ten as the number of modals in this paper, which is the second reason for selecting the first few modals.

In summary, this paper adopts array neural network to predict each row of $\hat{\Lambda}$, the equivalent of which is that each row of $\hat{\Lambda}$ corresponds to a separate neural network. Thus, we can maximise the use of the flow field information contained in each modal, which also meets the conclusion above that there is no correlation between the information of different modals. Thus, we can reconstruct the flow field more comprehensively.

6.0 CONCLUSION

In this paper, we improve the traditional method based on the artificial neural network. A flow field reconstruction method based on array neural network is proposed. Compared to traditional methods, array neural network method has the faster convergence speed and smaller error. What's more, we have fully considered the linear independence between modals and the fit of the neural network. We choose each row of the matrix Λ as the output of the array neural network method, which significantly reduces the output of the traditional method. This is also one of the most important reasons for the significant increase in accuracy.

https://doi.org/10.1017/aer.2020.83 Published online by Cambridge University Press

In order to test the proposed flow field reconstruction method based on array neural network, a typical testing set is carried out, mainly for airfoil flow field. In general, the proposed method can reconstruct the flow field better and has wide range of application, which can significantly improve the accuracy compared with the traditional method. For the three hidden layers in array neural network, no obvious jumps and huge errors are found in results. In addition, the solution vectors of each sample can be a single component or a combination of multiple field components because the array neural network is nonlinear. We select the single pressure field in this paper, but in fact it is theoretically feasible to output both the pressure field and the velocity field. For the POD method adopted, the calculation time and space loss are significantly reduced with little loss of accuracy, which is obviously superior to the the case of no dimension reduction and all modals are retained.

However, for a typical flow structure, it can be found that when the angles of attack and the Mach numbers of the flow fields are too high, fully considering the entire parameter space is necessary, which needs to increase the number of the input parameters and a large amount of sample data. In this case, the effect of field reconstruction will reduce. This is because the model in this paper is sensitive to parameter changes and the array neural network is data dependent. In a word, the work of this paper is mainly to study the flow field with small angle of attack with good characteristics and the sample set with stable flow field structure. By improving the traditional method, this method has good prediction accuracy for the flow field under such conditions.

Our method does not perform as expected if the flow fields have very different structures. Especially in some areas of the flow field, there will be a phenomenon that the local flow gradient is large, the change is dramatic, and the position is not fixed. That is to say, the effect of field reconstruction will reduce when the flow field has high Mach number and high angle of attack. A flow field with a large angle of attack or a wider speed range will contain more abundant flow field characteristics. Such as shock waves, separation, etc. This makes the energy of the fundamental modes decomposed by the POD method not concentrated, at least the first few modals are not significant. So we may need to improve some aspects, such as increasing sample parameters, improving the depth and complexity of the neural network, considering changing the structure of the array neural network, combining machine learning with other related algorithms and so on. In the future, our work also hopes to deeply study the generalization ability of the method in this paper.

ACKNOWLEDGEMENTS

This study was co-supported by the National Key R&D Program of China (No.2017YFB0202803), the Strategic Priority Research Program (B) of the Chinese Academy of Sciences (No.XDB22020102) and the e-Science Foundation of Chinese Academy of Sciences (No.XXH13506-204).

REFERENCES

1. LORENTE, L., VEGA, J., and VELAZQUEZ, A. Generation of aerodynamic databases using high-order singular value decomposition, *J. Aircr.*, 2008, **45**, (5), pp 1779–1788. doi: 10.2514/1.35258
2. HESTHAVEN, J.S., ROZZA, G., and STAMM, B., *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, Springer, 2016. doi: 10.1007/978-3-319-22470-1

https://doi.org/10.1017/aer.2020.83 Published online by Cambridge University Press

3. LIANG, Y., LEE, H., LIM, S., LIN, W., LEE, K., and WU, C. Proper orthogonal decomposition and its applications—Part 1: theory, *J. Sound Vibr.*, 2002, **252**, (3), pp 527–544. doi: 10.1006/jsvi.2001.4041
4. HESTHAVEN, J.S., STAMM, B., and ZHANG, S. Efficient greedy algorithms for high-dimensional parameter spaces with applications to empirical interpolation and reduced basis methods, *ESAIM-Math. Model. Num.*, 2014, **48**, (1), pp 259–283. doi: 10.1051/m2an/2013100
5. SHAH, A., XING, W. and TRIANTAFYLLIDIS, V. Reduced-order modelling of parameter-dependent, linear and nonlinear dynamic partial differential equation models, *Proc. Royal Soc. A*, 2017, **473**, (2200), Paper 20160809. doi: 10.1098/rspa.2016.0809
6. BENNER, P., GUGERCIN, S. and WILLCOX, K. A survey of projection-based model reduction methods for parametric dynamical systems, *SIAM Rev.*, 2015, **57**, (4), pp 483–531. doi: 10.1137/130932715
7. JIE, L., YANHUI, D. and JINSHENG, C. A quick method of flow field prediction based on proper orthogonal decomposition. *Adv. Aeronaut. Eng.*, 2014, **5**, (3), pp 350–357.
8. YASONG, Q., JUNQIANG, B. and JUN, H. Flow field estimation method based on proper orthogonal decomposition and surrogate model. *Hangkong Xuebao/Acta Aeronautica et Astronautica Sinica*, 2013, **34**, (6), pp 1249–1260.
9. ZHI, H., PENG-YIN, L., XIAO-CHENG, Z. and ZHAO-HUI, D. DMD analysis of airfoil's flow field under different degrees of dynamic stall. *J. Eng. Thermal Energ. Power*, 2017, **32**, (7), pp 52–57.
10. YU, J. and HESTHAVEN, J.S. Flowfield reconstruction method using artificial neural network. *AIAA J.*, 2019, **57**, (2), pp 482–498.
11. IUNGO, G.V., SANTONI-ORTIZ, C., ABKAR, M., PORTÉ-AGEL, F., ROTEA, M.A. and LEONARDI, S. Data-driven reduced order model for prediction of wind turbine wakes, *J. Phy. Conf. Ser.*, 2015, **625**, Paper 012009. doi: 10.1088/1742-6596/625/1/012009
12. PEHERSTORFER, B. and WILLCOX, K. Dynamic data-driven reduced-order models, *Comput. Methods Appl. Mech. Eng.*, 2015, **291**, pp 21–41. doi: 10.1016/j.cma.2015.03.018
13. TRACEY, B.D., DURAISAMY, K. and ALONSO, J.J. A Machine Learning Strategy to Assist Turbulence Model Development, 53rd AIAA Aerospace Sciences Meeting, AIAA Paper 2015-1287, 2015. doi: 10.2514/6.2015-1287
14. XIAO, H.,WU, J.-L.,WANG, J.-X., SUN, R. and ROY, C. Quantifying and reducing model-form uncertainties in Reynolds-averaged Navier–Stokes simulations: a data-driven, physics-informed Bayesian approach, *J. Comput. Phy.*, 2016, **324**, pp 115–136. doi: 10.1016/j.jcp.2016.07.038
15. PARISH, E.J. and DURAISAMY, K. A paradigm for data-driven predictive modeling using field inversion and machine learning, *J. Comput. Phy.*, 2016, **305**, pp 758–774. doi: 10.1016/j.jcp.2015.11.012
16. LING, J., KURZAWSKI, A. and TEMPLETON, J. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, *J. Fluid Mech.*, 2016, **807**, pp 155–166. doi: 10.1017/jfm.2016.615
17. DURAISAMY, K., ZHANG, Z.J. and SINGH, A.P. New Approaches in Turbulence and Transition Modeling Using Data-Driven Techniques, 53rd AIAA Aerospace Sciences Meeting, AIAA Paper 2015-1284, 2015. doi: 10.2514/6.2015-1284
18. LEE, C., KIM, J., BABCOCK, D. and GOODMAN, R. Application of neural networks to turbulence control for drag reduction, *Phy. Fluids*, 1997, **9**, (6), pp 1740–1747. doi: 10.1063/1.869290
19. KURTULUS, D.F. Ability to forecast unsteady aerodynamic forces of flapping airfoils by artificial neural network, *Neural Comput. Appl.*, 2009, **18**, (4), p 359. doi: 10.1007/s00521-008-0186-2
20. BUI-THANH, T., DAMODARAN, M. and WILLCOX, K. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition, *AIAA J.*, 2004, **42**, (8), pp 1505–1516. doi: 10.2514/1.2159
21. EVERSON, R. and SIROVICH, L. Karhunen–Loeve procedure for Gappy data, *J. Opt. Soc. Am.*, 1995, **12**, (8), pp 1657–1664. doi: 10.1364/JOSAA.12.001657
22. WILLCOX, K. Unsteady flow sensing and estimation via the Gappy proper orthogonal decomposition, *Comput. Fluids*, 2006, **35**, (2), pp 208–226. doi: 10.1016/j.compfluid.2004.11.006
23. TOSUN, E., AYDIN, K. and BILGILI, M. Comparison of linear regression and artificial neural network model of a diesel engine fueled with biodiesel-alcohol mixtures, *Alexandria Eng. J.*, 2016, **55**, (4), pp 3081–3089. doi: 10.1016/j.aej.2016.08.011
24. CYBENKO, G. Continuous Valued Neural Networks with Two Hidden Layers are Sufficient, TR, 1988.

https://doi.org/10.1017/aer.2020.83 Published online by Cambridge University Press

25. CYBENKO, G. Approximation by superpositions of a sigmoidal function, *Math. Control Signal Syst.*, 1989, **2**, (4), pp 303–314. doi: 10.1007/BF02551274
26. SHOJAEEFARD, M.H., KHORAMPANAHI, A. and MIRZAEI, M. Numerical investigation of oscillation frequency and amplitude effects on the hydrodynamic coefficients of a body with NACA0012 hydrofoil section. *J. Mech. Sci. Tech.*, 2017, **31**, (5), pp 2251–2260.
27. TAIRA, K., BRUNTON, S.L., DAWSON, S.T., ROWLEY, C.W., COLONIUS, T., MCKEON, B.J., SCHMIDT, O.T., GORDEYEV, S., THEOFILIS, V. and UKEILEY, L.S. Modal analysis of fluid flows: an overview, *AIAA J.*, 2017, **55**, (12), pp 4013–4041. doi: 10.2514/1.J056060
28. ROWLEY, C.W. and DAWSON, S.T. Model reduction for flow analysis and control, *Ann. Rev. Fluid Mech.*, 2017, **49**, pp. 387–417. doi: 10.1146/annurev-fluid-010816-060042
29. ABADI, M., BARHAM, P., CHEN, J., CHEN, Z., DAVIS, A., DEAN, J., DEVIN, M., GHEMAWAT, S., IRVING, G., ISARD, M., KUDLUR, M., LEVENBERG, J., MONGA, R., MOORE, S., MURRAY, D.G., STEINER, B., TUCKER, P., VASUDEVAN, V., WARDEN, P., WICKE, M., YU, Y. and ZHENG, X. Tensor flow: a system for large-scale machine learning, *OSDI*, 2016, **16**, pp 265–283.
30. KARLIK, B. and OLGAC, A.V. Performance analysis of various activation functions in generalized MLP architectures of neural networks, *Int. J. Artif. Intell. Expert Syst.*, 2011, **1**, (4), pp 111–122.
31. GLOROT, X., BORDES, A. and BENGIO, Y. Deep Sparse Rectifier Neural Networks, Proceedings of the 14th International Conf. on Artificial Intelligence and Statistics, 2011, pp 315–323.
32. KINGMA, D. and BA, J. Adam:AMethod for Stochastic Optimization, arXiv preprint arXiv:1412.6980, 2014.
33. ECONOMON, T.D., PALACIOS, F., COPELAND, S.R., LUKACZYK, T.W. and ALONSO, J.J. SU2: an open-source suite for multiphysics simulation and design, *AIAA J.*, 2016, **54**, (3), pp. 828–846. doi: 10.2514/1.J053813
34. ALONSO, D., VEGA, J. and VELAZQUEZ, A. Reduced-order model for viscous aerodynamic flow past an airfoil, *AIAA J.*, 2010, **48**, (9), pp 1946–1958. doi: 10.2514/1.J050153
35. WATANABE, S., ISHIMOTO, S. and YAMAMOTO, Y. Aerodynamic characteristics evaluation of hypersonic flight experiment vehicle based on flight data, *J Spacecraft Rockets*, 1997, **34**, (4), pp 464–470. doi: 10.2514/2.3259
36. KURGANOV, A. and TADMOR, E. Solution of two-dimensional Riemann problems for gas dynamics without Riemann problem solvers, *Numer. Methods Part. Different. Equations*, 2002, **18**, (5), pp 584–608. doi: 10.1002/(ISSN)1098-2426
37. HANSEN, P.C. The truncated SVD as a method for regularization, *Bit Numer. Math.*, 1987, **27**, (4), pp 534–553.
38. SHNAYDERMAN, A., GUSEV, A. and ESKICIOGLU, A.M. An SVD-based grayscale image quality measure for local and global assessment, *IEEE Trans. Image Process.*, 2006, **15**, (2), pp 422–429.
39. DRMA, Z and VESELI, K. New fast and accurate Jacobi SVD algorithm. *SIAM J. Matrix Anal. Appl.*, 2008, **29**, (4), pp 1322–1342.
40. BERKOOZ, G., HOLMES, P. and LUMLEY, J.L. The proper orthogonal decomposition in the analysis of turbulent flows. *Ann. Rev. Fluid Mech.*, 1993, **25**, (1), pp 539–575.
41. ROWLEY, C.W. Model reduction for fluids, using balanced proper orthogonal decomposition. *Int. J. Bifurcat. Chaos*, 2011, **15**, (3), pp 997–1013.
42. VOLKWEIN, K.K. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J Numer. Anal.*, 2003, **40**, (2), pp 492–515.
43. LIANG, Y.C., LEE, H.P. and LIM, S.P., et al. Proper orthogonal decomposition and its applications—Part I: Theory. *J. Sound Vibr.*, 2002, **252**, (3), pp 527–544.

https://doi.org/10.1017/aer.2020.83 Published online by Cambridge University Press