Can CNNs Construct Highly Accurate Model Efficiently with Limited Training Samples for Hundreds-Dimensional Problems?

Yu Li, first author
State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University
Changsha, 410082, PR China
liyu_hnu@hnu.edu.cn

Hu Wang, second author¹
State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University
Changsha, 410082, PR China

Juanjuan Liu, third author
State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University
Changsha, 410082, PR China
jjliu@hnu.edu.cn

ABSTRACT

With the increase of the nonlinearity and dimension, it is difficult for the present popular metamodeling techniques to construct reliable metamodels. To address this problem, Convolutional Neural Network (CNN) is introduced to construct a highly accurate metamodel efficiently. Considering the inherent characteristics of the CNN, it is a potential modeling tool to handle highly nonlinear and dimensional problems (hundreds-dimensional problems) with the limited training samples. In order to evaluate the proposed CNN metamodel for hundreds-dimensional and strong nonlinear problems, CNN is compared with other metamodeling techniques. Furthermore, several high-dimensional analytical functions are

¹ Corresponding author. Tel.: +86 0731 88655012; fax: +86 0731 88822051.
E-mail address: wanghu@hnu.edu.cn (H. Wang)
also employed to test the CNN metamodel. Testing and comparisons confirm the efficiency and capability of the CNN metamodel for hundreds-dimensional and strong nonlinear problems. Moreover, the proposed CNN metamodel is also applied to IsoGeometric Analysis (IGA)-based optimization successfully.

1. INTRODUCTION

At present, many engineering analyses appear complicated with huge simulation codes, such as Finite Element Analysis (FEA). Despite the increasing of speed and capability of computer processors, the huge computational costs of the complex analyses restrict pace. A popular way to overcome this is to generate an approximation model of the complex analysis process at a much lower cost, which is usually called metamodel [1]. Mathematically, on the assumption that the input to the actual analysis is $x$ and the output is $y$, the true analysis code evaluates

$$y = f(x)$$

where $f(x)$ is a complex function. The metamodel approximation can be presented as

$$\hat{y} = g(x)$$

such that

$$y = \hat{y} + \varepsilon$$

where $\varepsilon$ includes both approximation and random errors.

There are multiple kinds of metamodeling techniques to approximate $f(x)$ by $g(x)$ [2], e.g. Polynomial Regression (PR) [3, 4], Multivariate Adaptive Regression Spline (MARS) [5], Radial Basis Function (RBF) [6-8], Kriging (KG) [1, 9, 10], Support Vector Machine (SVM) [11-13], Back Propagation Neural Network (BPNN) [14, 15], etc. These models have become increasingly popular and a continuous improvement in recent
years. However, they are suffered from some well-known bottlenecks in high-dimensional design optimizations [10, 16, 17]. In the past 10 years, the number of design variables for complex design problems has been increased significantly. For most of recently published literatures, the surrogate assisted model can handle more than 10 design variables [2, 18, 19]. For Congress on Evolutionary Computation (CEC), the dimensionality of the high-dimensional problems has been regarded as more than 100. Practically, some original design problems, such as vehicle body structure, aircraft design and others, possess more than several hundreds design variables. Due to difficulties for modeling, most of these problems should be decomposed of several problems. However, some decomposed problems are actually not weak correlative. In order to ameliorate the "curse of dimensionality" [17, 20], a CNN is suggested for the geometry optimization. Simultaneously, it should be expected that the proposed CNN metamodel can be applied to the high-dimensional engineering problems.

In this study, the CNN is investigated as an alternative algorithm for approximations of complex analyses whose dimension are more than hundreds. Inspired by Hubel and Wiesel’s breakthrough findings in the cells of animal visual cortex [21, 22], Fukushima [23] proposed a hierarchical model called Neocognitron which could be regarded as the predecessor of the CNNs. Until 2012, with the appearance of AlexNet [24], the CNN’s advantages of local connection, weight sharing, and local pooling are widely recognized. Actually, the CNN has achieved state-of-the-art results in classification tasks [25] and been successfully utilized to many different tasks including speech processing [26, 27], image recognition [28-30], object detection [31-
Considering the success of the CNN for pattern recognition, the CNN might create a feasible way for hundreds-dimensional problems, because the actual inputs to the CNN are inputs’ pixel matrix whose dimensions can be increased to million level. Thus, we suggest the CNN to construct a higher-dimensional model compared with existing metamodel techniques.

The remainder of this paper is organized as follows. In Section 2, some related works are reviewed. In Section 3, the CNN is applied to analyze the hundreds-dimensional and strong nonlinear problems. The detailed experimental testing, results, and analyses are shown in Section 4. After a series of observations and analyses, the proposed CNN metamodel is applied to the optimization of a geometric structure in Section 5. The summaries are given in the final section.

2. RELATED WORKS

2.1. Metamodelling techniques

Response Surface Methodology (RSM) \[8\] approximates functions by the Least Squares method on a series of points in the design space. Low-order polynomials are the most widely used response surface approximating functions. The first-order and second-order polynomials are calculated by Eqs. (4) - (5) respectively.

\[
\hat{y} = b_0 + \sum_{i=1}^{k} b_i x_i 
\]  

\[
\hat{y} = b_0 + \sum_{i=1}^{k} b_i x_i + \sum_{i=2}^{k} b_{ii} x_i^2 + \sum_{i,j} b_{ij} x_i x_j \] 

where the constants \((b_0, b_i, b_{ii}, b_{ij})\) are determined by Least Squares regression.
Radial Basis Function (RBF) \[6-8\] attempts to approximate by a linear combination of radially symmetric functions. The RBF has produced good approximations to arbitrary contours. Mathematically, the model can be expressed as

\[ \hat{y} = \sum a_i \| x - x_{oi} \| \]  \hspace{1cm} (6)

where \( a_i \) is a real-valued weight, and \( X_{oi} \) is the input vector.

Kriging (KG) \[1, 9, 10\] is a combination of a known function and departures of the form.

\[ y(x) = f(x) + Z(x) \]  \hspace{1cm} (7)

where \( f(x) \) is a polynomial function which is often taken as a constant, and \( Z(x) \) is the correlation function which is a realization of a stochastic process with mean zero, variance \( \sigma^2 \), and nonzero covariance.

Moving Least Squares Method (MLSM) \[36\] is a metamodel technique that has been utilized in the meshless. It is a generalization of a traditional weighted Least Squares model where weights are the functions of Euclidian distance from a sampling point to a point \( x \).

Multivariate Adaptive Regression Splines (MARS) \[5\] is a nonparametric regression procedure that makes no assumption about the underlying functional relationship between dependent and independent variables. Instead, the MARS constructs this relation from a set of coefficients and basis functions which are determined from regression data. The input space is partitioned by regions of various regression equations. The MARS is suitable for problems with high dimension where the "curse of dimensionality" would likely create difficulties for other techniques.
Support Vector Machine (SVM) [11-13] is a binary linear classification technique in Machine Learning (ML), which separates the classes with largest gap (called optimal margin) between the border line instances (called Support Vectors).

Back Propagation Neural Network (BPNN) was introduced by Rumelhart and McClelland [14, 15]. It is a type of multi-layered feed-forward neural networks which minimize errors backward and where information is transmitted forward [37]. The BPNN has been widely used in many fields (e.g. image recognition [38-40], fault diagnosis [41-43], optimization [44-46], and others), and many intelligent evolution algorithms have also been used to select the initial connection weights and thresholds (e.g. Genetic Algorithm (GA) [47], Particle Swarm Optimization (PSO) [48]).

2.2. Convolutional Neural Network

According to Fig. 1, The first CNN architecture is proposed by LeCun [49, 50] in 1990. However, due to lack of enough training samples and computational power at that time, the CNN couldn't perform well for very complex problems. With the success of AlexNet [24] until 2012, many models have been proposed to improve its performance, e.g. ZFNet [51], VGGNet [52], GoogleNet [53], ResNet [54], InceptionNet [55, 56], Network in Network [57], etc. Since then, the study of the CNN can be mainly divided into five directions: optimization for network architecture [52, 58, 59], enhancement for convolutional layer [51, 56, 57, 60], attention for detection task [61-63], addition of new architectures [55, 64, 65], and unsupervised learning [66-69].

As for the evolution of the CNN architecture, a typical trend is that the network is getting deeper. However, deeper network increases its complexity at the same time,
which makes the network more difficult to be optimized and easier to be overfitting [33, 70, 71]. Along this way, different methods have been proposed to improve these problems as shown in Table 1.

3. CONVOLUTIONAL NEURAL NETWORK with MIXED POOLING

The key features of the CNN architecture are described below:

i. **Input layer.** This layer takes an input matrix of $W \times H \times 3$ (RGB image) or $W \times H \times 1$ (monochrome image);

ii. **Convolutional layer.** Compared with the classical full connection layer, the input to each neuron of this layer is a small part from the neurons of the upper layer;

iii. **Pooling layer.** It reduces the first two dimensions of the input matrix, and the third dimension will not be changed, which lowers the computational burden of the full connection layer;

iv. **Full connection layer.** After several convolutional and pooling layers, it can be considered that the input information has been abstracted into a very high level. Then a full connection layer is still necessary to complete the model;

v. **Softmax layer.** The softmax layer is mainly used for the classification problems, if the problem is regression, this layer is not necessary.

As shown in Fig. 2, the CNN employed in this study has two convolutional layers, two pooling layers and two full connection layers. Overfitting problems might be generated in very complicated problems despite limited samples, when the purpose of the CNN in this study is to use limited samples for modeling. To circumvent this, the mixed pooling layer [72] is employed.
In the CNN, two popular pooling methods, max pooling and average pooling, are well known and employed. However, both the max pooling and average pooling strategies have their own drawbacks. For max pooling, it only considers the maximum value and ignores the others in the pooling region. Sometimes, it might lead to an unacceptable result \([72]\). Regarding average pooling, it calculates the mean of all values within the pooling region. It will take all the low magnitudes into consideration and the contrast of the new feature map after pooling will be reduced. Even worse, if there are too many zeros, the characteristic of the feature map will be reduced largely \([72]\).

Through experiments, it demonstrates that the mixed pooling is superior to the traditional max pooling and average pooling to address the over-fitting problems and improve the accuracy. The mixed pooling generates the pooled output with the following formula.

\[
y_{i,j,k} = \lambda \max_{(m,n) \in S_k} \{a_{m,n,k}\} + (1-\lambda) \frac{1}{\#I_{(m,n) \in S_k}} \sum_{(m,n) \in S_k} a_{m,n,k}
\]

(8)

where \(\lambda\) is a random value in \((0, 1)\), \(a_{m,n,k}\) is the feature value at location \((m, n)\) within the pooling region \(S_k\) in \(k\)-th feature map.

The activation function is the Leaky Rectified Linear Units (Leaky ReLU) function \([73]\) which alleviates the potential disadvantage of ReLU that it has zero gradient when the input is negative, and the Leaky ReLU is calculated by

\[
f_{\text{ReLU}}(x) = \max(x, 0) + \lambda \min(x, 0)
\]

(9)

where \(x\) is the input matrix to the activation function.

The loss function is the Mean Square Error (MSE) plus \(L_2\) regularization which
makes the optimization of loss function easier. The loss function can be represented as

\[
\text{total _ losses} = \text{MSE}(y, \hat{y}) + \lambda R_2(w)
\]  

(10)

where

\[
\text{MSE}(y, \hat{y}) = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}
\]  

(11)

\[
R_2(w) = \|w\|_2^2 = \sum |w_i|^2
\]  

(12)

where \(y_i\) is the actual value, \(\hat{y}_i\) is the predicted value, \(n\) is the training sample size, and \(w_i\) is each parameter of the weight matrix \(w\).

The optimization algorithm used in this study is Adaptive Moment Estimation (Adam) which is essential RMSProp with momentum factor, and its advantages are as follows.

i. Adam integrates the advantages of AdaGrad to handle sparse gradients and RMSProp for non-stationary targets;

ii. Adam spends lower computational cost;

iii. Adam can calculate different adaptive learning rates for different parameters;

iv. Adam performs well for most nonconvex optimizations, large data sets, and high-dimensional spaces.

Mathematically, Adam can be defined as

\[
g \leftarrow 1/n \nabla_{\theta} \sum_i l(f(x_i; \theta), y_i)
\]  

(13)

\[
s \leftarrow \rho_1 s + (1 - \rho_1)g
\]  

(14)

\[
r \leftarrow \rho_2 r + (1 - \rho_2)g \odot g
\]  

(15)

\[
s' \leftarrow \frac{s}{1 - \rho_1}
\]  

(16)
\[
\hat{r} \leftarrow \frac{r}{1 - \rho_2} \tag{17}
\]
\[
\Delta \theta = -\varepsilon \frac{\hat{s}}{\sqrt{\hat{r} + \delta}} \tag{18}
\]
\[
\theta \leftarrow \theta + \Delta \theta \tag{19}
\]

where \( L(x) \) is the loss function, \( \theta \) is the initial parameter, \( x_i \) is the training sample and \( y_i \) is corresponding label, \( s \) and \( r \) are the first and second moment estimations respectively, \( \rho_1 \) and \( \rho_2 \) are the attenuation coefficients, and \( \varepsilon \) is the learning rate. In this study, \( \delta = 10^{-8} \), \( \rho_1 = 0.9 \), and \( \rho_2 = 0.999 \).

4. EXPERIMENTS and ANALYSES

4.1. Analytical examples

In order to evaluate the CNN's performance, several typical high-dimensional and strong nonlinear mathematical functions shown in Table 2 are employed.

4.2. Performance Criteria

There are several commonly used performance criteria for approximation models. However, as mentioned in Ref. [7], there are no specially defined performance criteria for high-dimensional approximation models in the open literatures. To be consistent with Refs. [7] and [19], which will be subsequently used for the comparison with the proposed CNN metamodel, three commonly used performance criteria, \( R^2 \), Relative Average Absolute Error (RAAE), and Relative Maximum Absolute Error (RMAE), which are shown in Table 3, are employed for validating approximation models. A small RMAE is preferred. The larger the \( R^2 \) is, the more accurate the metamodel is. Large
RMAE indicates large error in the design space. However, since RMAE cannot show the overall performance in the design space, it is not as important as $R^2$ and RAAE.

4.3. Results and discussions

4.3.1. Performances of different metamodels. The Kriging (KG) shown in Ref. [7] is competent in both imitating implementing the dynamic simulation accurately and estimating the error of the predictor efficiently. At the same time, the Radial Basis Function with High Dimensional Model Representation (RBF-HDMR) proposed by Prof. Wang [10] works well in high-dimensional simulation-based problems. Furthermore, Refs. [10] and [7] have attracted more attention. Therefore, the KG and RBF-HDMR represented in Refs. [10] and [7] have been employed and compared with the proposed CNN metamodel respectively.

Jin selected 14 problems and classified them into two categories: large scale and small scale in Ref. [19]. The large-scale problems include one 14-dimension, one 16-dimension and four 10-dimension problems. The small-scale includes five 2-dimension and three 3-dimension problems. As for Ref. [7], Shan and Wang dealt with high-dimensional problems. Therefore, only the test results of large problems (Problems 1 - 5 shown in Ref. [19]) are reported with the first and second order RBF-HDMR models. In Ref. [7], each problem runs ten times independently and then takes the average of ten runs for each problem.

According to Ref. [7] in Table 2, a strong nonlinear problem for large-scale optimization is chosen to study the performance of the RBF-HDMR, and the problem is tested when $d=30, 50, 100, 150, 200, 250$ and $300$. Since this study deals with hundreds-
dimensional problems, only the tested results for \(d=100, 150, 200, 250\) and \(300\) are reported in Table 4 and Fig. 3.

As shown in Fig. 3, ten runs are taken independently and the mean values of \(R^2\), RAAE, and RMAE are charted for each \(d\). According to Fig. 3, where the CNN is marked in orange and the RBF-HDMR is marked in blue, the performance criteria of the CNN are better than the RBF-HDMR's. The RBF-HDMR's criteria deteriorates slightly as \(d\) increases while the CNN remains stable. Moreover, as for necessary samples, shown in Table 4, the sample size of the RBF-HDMR is far larger than the size of the CNN when the dimension is hundreds, which clearly shows the computational advantage of the proposed CNN metamodel. In our opinion, if the sample size is far smaller than size in Ref. [7], the samples are so called "limited samples" in this study. The suitable sample size for the CNN will be further analyzed in Section 4.3.2.

Bouhlel considered the increasingly popular KG model suffers from some well-known drawbacks in high dimension and proposed a new method combined the KG with the Partial Least Squares (KPLS) which can be found in Ref. [10]. In Ref. [10], two academic functions are used. The first function is \(g07\) with 10 dimensions, and the second is the Griewank function, whose dimension is set to be 5, 7, 10, 20, 60 and interval is \([-600, 600]\). The following relative error calculated by Eq. (20) is employed to evaluate the performance of the KPLS model.

\[
\text{Error} = \frac{\|y - \hat{y}\|_2}{\|y\|_2} \times 100\%
\]  

(20)

where \(\|\cdot\|\) represents the usual \(L_2\) norm.

As shown in Tables 3 - 8 of Ref. [10], shown in Fig. 4, the error of the KPLS model
are satisfied when the dimension is within 10. However, when the dimension increases to dozens, the error of KPLS has a significant increase. For the CNN model, shown in Table 5, the error, which is the mean value of ten runs, doesn’t have an obvious change when the dimension increases to 784.

Therefore, for the Griewank function over the interval \([-600, 600]\), the results obtained by the CNN are better than results by the KPLS. The proposed CNN method thus appears appealing, particularly for hundreds-dimensional problems. The comparative data might explain that the CNN metamodel has a better modeling ability for higher-dimensional problems.

4.3.2. Discussions for suitable sample size of CNN. It’s necessary to find the suitable sample size between different dimensions for the CNN. The Griewank function is employed and its input interval is divided into four sub-intervals, \([0, 5]\), \([0, 10]\), \([0, 100]\) and \([0, 600]\), to test the suitable sample size for different dimensions. For each interval, the input dimension is divided into 100d, 324d, and 784d. In this section, ten runs have been taken for each dimension and interval, and the mean criteria have been recorded. According to the mean criteria shown in Tables 6 - 8, the suitable sample sizes for 100d, 324d and 784d are all 10000 which are marked in blue.

4.3.3. Performance of CNN for analytical examples. As shown in Fig. 5, some other mathematical models shown in Table 2, whose dimensions are set to 324, are tested to further evaluate the CNN metamodel’s performance for hundreds-dimensional and strong nonlinear problems. It can be seen that the minimum (worst) \(R^2\) is 0.81, the maximum (worst) RAAE is 0.56, and the maximum (worst) RMAE is 2.61. The data
explains the CNN can model for hundreds-dimensional and strong nonlinear problems satisfactorily.

5. GEOMETRIC MODELING by CNN

In the traditional design-and-analysis procedure, the geometric design and physical analysis are commonly considered as completely different engineering fields. In the process of design, the gap between Computer Aided Design (CAD) and Computer Aided Engineering (CAE) models always exists. Thus, most of time is wasted on the match between CAD and CAE models in the traditional design [74]. In order to integrate CAD and CAE models seamlessly, Hughes [74] proposed a CAD/CAE integration method named as IsoGeometric Analysis (IGA), where Non-Uniform Rational B-Spline (NURBS) is employed as a discretization technology for geometry design and analysis. The IGA shows its unique advantages [75, 76] as follows.

i. The division of the parametrical domain for spline model and the mapping from parametrical domain to physical model avoid the time consumption of grid calculation in traditional finite element;

ii. The interaction between geometric model and analysis model is avoided due to the unified expression;

iii. The parameterization of model creates a feasible way for structural optimization;

iv. The polynomial of basis function is directly derived from the geometric model, which avoids the error caused by interpolation approximation.

The advantages shown in IGA have attracted more scholars to perfect its theories
and applications. Wall [77] and Fußeder [78] combined the IGA with Gradient-driven Optimization (GDO) to achieve the structural optimization of two-dimensional shape. Nagy [79] presented the structural size and shape optimization of curved beam structures. Manh [80] applied the IGA to solve the optimization problem of membrane vibration. Additionally, isogeometric shape optimization has been used to address the problems of electromagnetic scattering [81], heat conduction [82, 83], fluid mechanics [84, 85], etc.

5.1. The bases of isogeometric design

It is well known that the NURBS is a basis of geometric design for the IGA, and its basis function can be mapped by the B-spline function.

\[
R_{i,j}^{p,q}(\xi,\eta) = \frac{N_i(\xi)M_j(\eta)\omega_{i,j}}{\sum_{i=1}^{n} \sum_{j=1}^{m} N_i(\xi)M_j(\eta)\omega_{i,j}}
\]

(21)

where \( p, q \) are the polynomial orders in different dimensions, \( N_i(\xi) \) and \( M_j(\eta) \), whose numbers are \( n \) and \( m \) respectively, are the standard B-spline basis functions in two dimensions, and \( \omega_i \) is the weight value of each control point.

Given a set of control points \( P_i \) and a combination of bivariate NURBS basis functions, NURBS surface can be defined as

\[
S(\xi,\eta) = \sum_{i=1}^{n} \sum_{j=1}^{m} P_{i,j} R_{i,j}^{p,q}(\xi,\eta)
\]

(22)

When the parametric model is constructed, the displacement of the IGA can be obtained as

\[
u(\xi,\eta) = \sum_{i=1}^{n} \sum_{j=1}^{m} R_{i,j}^{p,q}(\xi,\eta)d
\]

(23)
\[ \delta u = \sum_{i=1}^{n} \sum_{j=1}^{m} R^{ij}_{p,q} (\xi, \eta) \delta d \]  

(24)

where \( d \) and \( \delta d \) are the displacement vector at control points and displacement differential vectors respectively.

5.2. Modeling

In this study, Rhinoceros software is employed to provide the input information of a tubular structure as shown in Fig. 6. As shown in Figs. 7(b) - (c), the Degree of Freedoms (DOFs) marked in red are fixed. In addition, three \( 3 \times 10^5 N \) concentrated forces in \( y \) direction, shown in Fig. 7(a), are enforced to the control points marked in orange. Finally, the stress distribution of the initial model obtained by the IGA is displayed in Fig. 8.

5.3. Metamodel by CNN

The control points are used as the input variables and the maximum stress after 2500 iterations is each sample's label. For each sample, design variables include 324 control points, and each control point includes coordinate \( (x_i, y_i, z_i) \) and a weight factor \( w_i \). However, the tubular structure particularly changes only in the direction of the \( y \) axis, and each \( w_i \) is fixed. Therefore, the input matrix contains 324 \( y_i \)s.

Each \( y_i \) changes randomly in the interval \([0.8y_i, 1.2y_i]\), and 10000 training and 10000 testing samples are obtained. The total losses during the training process are shown in Fig. 9 and the corresponding performance criteria are shown in Table 9. It can be found that RAAE, RMAE are small enough. As \( R^2 \) is 0.997, it indicates that the trained CNN metamodel is good enough.
Then, the constructed metamodel is optimized by Particle Swarm Optimization (PSO). The objective function to be minimized is the maximum stress in the tubular structure, and the $y_i$s of control points are chosen as the design variables. The optimization problem is stated as

$$
\min f = \text{CNN}(\mathbf{B}(y_1, y_2, \ldots, y_{324}))
$$

(25)

s.t.

$$
y_i \in [0.8y_{i\text{-init}}, 1.2y_{i\text{-init}}], \ i = 1, \ldots, 324
$$

(26)

where $\text{CNN}(\mathbf{x})$ is the maximum stress in the tubular structure, $\mathbf{B}$ is the control points, and $y_{i\text{-init}}$ is the initial value of each $y_i$.

The comparison between optimizations of the CNN and the IGA is shown in Fig. 10. The optimal stress distribution by the CNN is shown in Fig. 11. The optimum by the CNN is 1,340,470$N$ and the optimum by the IGA is 1,354,991$N$. It can be seen that the optimum by the CNN is 1.072% less than the optimum by the IGA. Moreover, the optimization process of CNN begins to converge at the 60$^{th}$ step and IGA begins at the 91$^{st}$ step. The computational time for each iteration of CNN and IGA are 20$s$ and 500$s$, respectively. It can be inferred that the optimization by the CNN can significantly reduce the time with high accuracy.

**CONCLUSIONS**

In this study, a metamodeling method based on Convolutional Neural Network (CNN) is proposed for hundreds-dimensional and strong nonlinear problems. The contributions can be summarized as follows.
In order to evaluate the proposed CNN metamodel, Radial Basis Function with High Dimensional Model Representation (RBF-HDMR) and Kriging (KG) in Refs. [7] and [10] respectively have been employed and compared. The compared results demonstrate that the proposed CNN metamodel has an obvious advantage for hundreds-dimensional and strong nonlinear problems.

For further analyzing the CNN metamodel, several high-dimensional and nonlinear analytical functions are selected to test CNN metamodel's performance. The testing results are also satisfied.

Finally, the proposed CNN metamodel is applied to an IGA-based optimization. It is found that the proposed CNN metamodel can be applied to hundreds-dimensional and strong nonlinear applications well by using limited samples and has a faster convergence rate for optimization compared with the IGA.

ACKNOWLEDGMENT

This work has been supported by Project of the National Key R&D Program of China 2017YFB0203701 and Key Program of National Natural Science Foundation of China under the Grant Numbers 11572120.
NOMENCLATURE

\( x \quad \text{The input matrix.} \)

\( y \quad \text{The output matrix from the metamodel.} \)

\( \hat{y} \quad \text{The approximation of } y. \)

\( (b_0, b_i, b_{ii}, b_{ij}) \quad \text{Constants determined by Least Squares regression.} \)

\( a_i \quad \text{The real-valued weight.} \)

\( f(x) \quad \text{The polynomial function which is often taken as a constant.} \)

\( Z(x) \quad \text{The correlation function which is a realization of a stochastic process} \)

\( \lambda \quad \text{A random value in } (0, 1). \)

\( a_{m,n,k} \quad \text{feature value at location } (m, n) \text{ within the pooling region } R_{i,j} \text{ in } k\text{-th} \)

\( x_i \quad \text{The training sample.} \)

\( y_i \quad \text{The actual value.} \)

\( \hat{y}_i \quad \text{The predicted value.} \)

\( n \quad \text{The training sample size.} \)

\( w_i \quad \text{The each parameter of the weight matrix } w. \)

\( L(x) \quad \text{The loss function.} \)

\( \theta \quad \text{The initial parameter of the neural network.} \)

\( s, r \quad \text{The first and second moment estimations respectively.} \)
| Symbol | Description |
|--------|-------------|
| $\varepsilon$ | The learning rate. |
| $\rho_1, \rho_2$ | The attenuation coefficients. |
| $p, q$ | The polynomial orders in different dimensions. |
| $d, \delta d$ | The displacement vector at control points and displacement differential vectors respectively. |
| $CNN(x)$ | The maximum stress in the tubular structure. |
| $B$ | The control points. |
| $y_{i-init}$ | The $y$-axis initial coordinate value of each point in the $B$. |
REFERENCES

[1]. Roshan V, Ying H, Sudjianto A. Blind Kriging: A New Method for Developing Metamodels. Journal of Mechanical Design. 2008;130(3):350-3.

[2]. Qiu H, Xu Y, Gao L, Li X, Chi L. Multi-stage design space reduction and metamodeling optimization method based on self-organizing maps and fuzzy clustering. Expert Systems with Applications An International Journal. 2016;46(C):180-95.

[3]. AIAA. Aerodynamic configuration design using response surface methodology analysis. NASA STI/Recon Technical Report A. 1993;94.

[4]. Unal R, Lepsch R, Engelund W, Stanley D, editors. Approximation model building and multidisciplinary design optimization using response surface methods. Symposium on Multidisciplinary Analysis and Optimization; 2013.

[5]. Friedman JH. Multivariate Adaptive Regression Splines. Annals of Statistics. 1991;19(1):1-67.

[6]. Hardy RL. Multiquadric equations of topography and other irregular surfaces. Journal of Geophysical Research. 1971;76(8):1905-15.

[7]. Shan S, Wang GG. Metamodeling for High Dimensional Simulation-Based Design Problems. Journal of Mechanical Design. 2010;132(5).

[8]. Simpson TW, Piplinski JD, Koch PN, Allen JK. Metamodels for Computer-based Engineering Design: Survey and recommendations. Engineering with Computers. 2001;17(2):129-50.

[9]. Li M, Li G, Azarm S. A Kriging Metamodel Assisted Multi-Objective Genetic Algorithm for Design Optimization. Journal of Mechanical Design. 2006;130(3):405-14.

[10]. Bouhlel MA, Bartoli N, Otsmane A, Morlier J. Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction. Structural & Multidisciplinary Optimization. 2016;53(5):935-52.

[11]. Boser BE, Guyon IM, Vapnik VN, editors. A training algorithm for optimal margin classifiers. The Workshop on Computational Learning Theory; 1992.

[12]. Clarke SM GJ, Simpson TW, editors. Analysis of Support Vector Regression for Approximation of Complex Engineering Analyses. Journal of Mechanical Design. NOVEMBER 2005;127 / 1077.

[13]. Wang H, Shan S, Wang GG, Li G. Integrating Least Square Support Vector Regression and Mode Pursuing Sampling Optimization for Crashworthiness Design. Journal of Mechanical Design. 2011;133(4):181-9.

[14]. Rumelhart DE, editor McClelland and the PDP Research Group. Symposium on Parallel and Distributed Processing; 1986.

[15]. Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature. 1986;323(6088):533-6.

[16]. Xu S, Chen H. Nash game based efficient global optimization for large-scale design problems. Journal of Global Optimization. 2018(4):1-21.

[17]. Shan S, Wang GG. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. Structural & Multidisciplinary Optimization. 2010;41(2):219-41.

[18]. Simpson T, Toropov V, Balabanov V, Viana F, editors. Design and Analysis of Computer Experiments in Multidisciplinary Design Optimization: A Review of How Far We Have Come - Or Not2008.

[19]. Jin R, Chen W, Simpson TW. Comparative studies of metamodeling techniques under multiple modelling criteria. Structural & Multidisciplinary Optimization. 2001;23(1):1-13.
[20]. Koch PN, Simpson TW, Allen JK, Mistree F. Statistical Approximations for Multidisciplinary Design Optimization: The Problem of Size. Journal of Aircraft. 1999;36(1):275-86.
[21]. Hubel DH, Wiesel TN. Receptive fields of single neurones in the cat's striate cortex. Journal of Physiology. 1959;148(3):574.
[22]. Hubel DH, Wiesel TN. Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. J Physiol. 1962;160(1):106-54.
[23]. Fukushima K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics. 1980;36(4):193-202.
[24]. Krizhevsky A, Sutskever I, Hinton GE, editors. ImageNet classification with deep convolutional neural networks. International Conference on Neural Information Processing Systems; 2012.
[25]. Koziarski M, Cyganek B. Image recognition with deep neural networks in presence of noise – Dealing with and taking advantage of distortions. Integrated Computer Aided Engineering. 2017(4):1-13.
[26]. Zhao R, Ouyang W, Li H, Wang X, editors. Saliency detection by multi-context deep learning. Computer Vision and Pattern Recognition; 2015.
[27]. Sercu T, Goel V. Advances in Very Deep Convolutional Neural Networks for LVCSR. 2016.
[28]. Tong N, Lu H, Xiang R, Yang MH, editors. Salient object detection via bootstrap learning. Computer Vision and Pattern Recognition; 2015.
[29]. Gidaris S, Komodakis N, editors. Object Detection via a Multi-region and Semantic Segmentation-Aware CNN Model. IEEE International Conference on Computer Vision; 2015.
[30]. Gatys LA, Ecker AS, Bethge M. Texture Synthesis Using Convolutional Neural Networks. 2015;70(1):262-70.
[31]. Vinay A, Reddy DN, Sharma AC, Daksha S, Bhargav NS, Kiran MK, et al., editors. G-CNN and F-CNN: Two CNN based architectures for face recognition. International Conference on Big Data Analytics and Computational Intelligence; 2017.
[32]. Shin HC, Roth HR, Gao M, Lu L, Xu Z, Nogues I, et al. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. IEEE Transactions on Medical Imaging. 2016;35(5):1285-98.
[33]. Wang P, Bai X. Regional parallel structure based CNN for thermal infrared face identification. Integrated Computer Aided Engineering. 2018(9):1-14.
[34]. Jaderberg M, Simonyan K, Vedaldi A, Zisserman A. Reading Text in the Wild with Convolutional Neural Networks. International Journal of Computer Vision. 2016;116(1):1-20.
[35]. Shi B, Bai X, Yao C. An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. IEEE Transactions on Pattern Analysis & Machine Intelligence. 2017;PP(99):1-1.
[36]. Toropov VV, Schramm U, Sahai A, Jones RD, Zeguer T. Design Optimization and Stochastic Analysis based on the Moving Least Squares Method. 2005.
[37]. Zhang G, Patuwo BE, Hu MY. Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting. 1998;14(1):35-62.
[38]. Agarwal P, Prakash N. An efficient back propagation neural network based face recognition system using haar wavelet transform and PCA. International Journal of Computer Science & Mobile Computing. 2013;2(5).
[39]. Murugan A, Savithiri G. Fragmented Iris Recognition System using BPNN. International Journal of Computer Applications. 2011;36(4):28-33.
[40]. Arun Joshi MSB, Ms. Jaspreet Kaur. GAIT RECOGNITION OF HUMAN USING SVM AND BPNN.
CLASSIFIERS. International Journal of Computer Science and Mobile Computing. January 2014;Vol. 3(Issue. 1):9.

[41].Deng S, Lin SY, Chang WL. Application of multiclass support vector machines for fault diagnosis of field air defense gun. Expert Systems with Applications. 2011;38(5):6007-13.

[42].Jin HS, Liu J, Sun YJ, Qiu DD. The Fault Diagnosis System of Measure and Control Radar Servo Motor Based on Ship in BP Neural Network. Applied Mechanics & Materials. 2013;401-403(2):1139-43.

[43].Kuo HC, Wu LJ, Chen JH. Neural-fuzzy fault diagnosis in a marine propulsion shaft system. Journal of Materials Processing Technology. 2002;122(1):12-22.

[44].Mu Y, Zhao G, Wu X, Zhang C. An optimization strategy for die design in the low-density polyethylene annular extrusion process based on FES/BPNN/NSGA-II. International Journal of Advanced Manufacturing Technology. 2010;50(5-8):517-32.

[45].Chen XY, Chen Z, Zhao Y. Numerical research on virtual reality of vibration characteristics of the motor based on GA-BPNN model. Neural Computing & Applications. 2017(5):1-13.

[46].Huang Y, Blackwell PL. Prediction of mechanical properties of superplastic Inconel* 718 using artificial neural networks. Metal Science Journal. 2002;18(10):1104-8.

[47].Irani R, Nasimi R. Evolving neural network using real coded genetic algorithm for permeability estimation of the reservoir. Expert Systems with Applications. 2011;38(8):9862-6.

[48].Zhang JR, Zhang J, Lok TM, Lyu MR. A hybrid particle swarm optimization-back-propagation algorithm for feedforward neural network training. Applied Mathematics & Computation. 2007;185(2):1026-37.

[49].Lecun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, et al. Backpropagation applied to handwritten zip code recognition. Neural Computation. 2014;1(4):541-51.

[50].Cun YL, Boser B, Denker JS, Howard RE, Hubbard W, Jackel LD, et al., editors. Handwritten digit recognition with a back-propagation network. Advances in Neural Information Processing Systems; 1990.

[51].Zeiler MD, Fergus R. Visualizing and Understanding Convolutional Networks. 2013;8689:818-33.

[52].Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. Computer Science. 2014.

[53].Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. 2014:1-9.

[54].He K, Zhang X, Ren S, Sun J, editors. Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision and Pattern Recognition; 2016.

[55].Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the Inception Architecture for Computer Vision. 2015:2818-26.

[56].Szegedy C, Ioffe S, Vanhoucke V, Alemi A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. 2016.

[57].Lin M, Chen Q, Yan S. Network In Network. Computer Science. 2013.

[58].Tenenbaum JB, Freeman WT. Separating style and content with bilinear models. Neural Computation. 2000;12(6):1247.

[59].He S, Lau RWH, Liu W, Huang Z, Yang Q. SuperCNN: A Superpixelwise Convolutional Neural Network for Salient Object Detection. International Journal of Computer Vision. 2015;115(3):330-44.

[60].Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the Inception Architecture for Computer Vision. Computer Science. 2015:2818-26.

[61].Girshick R, Donahue J, Darrell T, Malik J. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. IEEE Trans Pattern Anal Mach Intell. 2015;38(1):142-58.
[62]. Girshick R. Fast R-CNN. Computer Science. 2015.
[63]. Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis & Machine Intelligence. 2017;39(6):1137-49.
[64]. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis & Machine Intelligence. 2017;39(4):640-51.
[65]. Jaderberg M, Simonyan K, Zisserman A, Kavukcuoglu K. Spatial Transformer Networks. 2015:2017-25.
[66]. Radford A, Metz L, Chintala S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. Computer Science. 2015.
[67]. Martin Arjovsky SC, Leon Bottou. Wasserstein Generative Adversarial Networks. Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia. 2017.
[68]. Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville A. Improved Training of Wasserstein GANs. 2017.
[69]. Li Y, Wang H, Mo K, Zeng T. Reconstruction of Simulation-Based Physical Field with Limited Samples by ReConNN. 2018.
[70]. Koziarski M, Cyganek B. Image recognition with deep neural networks in presence of noise – Dealing with and taking advantage of distortions. Integrated Computer Aided Engineering. 2017;24(4):1-13.
[71]. Sánchez Á, Moreno AB, Vélez D, Vélez JF. Analyzing the influence of contrast in large-scale recognition of natural images. Integrated Computer Aided Engineering. 2016;23(3):221-35.
[72]. Yu D, Wang H, Chen P, Wei Z, editors. Mixed Pooling for Convolutional Neural Networks. the 9th international conference on rough sets and knowledge technology; 2014.
[73]. A.L. Maas AYH, A.Y. Ng. Rectifier nonlinearities improve neural network acoustic models. Proceedings of the International Conference on Machine Learning (ICML). 30, 2013.
[74]. Hughes TJR, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Computer Methods in Applied Mechanics and Engineering. 2005;194(39):4135-95.
[75]. Zijun WU. Perspectives on Isogeometric Analysis. Journal of Mechanical Engineering. 2015;51(5):114.
[76]. Nguyen VP, Anitescu C, Bordas SPA, Rabczuk T. Isogeometric analysis: An overview and computer implementation aspects. Mathematics & Computers in Simulation. 2015;117:89-116.
[77]. Wall WA, Frenzel MA, Cyron C. Isogeometric structural shape optimization. Computer Methods in Applied Mechanics and Engineering. 2008;197(33–40):2976-88.
[78]. Fußeder D, Simeon B, Vuong A-V. Fundamental aspects of shape optimization in the context of isogeometric analysis. Computer Methods in Applied Mechanics & Engineering. 2015;286:313-31.
[79]. Nagy AP, Abdalla MM, Gürdal Z. Isogeometric sizing and shape optimisation of beam structures. Computer Methods in Applied Mechanics & Engineering. 2010;199(17):1216-30.
[80]. Manh ND, Evgrafov A, Gersborg AR, Gravesen J. Isogeometric shape optimization of vibrating membranes. Computer Methods in Applied Mechanics & Engineering. 2010;200(13):1343-53.
[81]. Dang MN, Evgrafov A, Gravesen J. Isogeometric Shape Optimization for Electromagnetic Scattering Problems. Progress in Electromagnetics Research B. 2012;45(45):117-46.
[82]. Anders D, Weinberg K, Reichardt R. Isogeometric analysis of thermal diffusion in binary blends. Computational Materials Science. 2012;52(1):182-8.
[83]. Cho S, Ha SH. Isogeometric shape design optimization: exact geometry and enhanced sensitivity. Structural & Multidisciplinary Optimization. 2009;38(1):53-70.
[84]. Nørtoft P, Gravesen J. Isogeometric shape optimization in fluid mechanics. Structural & Multidisciplinary Optimization. 2013;48(5):909-25.

[85]. Bazilevs Y, Calo VM, Hughes TJR, Zhang Y. Isogeometric fluid-structure interaction: theory, algorithms, and computations. Computational Mechanics. 2008;43(1):3-37.

[86]. Ngiam J, Chen Z, Chia D, Koh PW, Le QV, Ng AY. Tiled convolutional neural networks. Nips in Press. 2010:1279-87.

[87]. Yu F, Koltun V. Multi-Scale Context Aggregation by Dilated Convolutions. 2015.

[88]. A H, U K. Complex cell pooling and the statistics of natural images. Network (Bristol, England). 2007;18(2):81.

[89]. Zeiler MD, Fergus R. Stochastic Pooling for Regularization of Deep Convolutional Neural Networks. Eprint Arxiv. 2013.

[90]. Gong Y, Wang L, Guo R, Lazebnik S. Multi-scale Orderless Pooling of Deep Convolutional Activation Features. 2014;8695:392-407.

[91]. Rippel O, Snoek J, Adams RP, editors. Spectral representations for convolutional neural networks. International Conference on Neural Information Processing Systems; 2015.

[92]. He K, Zhang X, Ren S, Sun J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. IEEE Transactions on Pattern Analysis & Machine Intelligence. 2014;37(9):1904-16.

[93]. Nair V, Hinton GE, editors. Rectified linear units improve restricted boltzmann machines. International Conference on International Conference on Machine Learning; 2010.

[94]. A.L. Maas AYH, A.Y. Ng, . Rectifier nonlinearities improve neural network acoustic models. Proceedings of the International Conference on Machine Learning (ICML). 2013;30.

[95]. He K, Zhang X, Ren S, Sun J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. 2015:1026-34.

[96]. Xu B, Wang N, Chen T, Li M. Empirical Evaluation of Rectified Activations in Convolutional Network. Computer Science. 2015.

[97]. Clevert D, Unterthiner T, Hochreiter S. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). Computer Science. 2015.

[98]. Goodfellow IJ, Warde-Farley D, Mirza M, Courville A, Bengio Y. Maxout Networks. Computer Science. 2013:1319-27.

[99]. Springenberg JT, Riedmiller M. Improving Deep Neural Networks with Probabilistic Maxout Units. Computer Science. 2013.

[100]. Jin J, Fu K, Zhang C. Traffic Sign Recognition With Hinge Loss Trained Convolutional Neural Networks. IEEE Transactions on Intelligent Transportation Systems. 2014;15(5):1991-2000.

[101]. Liu W, Wen Y, Yu Z, Yang M, editors. Large-margin softmax loss for convolutional neural networks. International Conference on International Conference on Machine Learning; 2016.

[102]. Lin J, Moreo O, Chandrasekhar V, Veillard A, Goh H. DeepHash: Getting Regularization, Depth and Fine-Tuning Right. Mccarthy. 2015.

[103]. Schroff F, Kalenichenko D, Philbin J. FaceNet: A Unified Embedding for Face Recognition and Clustering. 2015:815-23.

[104]. Tikhonov AN. On the stability of inverse problems. Dolkadaknauk Sssr. 1943;39(5):176-9.

[105]. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR. Improving neural networks by preventing co-adaptation of feature detectors. Computer Science. 2012;3(4):págs. 212-23.

[106]. Wang SI, Manning CD, editors. Fast dropout training. International Conference on International Conference on Machine Learning; 2013.
[107]. Ba LJ, Frey B, editors. Adaptive dropout for training deep neural networks. International Conference on Neural Information Processing Systems; 2013.

[108]. Tompson J, Goroshin R, Jain A, Lecun Y, Bregler C. Efficient object localization using Convolutional Networks. 2014:648-56.

[109]. Wan L, Zeiler MD, Zhang S, Lecun Y, Fergus R, editors. Regularization of Neural Networks using DropConnect. International Conference on Machine Learning; 2013.
Figure Captions List

Fig. 1  The development of CNN.

Fig. 2  The architecture of CNN with mixed pooling.

Fig. 3  Mean values of performance criteria for CNN and RBF-HDMR.

Fig. 4  Griewank function over the interval [-600, 600] by KPLS.

Fig. 5  Performance for hundreds-dimensional problems by CNN metamodel.

Fig. 6  The control lattice for the tubular structure based on IGA.

Fig. 7  Constraints and load of the tubular structure.

Fig. 8  Von Mises stress distribution of the initial model.

Fig. 9  Training process of CNN metamodel.

Fig. 10 The comparison between optimization of CNN and IGA.

Fig. 11 The optimal shape and stress distribution of the optimal structure by CNN.
Table Caption List

Table 1  The improvements of each aspect on CNN.
Table 2  Mathematical functions.
Table 3  Criteria for performance evaluation.
Table 4  Comparisons of modeling cost.
Table 5  Mean value of error for Griewank function over the interval [-600, 600] by CNN.
Table 6  Mean value of performance criteria when dimension is 100d.
Table 7  Mean value of performance criteria when dimension is 324d.
Table 8  Mean value of performance criteria when dimension is 784d.
Table 9  The performance criteria for trained CNN metamodel.
## Information Regarding Figures and Tables

**Table 1** The improvements of each aspect on CNN.

| Aspect               | Literature | Contribution                                      | Improvement                                                                 |
|----------------------|------------|---------------------------------------------------|----------------------------------------------------------------------------|
| Convolutional layer  | Ngiam [86] | Tiled CNN                                         | It tiles and multiples feature maps to learn rotational and scale invariant features. |
|                      | Zeiler [51]| Transposed CNN                                    | It can be seen as the backward pass of traditional CNNs.                   |
|                      | Yu [87]    | Dilated CNN                                       | It introduces more hyper-parameters to CNN which can increase receptive field size and cover more relevant information. |
|                      | Hyvärinen [88]| \(L_p\) Pooling                             | It has a better generalization than max pooling.                          |
| Pooling layer        | Zeiler [89]| Stochastic Pooling                               | It’s a dropout-inspired pooling method which increases generalization of CNN. |
|                      | Yu [72]    | Mixed Pooling                                     | It can better address the overfitting problems.                          |
|                      | Gong [90]  | Multi-scale Orderless Pooling                     | It improves the invariance of CNNs without degrading discriminative power. |
|                      | Rippel [91]| Spectral Pooling                                  | Compared with max pooling, it can preserve more information for the same output dimensionality. |
|                      | He [92]    | The second level headings                         | It can generate a fixed-length representation regardless of the input sizes.  |
|                      | Nair [93]  | ReLU                                              | It is one of the most notable nonsaturated activation functions.           |
|                      | Maas [94]  | Leaky ReLU                                        | It improves ReLU’s disadvantage of having zero gradient.                  |
|                      | He [95]    | Parametric ReLU                                   | It reduces the risk of overfitting and improves the accuracy.             |
|                      | Xu [96]    | Randomized ReLU                                   | It also reduces the risk of overfitting and improve the accuracy.         |
| Activation function  | Clevert [97]| ELU                                              | It enables faster learning of DNNs and leads to higher classification accuracies. |
|                      | Goodfellow [98]| Maxout                              | It enjoys all the benefits of ReLU and it is well suited for training with dropout. |
|                      | Springenberg [99]| Probout                            | It can achieve the balance between preserving the desirable properties of maxout units and improving their invariance properties. |
| Loss function        | Jin [100]  | Hinge loss                                        | It accelerates the training and improves the accuracy.                   |
|                      | ----       | Softmax Loss                                      | It is the combination of Multinomial Logistic.                           |
| Name          | Method                          | Description                                                                 |
|---------------|---------------------------------|-----------------------------------------------------------------------------|
| Liu [101]     | Large-Margin Softmax            | It performs better than the Softmax.                                         |
| Lin [102]     | Double Margin Loss              | It improves the training accuracy.                                          |
| Schroff [103] | Triplet Loss                    | Its object is to minimize the distance between the anchor and positive, and maximize the distance between the negative and the anchor. |
|               | Kullback-Leibler Divergence     | It is widely used as a measure of information loss in the objective function of various Autoencoders. |
| Tikhonov [104]| Regularization I²-norm          | It makes full use of the sparsity of weights to get a better optimization. |
|               | Regularization I₀-norm          | It makes the optimization easier.                                           |
| Hinton [105]  | Dropout                         | It is very effective in reducing overfitting.                               |
| Wang [106]    | Fast Dropout                    | It samples using Gaussian Approximation.                                    |
| Ba [107]      | Adaptive Dropout                | It reduces overfitting.                                                     |
| Tompson [108] | Spatial Dropout                 | It reduces overfitting and is very suitable for the training of a small dataset size. |
| Wan [109]     | Drop Connect                    | It is used on the convolutional layers and reduces overfitting easier than dropout used on the full connection layers. |
Table 2. Mathematical functions.

| Model                      | Function                                                                 | Interval  |
|----------------------------|--------------------------------------------------------------------------|-----------|
| Griewank                   | $f(x) = \frac{\sum_{i=1}^{d} x_i^2}{4000} - \prod_{i=1}^{d} \cos \frac{x_i}{\sqrt{i}} + 1$ | [-600, 600] |
| Levy                       | $f(x) = \sum_{i=1}^{d} \left( (x_i - 1)^2 \left[ 1 + \sin^2 \left( 3\pi x_i \right) \right] \right)$ | [-10, 10] |
| Sum of Different Powers    | $f(x) = \sum_{i=1}^{d} |x_i|^{i+1}$                                                                          | [-1, 1]  |
| Rotated Hyper-Ellipsoid    | $f(x) = \sum_{i=1}^{d} \sum_{j=1}^{d} x_{ij}^2$                           | [-5, 5]  |
| Sphere                     | $f(x) = \sum_{i=1}^{d} x_i^2$                                              | [-6, 6]  |
| Sum Squares                | $f(x) = \sum_{i=1}^{d} |x_i|^2$                                                                 | [-10, 10] |
| Trid                       | $f(x) = \sum_{i=1}^{d} (x_i - 1)^2 - \sum_{i=1}^{d-1} x_{i}x_{i+1}$        | [-4, 4]  |
| Dixon-Price                | $f(x) = \sum_{i=1}^{d} (x_i - 1)^2 - \sum_{i=1}^{d-1} i \left( 2x_{i+1}^2 - x_i \right)^2$ | [-10, 10] |
Table 3 Criteria for performance evaluation.

where STD stands for standard deviation, MSE (Mean Square Error) represents the departure of the metamodel from the real simulation model, and the variance captures the irregular of the problem.

| Criteria                                | Expression                                                                 |
|-----------------------------------------|---------------------------------------------------------------------------|
| Relative Average Absolute Error (RAAE)  | $\frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n \cdot \text{STD}}$                                         |
| Relative Maximum Absolute Error (RMAE)  | $\frac{\max(|y_i - \hat{y}_i|)}{\text{STD}}$                              |
| $R$ square ($R^2$)                      | $1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} = 1 - \frac{\text{MSE}}{\text{variance}}$ |
### Table 4 Comparisons of modeling cost.

| d   | Sample size of CNN | Sample size of a second order expansion of RBF-HDMR | Sample size of a full second order expansion of HDMR |
|-----|--------------------|-----------------------------------------------------|-----------------------------------------------------|
| 100 | 10,000             | 6,116                                               | 79,601                                               |
| 150 | 10,000             | 12,807                                              | 179,401                                              |
| 200 | 10,000             | 22,042                                              | 319,201                                              |
| 250 | 10,000             | 33,762                                              | 499,001                                              |
| 300 | 10,000             | 47,979                                              | 718,801                                              |
Table 5 Mean value of error for Griewank function over the interval [-600, 600] by CNN.

| Dimension | 100d | 144d | 256d | 324d | 784d |
|-----------|------|------|------|------|------|
| Error (%) | 0.92 | 0.97 | 1.24 | 1.21 | 1.22 |
Table 6 Mean value of performance criteria when dimension is 100d.

| Interval | Sample size | RAAE   | RMAE   | $R^2$  |
|----------|-------------|--------|--------|--------|
| [0, 5]   | 5000        | 9.15E-01 | 3.92E+00 | -2.93E-01 |
|          | 7500        | 5.39E-01 | 3.04E+00 | 5.36E-01  |
|          | 10000       | 3.17E-01 | 1.42E+00 | 8.42E-01  |
|          | 12500       | 3.16E-01 | 1.44E+00 | 8.43E-01  |
|          | 15000       | 3.17E-01 | 1.43E+00 | 8.42E-01  |
|          | 5000        | 4.35E-01 | 1.99E+00 | 7.03E-01  |
|          | 7500        | 3.95E-01 | 2.04E+00 | 7.54E-01  |
| [0, 10]  | 10000       | 3.58E-01 | 2.28E+00 | 7.98E-01  |
|          | 12500       | 3.32E-01 | 1.71E+00 | 8.25E-01  |
|          | 15000       | 3.33E-01 | 1.75E+00 | 8.23E-01  |
|          | 5000        | 4.09E-01 | 1.87E+00 | 7.32E-01  |
|          | 7500        | 3.88E-01 | 2.03E+00 | 7.64E-01  |
| [0, 100] | 10000       | 3.55E-01 | 1.85E+00 | 8.00E-01  |
|          | 12500       | 3.55E-01 | 1.97E+00 | 8.01E-01  |
|          | 15000       | 3.56E-01 | 1.95E+00 | 7.95E-01  |
|          | 5000        | 4.19E-01 | 2.27E+00 | 7.22E-01  |
|          | 7500        | 3.85E-01 | 1.81E+00 | 7.67E-01  |
| [0, 600] | 10000       | 3.52E-01 | 1.85E+00 | 8.05E-01  |
|          | 12500       | 3.52E-01 | 1.84E+00 | 8.06E-01  |
|          | 15000       | 3.51E-01 | 1.82E+00 | 8.06E-01  |
Table 7 Mean value of performance criteria when dimension is 324d.

| Interval | Sample size | RAAE    | RMAE    | $R^2$  |
|----------|-------------|---------|---------|--------|
| [0, 5]   | 5000        | 6.97E-01| 3.72E+00| 2.42E-01|
|          | 7500        | 5.95E-01| 2.80E+00| 4.51E-01|
|          | 10000       | 4.12E-01| 1.92E+00| 7.34E-01|
|          | 12500       | 4.13E-01| 1.95E+00| 7.32E-01|
|          | 15000       | 4.11E-01| 1.90E+00| 7.36E-01|
|          | 5000        | 5.70E-01| 2.43E+00| 4.91E-01|
|          | 7500        | 4.67E-01| 2.39E+00| 6.57E-01|
| [0, 10]  | 10000       | 4.08E-01| 2.13E+00| 7.38E-01|
|          | 12500       | 4.08E-01| 2.14E+00| 7.38E-01|
|          | 15000       | 4.08E-01| 2.13E+00| 7.38E-01|
|          | 5000        | 4.84E-01| 2.02E+00| 6.35E-01|
|          | 7500        | 4.10E-01| 2.06E+00| 7.36E-01|
| [0, 100] | 10000       | 3.84E-01| 1.82E+00| 7.69E-01|
|          | 12500       | 3.83E-01| 1.82E+00| 7.70E-01|
|          | 15000       | 3.84E-01| 1.81E+00| 7.70E-01|
|          | 5000        | 4.22E-01| 1.86E+00| 7.22E-01|
|          | 7500        | 3.80E-01| 2.02E+00| 7.70E-01|
| [0, 600] | 10000       | 3.78E-01| 2.01E+00| 7.74E-01|
|          | 12500       | 3.78E-01| 2.01E+00| 7.74E-01|
|          | 15000       | 3.77E-01| 2.01E+00| 7.75E-01|
Table 8 Mean value of performance criteria when dimension is 784d.

| Interval | Sample size | RAAE   | RMAE   | \( R^2 \) |
|----------|-------------|--------|--------|-----------|
| [0, 5]   | 5000        | 6.97E-01 | 3.72E+00 | 2.42E-01 |
|          | 7500        | 5.11E-02 | 3.01E+00 | 4.69E-01 |
|          | 10000       | 4.08E-01 | 2.18E+00 | 7.38E-01 |
|          | 12500       | 4.09E-01 | 2.18E+00 | 7.37E-01 |
|          | 15000       | 4.08E-01 | 2.18E+00 | 7.39E-01 |
|          | 5000        | 5.70E-01 | 2.43E+00 | 4.91E-01 |
|          | 7500        | 4.96E-01 | 2.24E+00 | 6.78E-01 |
| [0, 10]  | 10000       | 3.63E-01 | 1.95E+00 | 7.94E-01 |
|          | 12500       | 3.64E-01 | 1.90E+00 | 7.94E-01 |
|          | 15000       | 3.63E-01 | 1.95E+00 | 7.93E-01 |
|          | 5000        | 4.84E-01 | 2.02E+00 | 6.35E-01 |
|          | 7500        | 3.98E-01 | 2.06E+00 | 7.49E-01 |
| [0, 100] | 10000       | 3.66E-01 | 1.90E+00 | 7.90E-01 |
|          | 12500       | 3.66E-01 | 1.90E+00 | 7.90E-01 |
|          | 15000       | 3.66E-01 | 1.90E+00 | 7.90E-01 |
|          | 5000        | 4.22E-01 | 1.86E+00 | 7.22E-01 |
|          | 7500        | 3.97E-01 | 1.85E+00 | 7.46E-01 |
| [0, 600] | 10000       | 3.86E-01 | 1.82E+00 | 7.65E-01 |
|          | 12500       | 3.85E-01 | 1.81E+00 | 7.66E-01 |
|          | 15000       | 3.86E-01 | 1.83E+00 | 7.65E-01 |
Table 9 The performance criteria for trained CNN metamodel.

| RAAE   | RMAE   | $R^2$   |
|--------|--------|---------|
| 0.033447 | 0.52820 | 0.99776 |
Fig. 1. The development of CNN.
Fig. 2. The architecture of CNN with mixed pooling.
Fig. 3. Mean values of performance criteria for CNN and RBF-HDMR.

where CNN is marked in orange and RBF-HDMR is marked in blue.
Fig. 4. Griewank function over the interval [-600, 600] by KPLS.
Fig. 5. Performance for hundreds-dimensional problems by CNN metamodel.
Fig. 6. The control lattice for the tubular structure based on IGA.
Fig. 7. Constraints and load of the tubular structure.
Fig. 8. Von Mises stress distribution of the initial model.
Fig. 9. Training process of CNN metamodel.
Fig. 10. The comparison between optimization of CNN and IGA.

where the population of PSO is 50, and iteration step for CNN and IGA are all 200.
Fig. 11. The optimal shape and stress distribution of the optimal structure by CNN.