



HAL
open science

An AdaBoost Based - Deep Stochastic Configuration Network

Chenglong Zhang, Ding Shifei, Ling Ding

► **To cite this version:**

Chenglong Zhang, Ding Shifei, Ling Ding. An AdaBoost Based - Deep Stochastic Configuration Network. 12th International Conference on Intelligent Information Processing (IIP), May 2022, Qingdao, China. pp.3-14, 10.1007/978-3-031-03948-5_1 . hal-04178745

HAL Id: hal-04178745

<https://inria.hal.science/hal-04178745v1>

Submitted on 8 Aug 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

An AdaBoost Based- Deep Stochastic Configuration Network

Chenglong Zhang¹, Shifei Ding^{1,*} and Ling Ding²

¹ School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China

² Xuhai College, China University of Mining and Technology, Xuzhou 221008, China
dingsf@cumt.edu.cn

Abstract. Deep stochastic configuration network (DSCN) is an incremental learning method for large-scale data analysis and processing, which has the advantages of lower human intervention, higher learning efficiency and stronger generalization ability. For improving the stability of DSCN, a deep stochastic configuration network based on AdaBoost is proposed, termed as AdaBoost-DSCN. In our proposed model, the AdaBoost learning approach is adopted, the weights of the base models are adjusted adaptively according to the training results, then the base models are combined to generate a stronger model, which is beneficial to reduce the influence of random parameters on network performance of DSCN. Experimental results on complex function approximation problems and large-scale regression datasets show that AdaBoost-DSCN has higher regression accuracy for large-scale data regression analysis compared with DSCN, AdaBoost-SCN, SCN.

Keywords: deep stochastic configuration network, ensemble learning, neural network, AdaBoost

1 Introduction

Stochastic configuration network (SCN) proposed by Wang and Li in 2017, is an incremental model, its nodes in hidden layer can be increased gradually according to the preset conditions [1]. As the parameters of SCN are configured by stochastic configuration (SC) algorithm, and the network structure can be determined by supervision mechanism adaptively, SCN has a series of advantages of less human intervention, faster learning efficiency and better generalization performance. It has been extensively studied and applied in the areas of power big data processing [2], EEG signal analysis [3], fault diagnosis [4], online measurement problems [5, 6], robotic grasping recognition [7] and defect recognition [8].

Since SCN was proposed, it has been favored by domestic and foreign researchers for its lower human intervention, higher learning efficiency and stronger generalization ability [1]. In the aspects of network structure, SCN was expanded into 2-D stochastic configuration network (2DSCN) and deep stochastic configuration network (DSCN) respectively, showing the great potential in the field of image processing [9, 10].

However, as the training process of SCN mainly depends on the training samples, with the increase of nodes in the hidden layer, it is easy to over-fitting, which reduces the robustness and generalization of the network. Training multiple neural network models using distributed and ensemble learning strategies can improve the accuracy and generalization performance of SCN significantly improved. Negative correlation learning (NCL) and cooperative learning paradigm were applied to calculate the parameters of ensemble models respectively [11, 12]. Parallel stochastic configuration network (PSCN) which used different objective functions to train multiple sub-networks in parallel, and adopts fuzzy evidence theory to fuse the results [13]. Unlike the above model, stochastic configuration network ensembles with selective base models was introduced [14]. Otherwise, ensemble learning methods include Stacking, Bagging and AdaBoost were introduced into SCN for solving different problems [15-17].

DSCN, as the deep structure of SCN, has been proved the performance in image processing and complex function approximation problems [10]. Except for the prediction interval of crude oil carbon residue, DSCN is seldom used in the field of regression [18]. During to the construction process of DSCN, some ensemble method used in SCN are no longer suitable for DSCN, in the first place, we cannot use these methods, including negative correlation learning, cooperative learning paradigm etc. to calculate the output weights of ensemble DSCN model [11, 12,14]; in the second place, parallel [13], stacking [15], Bagging [16] and AdaBoost [17] were applied to solve different scenarios. AdaBoost method can adjust the weights of base learners according to the training errors of samples and improve the performance of the learners by weighting, which is sensitive to outliers and noise data. It can effectively overcome the over-fitting problem of the DSCN model [19], like the use of AdaBoost in SCN [17], a deep stochastic configuration network based on AdaBoost is proposed, named as AdaBoost-DSCN.

The rest parts of this paper are given as lists: Section 2 reviews the framework of deep stochastic configuration network. In section 3, the theory of AdaBoost-DSCN is presented in detail. Section 4 gives some experimental results on complex function approximation problems and large-scale regression datasets. Finally, some conclusions and future works are described in section 5.

2 Revisit of DSCN

Deep stochastic configuration network (DSCN) is the deep version of SCN, unlike other deep neural networks, it can start from one hidden layer with only one node, gradually increase the number of nodes and hidden layers to build the deep structure, in which input weights and biases can be configured through supervision mechanism and the output weights are calculated by using the least square method. As can be seen in Fig. 1, the network structure of DSCN is differs from the structure of other deep models. The number of hidden layers and its nodes are not fixed and it can be increased gradually according to the training errors [10].

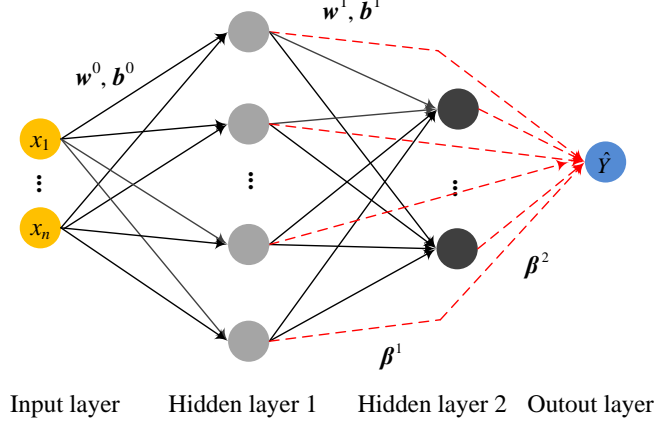


Fig. 1. The structure of stochastic configuration network [10]

The modeling learning process of DSCN is described as lists [10]:

Given a set of training data, $X = \{x_1, x_2, \dots, x_N\}$ and $Y = \{y_1, y_2, \dots, y_N\}$ represent the input data and corresponding label. $x_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,d}\} \in R^d$, d indicates the feature dimension; $y_i = \{y_{i,1}, y_{i,2}, \dots, y_{i,m}\} \in R^m$, m expresses the label dimension; $i = 1, 2, \dots, N$, N is the number of training samples.

It is supposed that $L_n - 1$ nodes in n th layer of DSCN have been configured, compute the training error vector by using Eq. (1), then judge whether meet the preset conditions $n = n_{\max}, L_n = L_{n_{\max}}$ or $\|e\|_F^2 \leq \varepsilon$, n_{\max} is defined as the maximum number of layers; $L_{n_{\max}}$ denotes the maximum number of nodes in n th layer; ε represents the preset error.

$$e_{L_n-1}^n = e_{L_n-1}^n(X^{n-1}) = [e_{L_n-1,1}^n(X^{n-1}), e_{L_n-1,2}^n(X^{n-1}), \dots, e_{L_n-1,m}^n(X^{n-1})]^T \quad (1)$$

Assign initial input weight $w_{L_n}^n$ and bias $b_{L_n}^n$ of node L_n in n th layer by using to Eq. (2) and Eq. (3):

$$w_{L_n}^n = \lambda \times [2 \times \text{rand}(N, T_{\max}) - 1] \quad (2)$$

$$b_{L_n}^n = \lambda \times [2 \times \text{rand}(1, T_{\max}) - 1] \quad (3)$$

where λ represent the scale factor of input weight and bias; T_{\max} expresses the maximum number of candidate nodes.

The candidate $w_{L_n}^{n*}$ and $b_{L_n}^{n*}$ with the largest value of $\xi_{L_n}^n = \sum_{j=1}^m \xi_{L_n,j}^n \geq 0$ are determined as the input weight and bias of node L_n :

$$h_{L_n}^n = \phi[(w_{L_n-1}^n)^T X^{n-1} + b_{L_n-1}^n]^T (X^0 = X, X^{n-1} = H_{L_n-2}^{n-2}, n \geq 2) \quad (4)$$

$$\xi_{L_n, j}^n = \frac{\langle e_{L_n-1, j}^n, h_{L_n}^n \rangle^2}{\langle h_{L_n}^n, h_{L_n}^n \rangle} - (1-r-\mu_{L_n}) \langle e_{L_n-1, j}^n, e_{L_n-1, j}^n \rangle \quad (5)$$

where $h_{L_n}^n$ is denoted as the output of node L_n ; $j=1,2,\dots,m$; $r \in (0,1)$; $\{\mu_{L_n}\}$, in which $\mu_{L_n} \leq 1-r$ and $\lim_{L_n \rightarrow +\infty} \mu_{L_n} = 0$, represents a nonnegative real number sequence.

The output weight matrix β can be evaluated according to Eq. (6):

$$\beta^* = \arg \min_{\beta} \|H\beta - Y\|_F^2 = H^+ Y \quad (6)$$

where $H = [H_{L_1}^1, H_{L_2}^2, \dots, H_{L_n}^n]$, among them $H_{L_n}^n = [h_{L_n}^n, h_{L_n}^n, \dots, h_{L_n}^n]$; H^+ is denoted as the Moore-Penrose inverse of H .

The output \hat{Y} of DSCN is calculated by using Eq. (7):

$$\hat{Y} = H\beta \quad (7)$$

3 AdaBoost-DSCN

AdaBoost [19, 20] is a classical ensemble learning algorithm which is widely used for the ensemble of neural networks. It can avoid over-fitting problem and improve the generalization ability of neural network models for solving classification and regression problems by combining multiple weak learners into a strong learner [21, 22].

In the process of incremental learning modeling of DSCN, the input weights and biases of hidden layer nodes are mainly dependent on the scale factor λ . However, for different large-scale training datasets, the performance of DSCN is easily limited by the setting range of the scale factor λ and affected by the randomness of the allocation of input weights and biases. To improve the stability and avoid over-fitting problem of DSCN, an AdaBoost-based deep stochastic configuration network (AdaBoost-DSCN) is proposed by training multiple DSCN-based learner models. Firstly, through training multiple DSCN-based learner models, iterative training is carried out based on AdaBoost algorithm; Secondly, the weighted majority voting method is adopted to reduce the weights of the training samples with better results and improve the weights of the training samples with poor results; Finally, a strong learner model is obtained by weighted combination according to the training results of each learner.

Therefore, the AdaBoost-DSCN model is described as follows [23, 24]:

Given a set of training data, $X = \{x_1, x_2, \dots, x_N\}$ indicates the input data, $x_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,d}\} \in R^d$, d represents the feature dimension; $Y = \{y_1, y_2, \dots, y_N\}$ is denoted as the corresponding label, $y_i = \{y_{i,1}, y_{i,2}, \dots, y_{i,m}\} \in R^m$, m is the label dimension; among them $i=1,2,\dots,N$, N is the number of training samples.

Step 1: Initialize the distribution weight of training samples through Eq. (8):

$$\alpha_k = [\alpha_{k,1}, \alpha_{k,2}, \dots, \alpha_{k,N}], \quad \alpha_{k,i} = \frac{1}{N} \quad (8)$$

where $k=1,2,\dots,K$, K is the number of base learners; $i=1,2,\dots,N$, N is the number of training samples.

Step 2: Training the K DSCN-based learners iteratively, and calculate the errors of each sample according to Eq. (9):

$$\sigma_k = f_k - Y \quad (9)$$

where $\sigma_k = [\sigma_{k,1}, \sigma_{k,2}, \dots, \sigma_{k,N}]$, f_k is the regression results of k th base learner, $k=1,2,\dots,K$.

Step 3: According to the sample errors of Eq. (9), update the distribution weight of training samples by using Eq. (10):

$$\alpha_{k,i} = \begin{cases} \alpha_{k,i} \times c & \text{if } |\sigma_{k,i}| > \mu \\ \alpha_{k,i} & \text{else} \end{cases} \quad (10)$$

where c is the control factor of the distribution weight; μ is the threshold of sample error which is used for determining the base learner outliers.

Step 4: Calculate the weight of DSCN-based learners according to Eq. (11):

$$\alpha_k = (0.5/e^{s_k}) / \sum_{k=1}^K (0.5/e^{s_k}) \quad (11)$$

where s_k represents the summation of regression errors of training samples, where $s_k = s_k + \alpha_{k,i}$ when $|\sigma_{k,i}| > \mu$.

Step 5: Construct the AdaBoost-DSCN model according to the weights of base learners as Eq. (12):

$$f = \sum_{k=1}^K f_k \alpha_k \quad (12)$$

where f is denoted as the output of AdaBoost-DSCN model.

4 Experiments

All the experimental codes of the paper are programed based on MATLAB 2019b, which running on a PC with Intel (R) Core (TM) i7-9750H 2.60 GHz CPU, NVIDIA GPU GTX1650 and 64 GB RAM.

4.1 Benchmark datasets

To illustrate the performance of AdaBoost-DSCN, two complex function approximation problems (Eq. (13) and Eq. (14)) and four large-scale regression datasets (Table.

1) of KEEL (Knowledge Extraction based on Evolutionary Learning, <http://www.keel.es/>) are selected as benchmark datasets. The descriptions of these benchmark datasets are introduced as lists.

Real-valued function $f_1(x)$ [25]:

$$f_1(x) = 0.2e^{-(10x-4)^2} + 0.5e^{-(80x-40)^2} + 0.3e^{-(80x-20)^2} \quad (13)$$

The training dataset has 1000 samples while the test dataset is consisted of 300 samples, all the training samples and test samples are uniformly distributed in $[0, 1]$ respectively.

Real-valued function $f_2(x)$ [26]:

$$f_2(x) = 0.8e^{-0.2x} \sin(10x) \quad (14)$$

It is the same as $f_1(x)$, the training samples and test samples are generated from a regularly spaced grid over $[0, 5]$, and the training dataset and test dataset are consisted of 1000 and 300 samples respectively.

Table 1. The attributes of four large-scale regression datasets.

Datasets	Input Features	Instances
MV	10	40768
Compactiv	21	8192
Pole	26	14998
Ailerons	40	13750

To reduce the influence of different feature ranges on the model, the input and output vectors of four large-scale regression datasets from KEEL are normalized into $[0, 1]$, in which 75% of the samples are selected as the training dataset randomly while the rest are determined as the test dataset.

4.2 Evaluation indicator and parameters setting

4.2.1 Evaluation indicator

To verify the regression accuracy of AdaBoost-DSCN, SCN [1], DSCN [10], AdaBoost-SCN [17] are chosen as the compared models, then we run each model 10 times, root mean square error (RMSE) (Eq. (15)) and its standard deviations of all the results are determined for ensuring the fairness of the experiments.

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (15)$$

where y_i is the actual value of sample i ; \hat{y}_i denotes the regression result of sample i ; N indicates the number of samples.

4.2.2 Parameters setting

In the experiments, the parameters of related models are set as follows: all the models adopt the sigmoidal function (Eq. (16)) as activation function, the nodes in the hidden

layers of all the models are set as 25, 50 respectively, some other parameters of each model are given especially.

$$S(x) = \frac{1}{1 + e^{-x}} \quad (16)$$

SCN: the rand λ of SCN are chosen from set $\{0.9, 0.99, 0.9999, 0.99999, 0.999999\}$ and $\{0.5, 1, 5, 10, 30, 50, 100, 150, 200, 250\}$ adaptively; the maximum number T_{\max} of candidate nodes is set to 200; the tolerance error ε is set as 0.001.

DSCN: the maximum number of layers n_{\max} is set to 2; the r and λ of SCN are chosen from set $\{0.9, 0.99, 0.999, 0.9999, 0.99999, 0.999999\}$ and $\{0.5, 1, 5, 10, 30, 50, 100, 150, 200, 250\}$ adaptively; the maximum number T_{\max} of candidate nodes is set to 200; the tolerance error ε is set as 0.001.

The parameters of AdaBoost-SCN and AdaBoost-DSCN are similar as SCN and DSCN respectively, except for the number of base learner K is set to 5, the weight control factor c is set to 1.1, the threshold of sample point error μ is set to 0.001.

4.3 Results and discussion

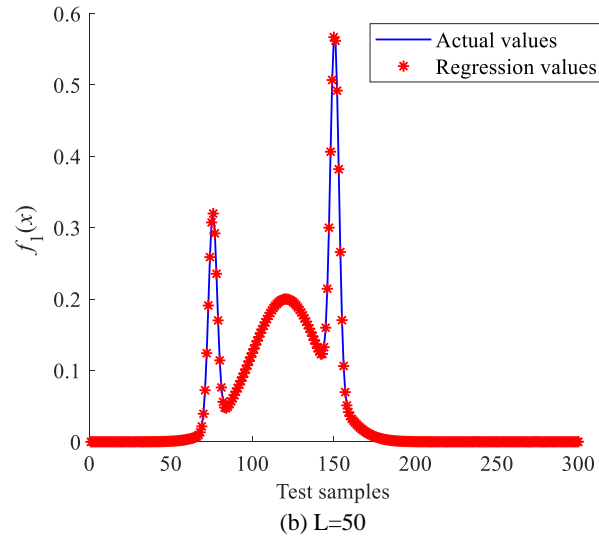
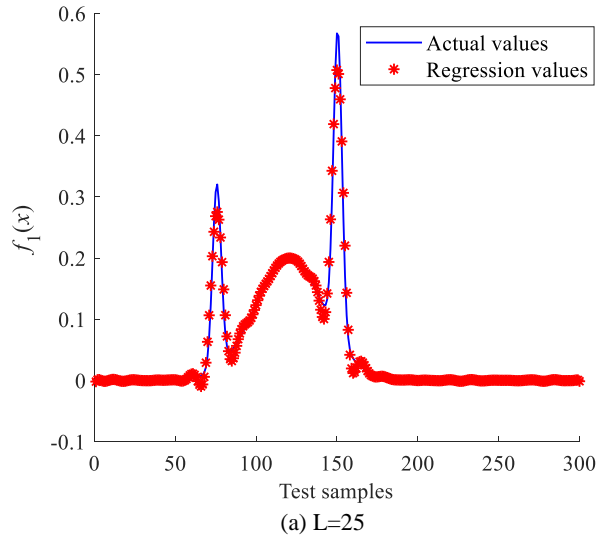
The average test results of AdaBoost-DSCN, SCN, DSCN and AdaBoost-SCN on function approximation problems are given in Table. 2 and Table. 3, in which L represents the maximum number of nodes in each hidden layer, DSCN and AdaBoost-DSCN has 2 hidden layers for solving these problems. As can be seen from Table. 2, it obviously that AdaBoost-DSCN outperform other models in terms of both average test results on two complex function approximation problems. With the increase of the maximum number of nodes, the test results of DSCN and AdaBoost-DSCN are enhanced greatly, which shows that AdaBoost-DSCN and DSCN have better fitting effect than SCN and AdaBoost-SCN for function approximation problems, the AdaBoost-DSCN and AdaBoost-SCN have higher regression accuracy than DSCN and SCN, verifies the effectiveness of AdaBoost ensemble learning. At the same time, the fitting curves of AdaBoost-DSCN model to function approximation problems are shown in Fig. 2.

Table 2. Performance comparison of four models on $f_1(x)$.

Models	Test results (RMSE)	
	$L=25$	$L=50$
SCN	3.8546e-02 \pm 3.0835e-03	8.9775e-03 \pm 3.4369e-03
DSCN	1.5178e-02 \pm 3.1029e-03	4.3556e-04 \pm 2.7920e-04
AdaBoost-SCN	3.7056e-02 \pm 1.7307e-03	7.6905e-03 \pm 1.4554e-03
AdaBoost-DSCN	1.1990e-02 \pm 2.1957e-03	1.8511e-04 \pm 8.2268e-05

Table 3. Performance comparison of four models on $f_2(x)$.

Models	Test results (RMSE)	
	$L=25$	$L=50$
SCN	$6.2263e-02 \pm 3.2845e-02$	$1.3861e-03 \pm 6.9197e-04$
DSCN	$6.8888e-03 \pm 6.3440e-03$	$5.2601e-05 \pm 3.0592e-05$
AdaBoost-SCN	$4.2098e-02 \pm 1.4116e-02$	$6.2471e-04 \pm 1.3571e-04$
AdaBoost-DSCN	$2.0983e-03 \pm 1.5434e-03$	$2.5195e-05 \pm 4.9656e-06$



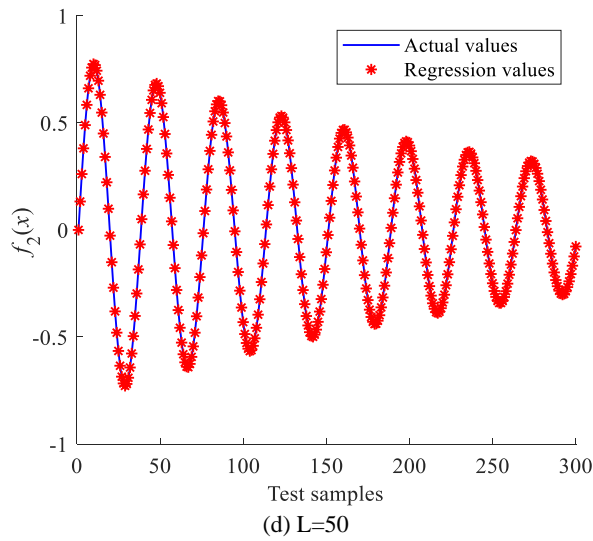
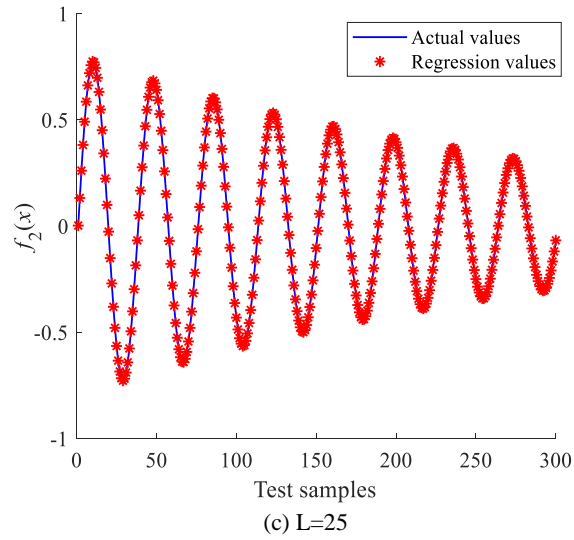


Fig. 2. Fitted curves of AdaBoost-DSCN on function approximation problems

Table. 4 - Table. 7 demonstrates the average test results of AdaBoost-DSCN, SCN, DSCN and AdaBoost-SCN on large-scale regression datasets, among them L indicates the maximum number of hidden layer nodes, meanwhile DSCN and AdaBoost-DSCN has 2 hidden layers for solving these problems.

Table 4. Performance comparison of MV dataset.

Models	Test results (RMSE)	
	$L=25$	$L=50$
SCN	4.4904e-02±4.7017e-03	2.2737e-02±1.6510e-03
DSCN	3.0270e-02±2.4272e-03	1.5521e-02±1.0531e-03
AdaBoost-SCN	3.6555e-02±1.6602e-03	1.8546e-02±3.5122e-04
AdaBoost-DSCN	2.3075e-02±1.4101e-03	1.2987e-02±2.6917e-04

Table 5. Performance comparison of Compactiv dataset.

Models	Test results (RMSE)	
	$L=25$	$L=50$
SCN	9.3403e-02±6.1868e-03	4.3051e-02±1.5580e-03
DSCN	4.0368e-02±5.0512e-03	2.8340e-02±8.0883e-04
AdaBoost-SCN	7.7458e-02±4.1736e-03	3.3792e-02±1.0012e-03
AdaBoost-DSCN	3.2994e-02±1.4950e-03	2.6748e-02±6.9538e-04

Table 6. Performance comparison of Pole dataset.

Models	Test results (RMSE)	
	$L=25$	$L=50$
SCN	2.8158e-01±7.9057e-03	2.3198e-01±8.0734e-03
DSCN	2.4445e-01±4.2805e-03	2.0740e-01±7.4167e-03
AdaBoost-SCN	2.7025e-01±4.0141e-03	2.2043e-01±1.8525e-03
AdaBoost-DSCN	2.3317e-01±3.6195e-03	1.9688e-01±3.6832e-03

Table 7. Performance comparison of Elevators dataset.

Models	Test results (RMSE)	
	$L=25$	$L=50$
SCN	3.8277e-02±1.2402e-03	3.6015e-02±1.0447e-03
DSCN	3.6754e-02±1.3041e-03	3.9248e-02±5.9949e-03
AdaBoost-SCN	3.7417e-02±1.1559e-03	3.5584e-02±5.1169e-04
AdaBoost-DSCN	3.5959e-02±6.0075e-04	3.4545e-02±1.1014e-03

As we can see from Table. 4 -Table. 7, AdaBoost-DSCN model has the smallest test errors on the large-scale regression data set of MV, Compactiv, Pole and Elevators

compared with SCN, DSCN and AdaBoost-SCN model. Compared with DSCN and SCN, AdaBoost-DSCN and AdaBoost-SCN can improve the regression accuracy furtherly, which indicates that DSCN has stronger capability of large-scale data regression compared with SCN, and AdaBoost ensemble learning approach can be used for solving the over-fitting problem of DSCN.

In conclusion, with the increase of data dimension and data scale, AdaBoost-based deep stochastic configuration network improves the regression accuracy and stability of DSCN, which is superior to SCN, DSCN and AdaBoost-SCN models.

5 Conclusion

DSCN can allocate weights and biases of nodes and construct deep network structure adaptively through supervision mechanism, which has faster learning efficiency and higher learning accuracy. To avoid over-fitting problem of the DSCN for solving regression problems, AdaBoost ensemble learning strategy is merged into DSCN, then deep stochastic configuration network based on AdaBoost is proposed, termed as AdaBoost-DSCN. Experimental results on the complex function approximation problems and some large-scale regression datasets demonstrate that AdaBoost-DSCN can improve the training accuracy of DSCN and overcome the problems of model instability and over-fitting which caused by random configuration of parameters and strong dependence on training samples. In the future, image classification is one of the important research fields of machine learning, AdaBoost-DSCN will be used to solve this kind of problem to obtain higher classification accuracy.

6 Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No.61976216 and No.61672522.

References

1. Wang, D., Li, M.: Stochastic configuration networks: fundamentals and algorithms. *IEEE Transactions on Cybernetics* 47(10), 3466–3479 (2017)
2. Huang, C., Huang, Q., Wang, D.: Stochastic configuration networks based adaptive storage replica management for power big data processing. *IEEE Transactions on Industrial Informatics* 16(1), 373–383 (2020)
3. Pang, L., Guo, L., Zhang, J.: Subject-specific mental workload classification using EEG and stochastic configuration network (SCN). *Biomedical Signal Processing and Control* 68, 102711 (2021)
4. Liu, J., Hao, R., Zhang, T.: Vibration fault diagnosis based on stochastic configuration neural networks. *Neurocomputing* 434, 98–125 (2021)
5. Wang, W., Wang, D.: Prediction of component concentrations in sodium aluminate liquor using stochastic configuration networks. *Neural Computing and Applications* 32(17), 13625–13638 (2020)

6. Wang, W., Jia, Y., Yu, W.: On-line ammonia nitrogen measurement using generalized additive model and stochastic configuration networks. *Measurement* 170, 108743 (2021)
7. Pan, J., Luan, F., Gao, Y.: FPGA-based implementation of stochastic configuration network for robotic grasping recognition. *IEEE Access* 8, 139966–139973 (2020)
8. Zhao, J., Hu, T., Zheng, R.: Defect recognition in concrete ultrasonic detection based on wavelet packet transform and stochastic configuration networks. *IEEE Access* 9, 9284–9295 (2021)
9. Li, M., Wang, D.: 2-D stochastic configuration networks for image data analytics. *IEEE Transactions on Cybernetics* 51(1), 359–372 (2021)
10. Wang, D., Li, M.: Deep stochastic configuration networks with universal approximation property. In: 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE, Rio de Janeiro Brazil (2018)
11. Wang, D., Cui, C.: Stochastic configuration networks ensemble with heterogeneous features for large-scale data analytics. *Information Sciences* 417, 55–71 (2017)
12. Ai, W., Wang, D.: Distributed stochastic configuration networks with cooperative learning paradigm. *Information Sciences* 540, 1–16 (2020)
13. Zhang, C., Ding, S., Zhang, J.: Parallel stochastic configuration networks for large-scale data regression. *Applied Soft Computing* 103, 107143 (2021)
14. Huang, C., Li, M., Wang, D.: Stochastic configuration network ensembles with selective base models. *Neural Networks* 137, 106–118 (2021)
15. Pratama, M., Wang, D.: Deep stacked stochastic configuration networks for lifelong learning of non-stationary data streams. *Information Sciences* 495, 150–174 (2019)
16. Li, K., Wang, W., Wang, Y.: Application of ensemble stochastic configuration network in aquaculture water quality monitoring. *Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE)* 36(4), 220–226 (2020)
17. Qu, H., Feng, T., Wang, Y.: AdaBoost SCN algorithm for optics fiber vibration signal recognition. *Applied Optics* 58(21), 5612–5623 (2019)
18. J. Lu, J. Ding, Construction of prediction intervals for carbon residual of crude oil based on deep stochastic configuration networks, *Information Sciences*. 486 (2019) 119–132.
19. Hansen, L. K., Salamon, P.: Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12(10), 993–1001 (1990)
20. Freund, Y., Schapire, R. E.: A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences* 55(1), 119–139 (1997)
21. Opitz, D. W., Maclin, R. F.: An empirical evaluation of bagging and boosting for artificial neural networks. In: *Proceedings of International Conference on Neural Networks (ICNN'97)*, pp. 1401–1405, IEEE, Houston, USA (1997)
22. Liu, L., Hua, Y., Zhao, Q.: Blind image quality assessment by relative gradient statistics and AdaBoosting neural network. *Signal Processing-Image Communication* 40, 1–15 (2016)
23. Luo, Y., Wang, B.: Prediction of negative conversion days of childhood nephrotic syndrome based on PCA and BP-AdaBoost neural network. *IEEE Access* 7, 151579–151586 (2019)
24. Li, S., Wang, J., Liu, B.: Prediction of market demand based on AdaBoost_BP neural network. In: 2013 International Conference on Computer Sciences and Applications, pp. 305–308, IEEE, Wuhan, China (2013)
25. Tyukin, I. Y., Prokhorov, D. V.: Feasibility of random basis function approximators for modeling and control. In: 2009 IEEE Control Applications, (CCA) & Intelligent Control, (ISIC), pp. 1391–1396, IEEE, St. Petersburg, Russia (2009)

26. Li, M., Wang, D.: Insights into randomized algorithms for neural networks: practical issues and common pitfalls. *Information Sciences* 382, 170–178 (2017)