Artificial Intelligence Technologies and Applications
C. Chen (Ed.)
© 2024 The Authors.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).
doi:10.3233/FAIA231352

Development of Reduced-Order Models Based on Singular Value Decomposition and Neural Network Regression

Qi TANG, Yujie SUN, Yuhan ZHU, Lei KANG, Changli LIU[1]
School of Mechanical and Power Engineering, East China University of Science and Technology, Shanghai 200237, China

Abstract. This article proposes a method for building reduced-order models (ROMs) to address the issue of long solving time in traditional ANSYS finite element models, which can provide fast simulation results based on different loading conditions. Firstly, an orthogonal experiment is designed to obtain simulation results from the ANSYS model as training data. Secondly, the singular value decomposition (SVD) is combined with neural network regression to establish the reduced-order model. Then, the ROM generator software is built based on MATLAB. Finally, a stress reduced-order model of a specific finite element model is constructed, and the accuracy and efficiency of the proposed method are verified by comparing the simulation results with the results calculated by the reduced-order model, which can improve the solving efficiency while ensuring the accuracy of the solution.

Keywords. Reduced-order model, singular value decomposition, neural network regression

1. Introduction

In recent years, the development of digital twin technology has placed higher demands on the computational speed of models. Some complex finite element models have large datasets, leading to long computational cycles [1]. In this context, this article focuses on developing reduced-order models (ROMs) [2] by refining full-order finite element models into low-order models. ROMs can describe the system with quantitative accuracy and achieve almost real-time computing effects at a much lower computational cost than numerical simulations.

Singular value decomposition (SVD) [3] is a widely used algorithm in reduced-order models. The principle of SVD is to use linear transformation to find the main components of high-dimensional vectors in the data and project them onto a low-dimensional vector space to reduce the dimensionality of the data. This set of feature vectors retains the main features of the original high-order vector and can be used to reconstruct the original high-order vector. Based on the principle of SVD, this article proposes a fast method for building reduced-order models, significantly improving computational efficiency.

Neural networks [4] can implement complex nonlinear mapping and theoretically fit any curve. The MATLAB neural network regression model [5] provides a

[1] Corresponding author: Changli LIU, School of Mechanical and Power Engineering, East China University of Science and Technology, Shanghai 200237, China; email: clliu@ecust.edu.cn

backpropagation (BP) [6] neural network that is trained by error backpropagation. It can easily create, train, evaluate, predict, and apply neural network models, learning and storing many input-output pattern mapping relationships without revealing a mathematical equation that describes this mapping relationship beforehand [7].

This paper studies the construction of reduced-order models (ROMs) that can quickly compute results based on working conditions using the principles of SVD decomposition and neural network regression. A universal ROM generator is developed based on MATLAB. Finally, an example of a stress field reduction model of a finite element model is constructed.

2. Acquiring Training Data for Reduced-order Model

This chapter introduces obtaining training data through the established finite element model. In this paper, the training data mainly refers to two parts, input data, and output data. The input data refers to working conditions like load, boundary, and contact conditions. The output data is the result data at each node, such as stress, strain, displacement, etc.

2.1 Orthogonal Experimental Design

The purpose of conducting orthogonal experimental design is to obtain equivalent results of a comprehensive large number of simulations with a minimum number of simulations, approximating the original finite element full-order model [8]. This paper uses the CCD sampling method in Latin hypercube design to design orthogonal experiments for a simple finite element model with 20 loads and temperatures, as shown in Table 1.

2.2 Training Data Set Construction

According to the designed 20 sets of orthogonal experiments, opposite loads were applied to the two loading holes of the model, as shown in Figure 1, and the temperature was applied to the whole model for numerical simulation. The stresses under 20 working conditions were obtained, as shown in Table 2. 20 working conditions were the input data. The stresses of all nodes under 20 sets of working conditions were the output data, collectively called the training data, which were later used to train the model to obtain the mapping relationship between working conditions and stresses.

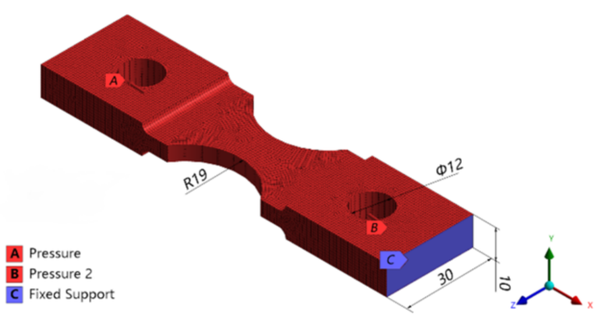

Figure 1. Finite element model loading method

Table 1. Work conditions for orthogonal test design

Solutions	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	……	c20
Force (MPa)	135	405	525	15	285	225	345	255	555	75	……	585
Temperature (℃)	531.25	581.25	406.25	556.25	431.25	643.75	543.75	631.25	481.25	468.75	……	418.75

Table 2. Stress obtained from finite element simulation

Solutions \ Nodes	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	……	c20
n1	55.329	88.227	44.699	50.98	81.566	41.134	46.827	75.239	37.825	42.921	……	239.79
n2	166	264.65	134.14	152.96	244.67	123.45	140.5	225.7	113.52	128.78	……	382.51
……	……	……	……	……	……	……	……	……	……	……	……	……
n234422	14.91	44.502	59.06	1.6524	31.68	24.55	38.054	27.863	63.765	8.3222	……	69.247

3. Order Reduction of Result Data

This chapter introduces how to lightweight the result data. Due to the fine mesh division of the finite element model and a large number of data points, it is cumbersome to directly establish the mapping relationship between working conditions and results, which cannot achieve real-time calculation. To establish a model for fast calculation, it is necessary to reduce the order of the result data and extract the main information of the result data.

3.1 Eigenvalue Decomposition and Order Reduction of Field Data

SVD decomposes the result data to obtain eigenvalues and eigenvectors. SVD can help us find the main components of the data by linear transformation and map high-dimensional data to low-dimensional space, thereby reducing the dimensionality of the data. As shown in Figure 1, assuming an m×n matrix A composed of result vectors (original data), it can be decomposed into U, S, and V matrices by SVD, also known as singular value decomposition. S is the singular value matrix, a diagonal matrix composed of the singular values of A. U is the left singular matrix, V is the right singular matrix, and they are composed of the left/right singular vectors of A, respectively [9]. Both U and V are unitary matrices [10]. In many cases, the sum of the first 10% or even 1% of singular values accounts for more than 99% of the total sum of singular values. We can also approximate the matrix using the first k singular values and corresponding left and right singular vectors. In this way, A is approximately decomposed into an eigenvector matrix U and a coefficient matrix C [11], as shown in Equation (1) and Figure 2.

$$A_{m\times n} = U_{m\times n} S_{n\times n} V_{n\times n}^{T} \approx U_{m\times k} S_{k\times k} V_{k\times n}^{T} = U_{m\times k} C_{k\times n} \tag{1}$$

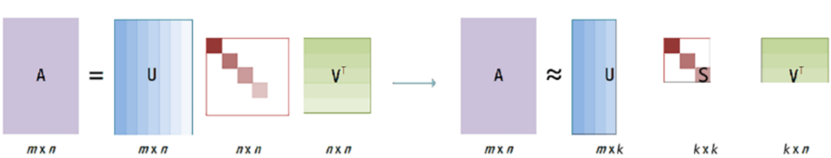

Figure 2. SVD Decomposition and Truncation of the First k-order Feature Vector Matrix

3.2 Reduced-order Error Analysis

Root Mean Square Error (RMSE), or Standard Error, is a commonly used indicator to measure the difference between actual and predicted values. It is the deviation between the predicted and actual values in a finite number of measurement times i=1, 2, 3, ...n [12]. RMSE is commonly expressed in the following Equation (2).

$$\text{RMSE}=\sqrt{\Sigma(\hat{y}_i-y_i)/n} \tag{2}$$

where n is the number of measurements, that is, the number of samples; y_i is the true value, $\hat{y}_i$ is the predicted value, and Σ denotes summing over all samples. ($\hat{y}_i$-y_i) is the deviation of each set of predicted values from the true values.

The smaller the value of RMSE, the better the model's predictive performance. Moreover, RMSE can consider both the deviation and the dispersion between the predicted and true values, so it is commonly used to measure the prediction accuracy of a model.

Using this method, the RMSE of $U_{m\times k}C_{k\times n}$ and $A_{m\times n}$ at each order can be calculated, which helps us find the optimal order for dimension reduction that meets high accuracy and fast computation requirements.

4. Prediction of Result Data

This chapter introduces how to establish the relationship between working conditions and feature vectors. After the dimension reduction in Chapter 3, the result matrix A has been simplified into a matrix of feature vectors and a coefficient matrix. Therefore, the problem of establishing the mapping relationship between working conditions and results has changed to establish the mapping relationship between working conditions and feature vectors.

4.1 Training Neural Network for Predicting Feature Vectors

Firstly, the structure of the neural network is defined, including the number of nodes, layers, activation functions, etc., for the input layer, hidden layer, and output layer. Then, a neural network object is created using a neural network configurator.

The data were divided into training, validation, and testing sets. The training set was used to train the data. The validation set was used to check if the network was generalizing and to stop training before overfitting. The testing set was used to test the network's generalization independently. Then, a training method was chosen, and the model training was initiated. The predictor variables were mapped to continuous responses by training the neural network.

We feed all the operating conditions from the orthogonal experimental design into the trained network for feature vector prediction. Then we multiply the resulting vectors by the matrix $C_{k\times n}$ to obtain the predicted result data.

4.2 Prediction Error Analysis

According to the root mean square error (RMSE) calculation method in Section 3.3, the error between the predicted results calculated by the prediction model and those calculated by the finite element model can be calculated for each set of operating conditions. This error includes the k-order reduction error from the previous chapter and the prediction error from this chapter.

5. Reduced-order Model Generator

The application platform of this system is developed based on MATLAB App Designer (R2022a) platform. Using MATLAB and the SVD principle and neural network regression technology described in the previous chapters, an application interface capable of generating reduced-order models is built, consisting of four parts:

Check module: It imports training data, validates the consistency of the project data and displays information on the interface.

Build module: It generates the reduced-order model and calculates and displays the reduced-order error under different orders.

Validate module: It analyzes the error between the reduced-order and full-order models, displays the cloud map under different working conditions, and the specific values of each data point.

Evaluate module: It exports and saves the reduced order model and displays the result cloud map based on the input working condition parameters.

This chapter demonstrates the entire operation of the software by generating a reduced-order model of the stress field of a simple finite element model as an example.

5.1 Check Module

The check module imports the training data required to generate the reduced-order model and the mesh information related to the image display. The training data includes input and output data. In this example, the input data refers to the operating conditions, mainly two parameters, load, and temperature. The output data is the stress field under each operating condition. The mesh information includes element information and node information. The element information contains the element type and the node numbers corresponding to each element, and the node information refers to the node coordinates, as shown in Figure 3.

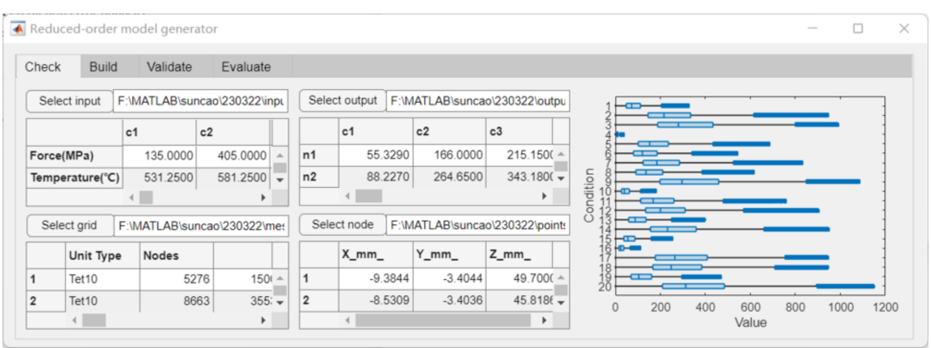

Figure 3. Input and output data and grid information

5.2 Build Module

In the Build module, the result matrix is subjected to SVD decomposition. Based on the root mean square error curve and the specific root mean square errors displayed in the table for different orders and singular values, an appropriate reduction order is selected to reduce the original result matrix. The corresponding U, S, and V matrices are truncated to the selected order. A MATLAB neural network regression model establishes the relationship between the input (operating conditions) and output (feature vectors), resulting in a reduced-order model, as shown in Figure 4.

For this example, the root mean square error curve stabilizes after the ninth-order model. Increasing the order further does not significantly improve accuracy but greatly increases computational complexity.

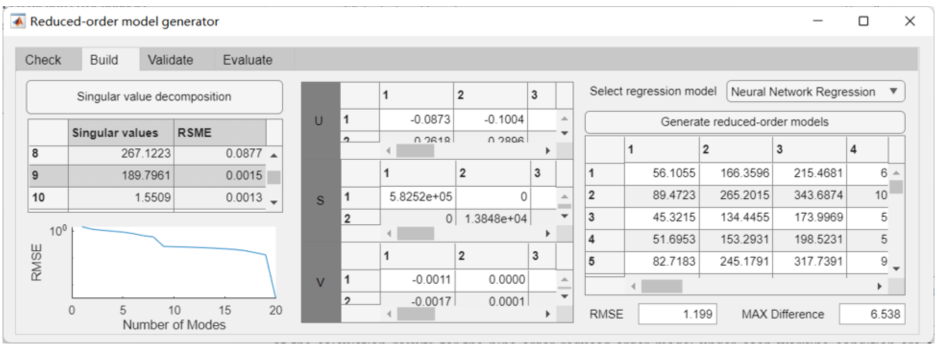

Figure 4. SVD decomposes the compressed field data as well as generates reduced-order models

5.3 Validate Module

In the Validate module, different working conditions can be selected to view the result contour plots and specific values of all nodes for both the full-order and reduced-order models and to compare the errors between them under different working conditions. The interface also displays the root-mean-square errors of the calculation results for the full-order and reduced-order models under each working condition, as shown in Figure 5. For the model presented in this paper, the errors of the calculation results for the nine-order reduced-order model under each working condition are acceptable, which verifies the feasibility and accuracy of the proposed reduced-order method.

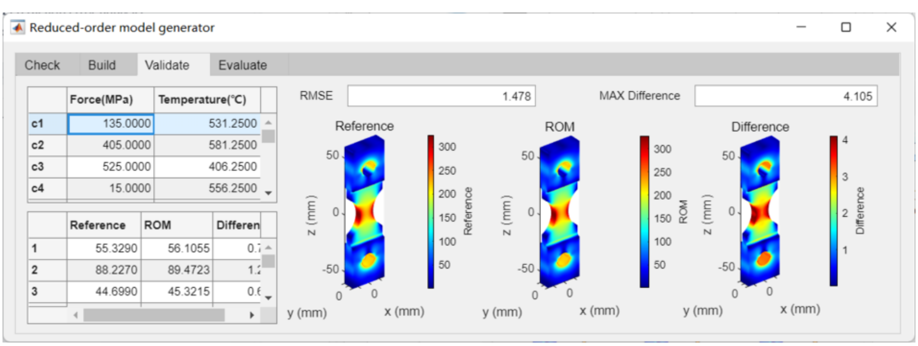

Figure 5. The error between the reduced-order model and the simulation model

5.4 Evaluate Module

In the Evaluate module, users can quickly predict results by dragging the slider to change the working condition parameters and are not limited to the working conditions in the previous training data. The results cloud map is displayed in real-time, as shown in Figure 6.

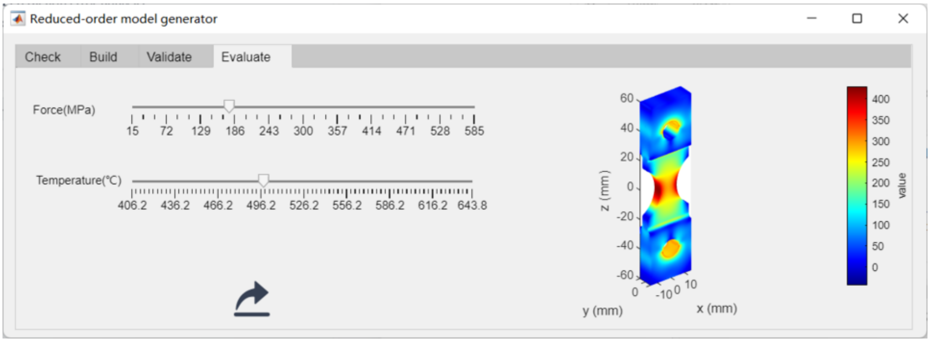

Figure 6. A rapidly calculated visual reduced-order model

6. Conclusion

This paper investigates a model order reduction method based on singular value decomposition (SVD) and neural network regression. An orthogonal experimental design was conducted, and training data was obtained through ANSYS finite element numerical simulation. Secondly, the principle of SVD reduction was explained, the result data was reduced, the feature vector was extracted, and the calculation method of each order error was analyzed. Then, the relationship between the working conditions and the feature vector was fitted, the trained regression model was applied to input the working conditions and obtain the predicted results, and the prediction error was analyzed and compared. Finally, a model order reduction generator was successfully built based on MATLAB, which can construct a reduced-order model using SVD decomposition and neural network regression technology by importing input-output data and mesh information of the finite element model, achieving the goal of quickly calculating results based on working conditions. The effectiveness and accuracy of the reduced-order model were demonstrated through a specific case.

References

[1] Jing, L.T., Dong, X.H., Yang, C., Fan, W., Li, T., Wang, L. (2023) Research on transformer temperature field finite element reduced-order modeling method for digital twin applications. High Voltage Technology: 1-12. https://doi.org/10.13336/j.1003-6520.hve.20220689.

[2] David, J. L., Philip, S.B., Walter, A.S. (2004) Reduced-order modeling: new approaches for computational physics. Progress in Aerospace Sciences, 40(1-2): 51-117. https://doi.org/10.1016/j.paerosci.2003.12.001.

[3] Wei, M., Ye, M., Wang, Q., et al. (2021) Remaining useful life prediction of lithium-ion batteries basedon stacked autoencoder and gaussian mixture regression. Journal of Energy Storage, 7: 2862-2871. https://doi.org/10.1016/j.egyr.2021.05.019.

[4] Jiao, L.C. (1990) Theory of neural network systems. In: Xi'an Electronic Science and Technology University Press. Xi'an. https://xueshu.baidu.com/usercenter/paper/show?paperid=aa11bdccf90a4a0a86261ad1ad8a7292&site=xueshu_se&hitarticle=1.
[5] Zhou, Z. (2008). A review of the current state of development of BP neural networks. Shanxi Electronics Technology, 2: 90-92. CNKI: SUN: SXDS.0.2008-02-037.
[6] Glorot, X., Bengio, Y. (2010) Understanding the difficulty of training deep feedforward neural networks. Journal of Machine Learning Research, 9: 249-256. https://www.xueshufan.com/publication/1533861849.
[7] Zhao, W., Dai, T.T., Wang, L.C., Lu, K., Chen, N. (2018) Short-term Load Forecasting Considering Meteorological Factors and Electric Vehicles. In: IOP Conference Series Materials Science and Engineering. Singapore.439(3). https://doi.org/10.1088/1757-899X/439/3/032114.
[8] Dai, H., Wang, K.K. (2020) Neighborhood point type method for simultaneous correction of mass and stiffness matrices of finite element models. Journal of Nanjing University of Aeronautics and Astronautics, 37(1): 12. https://www.nstl.gov.cn/paper_detail.html?id=225071796f3065fd00014c5f30ca597b.
[9] Shi, Y., Liu, S. (2018) Tag Recommendation Based on High Order Singular Value Decomposition. 2018 International Conference on Network, Communication, Computer Engineering (NCCE 2018). Chongqing. https://doi.org/10.2991/ncce-18.2018.122.
[10] Bai, Z.Y. (2011) Template for solving algebraic eigenvalue problems: a practical guide. Tsinghua University Press, Beijing. http://find.nlc.cn/search/showDocDetails?docId=1654505295963602295&dataSource=ucs01&query=978-7-302-24496-7.
[11] Xu, Y.G., Jiang, J., Tang, K.M., et al. (2020) Partial discharge narrow-band interference suppression method based on Hankel matrix and singular value decomposition. Grid Technology, 44(7): 8. https://doi.org/10.13335/j.1000-3673.pst.2019.1064.
[12] Shrestha, R., Di, L., Yu, E., Bai, Y. (2017) Regression model to estimate flood impact on corn yield using modis ndvi and usda cropland data layer. Journal of Integrative Agriculture, 16 (02): 398-407. https://doi.org/10.1016/S2095-3119(16)61502-2.