Hierarchical Feature Selection for Random Projection

Qi Wang, *Senior Member, IEEE*, Jia Wan, Feiping Nie, Bo Liu, Chenggang Yan, and Xuelong Li, *Fellow, IEEE*

***Abstract*— Random projection is a popular machine learning algorithm, which can be implemented by neural networks and trained in a very efficient manner. However, the number of features should be large enough when applied to a rather large-scale data set, which results in slow speed in testing procedure and more storage space under some circumstances. Furthermore, some of the features are redundant and even noisy since they are randomly generated, so the performance may be affected by these features. To remedy these problems, an effective feature selection method is introduced to select useful features hierarchically. Specifically, a novel criterion is proposed to select useful neurons for neural networks, which establishes a new way for network architecture design. The testing time and accuracy of the proposed method are improved compared with traditional methods and some variations on both classification and regression tasks. Extensive experiments confirm the effectiveness of the proposed method.**

***Index Terms*— Extreme learning machine (ELM), feature selection, neural networks, random projection.**

I. Introduction

Random projection algorithms have been utilized in many applications such as face recognition [1] and are confirmed to be efficient concerning with the training procedure. Neural networks are very popular in solving computer vision problems, such as intelligent vehicles [2], text detection [3], [4], motion estimation [5], [6], and image understanding [7]. One of the typical random projections implemented by the single-layer neural networks with randomly generated neurons is extreme learning machine (ELM) [8], which can be efficiently trained with a closed-form solution. However, more features are required when it is applied to rather large data sets since the features are randomly generated.

Generally, original features should be projected to a high-dimensional space to ensure the performance of the random projection, which will largely increase the computational cost. In order to select useful features, several algorithms are proposed to prune redundant features [9], [10]. Some of them are faster than the original algorithm in training, but can only be applied to the classification problems. Others can be used in both classification and regression tasks, but the training is complex and inefficient.

In this paper, an effective feature selection algorithm called feature selection for random projection (FSRP) is proposed to reduce the computational cost for both classification and regression tasks. For this algorithm, a feature selection method is presented in which a very simple yet efficient criterion is utilized to determine the usefulness of features. To further accelerate the selection of features, a hierarchical selection scheme is also presented. Through our experiments, this method can achieve superior performance than the original algorithms and some widely used classification methods with faster testing speed. And it can be used in both classification and regression tasks. The empirical idea of the proposed algorithm is shown in Fig. 1.

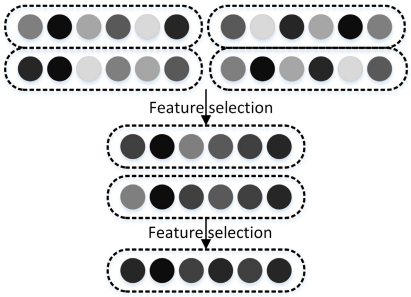


Fig. 1. Pipeline of the proposed hierarchical feature selection method. In this figure, each node represents a feature. The feature pool, which contains 24 features, is split into four groups. For each group, half of the features are selected each time. Then, a new feature pool, which contains 12 features, is generated. Finally, a compact network contains only six features are constructed. Note that, darker nodes indicate the features have higher weights.

The contributions of this paper are summarized as follows.

1) A simple yet efficient criterion is proposed to select useful features for neural networks. The criterion used for ranking the usefulness of features, which is based on the value of output weights is easy to calculate and effective to eliminate redundant and noisy features. Our method establishes a new way for network architecture design and is of inspiring contribution to this field.
2) A hierarchical selection scheme is proposed to choose useful features in a very efficient manner. To accelerate the selection of features on large-scale data sets, useful features are selected hierarchically, which avoids the selection from the whole feature pool. Therefore, the large matrix reversion is evaded, which largely accelerates the selection procedure.
3) Extensive experiments are performed on a variety of data sets including both classification and regression tasks and confirm the effectiveness and efficiency of the proposed method.

II. Related Works

In this section, we briefly review the compression algorithms based on the single-layer neural network with random generated features, which can be divided into two categories.

Manuscript received March 7, 2018; revised June 5, 2018 and August 3, 2018; accepted September 2, 2018. Date of publication September 27, 2018; date of current version April 16, 2019. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1107403, in part by the National Natural Science Foundation of China under Grant 61773316, in part by the Natural Science Foundation of Shaanxi Province under Grant 2018KJXX-024, in part by the Fundamental Research Funds for the Central Universities under Grant 3102017AX010, and in part by the Open Research Fund of Key Laboratory of Spectral Imaging Technology, Chinese Academy of Sciences. *(Corresponding author: Feiping Nie.)*

Q. Wang is with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China, also with the Center for OPTical IMagery Analysis and Learning, Northwestern Polytechnical University, Xi'an 710072, China, and also with the Unmanned System Research Institute, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: crabwq@gmail.com).

J. Wan and F. Nie are with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China, and also with the Center for OPTical IMagery Analysis and Learning, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: jiawan1998@gmail.com; feipingnie@gmail.com).

B. Liu is with the Department of Computer Science and Software Engineering, Auburn University, Auburn, AL 36849 USA (e-mail: boliu@auburn.edu).

C. Yan is with the Institute of Information and Control, Hangzhou Dianzi University, Hangzhou 310018, China (e-mail: cgyan@hdu.edu.cn).

X. Li is with the School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an 710072, China (e-mail: xuelong_li@nwpu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2018.2868836

The first type of algorithms performs network compression before or during the learning of features (i.e., matrix reversion calculation) which is faster in training than the original algorithm since the matrix reversion is performed on a smaller matrix. Huang and Chen [11] propose to incrementally add features from a candidate pool during the training process in which the feature which results in larger error decreasing will be appended to the existing features. Yang *et al.* [12] propose an algorithm, in which the relationship between a newly added feature and residual error is presented, so that the weight of the new feature can be calculated efficiently. Rong *et al.* [13] utilize statistical methods as the measurements of the relevance between features and class labels, in which the relevant features are removed to achieve compact expressions and robust performance. Another pruned method [14] is proposed, which utilizes the defined sensitivities to determine the necessary features. To remedy overfitting problem and generate compact network, Luo *et al.* [15] propose to learn sparse weights by Bayesian inference, in which the features with 0 weight can be pruned.

Another type of methods are proposed to select useful features based on the classification or regression results of the original features, which means that the training time is longer than the original algorithm. Miche *et al.* [16] utilize multiresponse sparse regression and leave-one-out validation to determine the usefulness of features so the algorithm can generate very compact network than the original features with comparable performance. Luo *et al.* [17] propose an adaptive reconstruction graph for feature selection. Wang *et al.* [18] propose a neighborhood discrimination index to utilize neighborhood relations. Sun *et al.* [19] propose a generative model for unsupervised learning. Kowalski and Kusy [20] propose to utilize local sensitivity analysis and to simplify network structure. Zhu *et al.* [21] propose to combine local and global information for feature selection. Shang *et al.* [22] propose a feature selection algorithm to exploit information in the feature space. Wang *et al.* [23] propose to rank and select significant features based on the partial orthogonal decomposition through recursive orthogonal least squares method.

III. Single-Layer Neural Network

Single-layer neural network with randomly generated neurons is a very fast algorithm since the learning of parameters does not need iterative optimization. The learning of the algorithm is rather simple. First of all, the weights and bias of input weights of the hidden layer are randomly assigned. Then, the output weights of the network are calculated to minimize the mean squared error (MSE) between the ground truth and the prediction.

Formally, in a single hidden layer neural network with N nodes, the output label y with respect to a d dimensional input vector x is denoted as

$$y = \sum_{i=1}^{N} \beta_i h_i(x) \tag{1}$$

where β_i is the weight of the ith node, and $h(x)$ indicates the nonlinear feature mapping, which can be written as

$$h_i(x) = \theta(a_i x + b_i) \tag{2}$$

where a_i and b_i are randomly assigned parameters and θ can be any nonlinear mapping functions such as sigmoid function, Gaussian function, or cosine function.

In this model, the only unknown parameter is β which can be learned by minimizing the MSE between the ground truth and the prediction

$$\beta = \min_{\beta} \|H\beta - G\|^2 \tag{3}$$

where $\|\cdot\|$ denotes the Frobenius norm and H is the output of hidden layer with randomly assigned parameters

$$H = \begin{bmatrix} h(x_1) \\ h(x_2) \\ \vdots \\ h(x_d) \end{bmatrix} = \begin{bmatrix} h_1(x_1) & h_2(x_1) & \dots & h_N(x_1) \\ h_1(x_2) & h_2(x_2) & \dots & h_N(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(x_d) & h_2(x_d) & \dots & h_N(x_d) \end{bmatrix} \tag{4}$$

and G is the ground truth matrix. If the algorithm is performed on a regression problem where the output target is a real value, G could be a vector

$$G = [g_1, g_2, \cdots, g_n]^\top. \tag{5}$$

Otherwise G is a matrix representing category labels in classification problems.

The optimization of (3) is very efficient since it has a closed-form solution

$$\beta^* = H^\dagger T \tag{6}$$

where $H^\dagger$ is defined in (4) and T is the target matrix.

Since it has no iteration steps, the optimization of the algorithm is very efficient especially when performed on a small data set in which the number and dimension of training samples are small. Under this condition, a small number of hidden nodes (features) are necessary to obtain good performance so that the matrix reversion can be easily calculated. However, a large amount of hidden nodes are usually required to guarantee the performance on large-scale data sets, because if the matrix reversion is performed on a large matrix, the calculation is often time-consuming.

IV. Hierarchical Feature Selection

In order to alleviate redundant features and avoid large matrix reversion, in this paper, a very simple yet effective feature selection method is proposed to hierarchically choose significant features from a large amount of randomly generated features. Particularly, the first step of the proposed method is to generate features randomly. Then, the features are split into several groups. For each group, the original algorithm is performed to learn the output weights, which can be utilized to evaluate the importance of features. Specifically, larger weight indicates the feature is more important, thus the features can be ranked based on their weights. After the ranking and selection, features from different groups are aggregated together. The selection procedure is performed several times hierarchically, and a very compact yet effective network can be generated from numerous features in a very efficient manner.

Since the input weight of hidden neurons is randomly assigned, it can be seen as random projection in which many redundant features might be yielded during the training process when a good performance is achieved. To relieve this problem, a simple criterion for feature selection is proposed to rank and choose features based on the output weight, which can be utilized to evaluate the importance of features. Particularly, after the learning of the original algorithm, the output weight β is obtained by calculating (6). As shown in the experiments, the features with larger absolute value have higher probability to generate better performance. So, the output weights are sorted in descending order. After the importance of features is ranked, a subset of features can be selected through index I. Proved by experiments, this method is very effective to prune redundant and noisy features and generate better performance than the original algorithm. However, the training time is increased especially when the feature pool is large as the calculation of output weight

Algorithm 1 Hierarchical FSRP

Require: $X \in \Re^{n\times d}$: the training features; $Y \in \Re^{n\times 1}$: the training labels; S: the size of feature pool; s: the size of each group; r: the ratio of selected features from each group

// for each node

for $i = 1, 2, ..., S$ **do**

Randomly assign parameters $a^{(i)}$, $b^{(i)}$

end for

//Suppose the hierarchical scheme has three layers

for $i = 1, 2, 3$ **do**

// Split features in pool into groups with size s

for each group **do**

Compute matrix H through (2)

Calculate β_j through (6)

Rank features' importance

Select features based on r and the ranking result

end for

Merge all selected features into a new feature pool

end for

Compute β_f of the final selected features by (6)

Ensure: Final selected features and the corresponding β_f

needs to solve a larger matrix inversion problem than the original algorithm.

To accelerate the selection of features by avoiding large matrix inversion, a hierarchical selection strategy is proposed. First of all, numerous features are randomly generated to serve as the feature pool. Then, features in the pool are split into several groups on which the feature selection procedure is performed. To obtain more compact network, the same feature selection can be conducted on the selected features which forms a hierarchical scheme. Suppose d features are selected from a feature pool with n features and the feature pool is divided into k groups. The computation complexity is reduced from $O(n^{2.373})$ to $O(n^{2.373}/k^{1.373})$.

Although there are some parameters which will affect the performance and efficiency of the proposed method, they are not very sensitive to different data sets. In all experiments, some parameters can be used for both classification and regression tasks over different data sets. The parameters are summarized as follows and its selection is investigated in experiments.

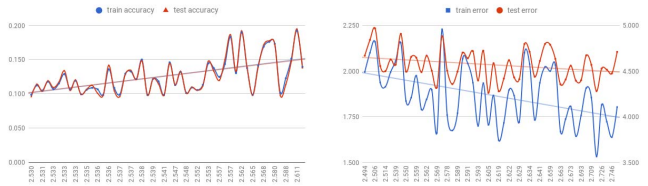
1) The first parameter that we should determine is the ratio of remained features in the selection procedure. Large ratio tends to yield better performance but the training speed is limited, while small ratio can be trained more efficiently. A proper ratio can ensure good performance and fast training speed at the same time.
2) The final size of selected features is a factor that affects performance and the testing speed of the algorithm. A large number will yield better performance; while a small number will be faster in training and testing.
3) Based on the ratio in feature selection procedure and the final remained number of features, the size of feature pool can be determined by the number of layers in a hierarchical selection scheme. The more the number of layers, the larger the feature pool. A large feature pool can produce better performance at most of the time. However, the training speed will increase a lot if hierarchical selection procedure with more layers is performed during training.

The proposed hierarchical feature selection method is summarized in Algorithm 1.

TABLE I

DETAILED INFORMATION OF EVALUATED DATA SETS FOR CLASSIFICATION

	size of dataset	dimension
dig	1797	64
msra	1799	25
palm	2000	256
tdt10	653	36771
text	1946	7511
usps	9298	256



(a) (b)

Fig. 2. Relationship between output weights and the performance of single-layer neural networks. In this figure, the horizontal axis is the values of the output weight in ascending order. The vertical axis is the accuracy or MSE. (a) Classification accuracy is proportional to the output weight. (b) Regression error is inversely proportional to the output weight.

V. EXPERIMENTAL RESULTS AND DISCUSSION

To confirm the efficiency of the proposed method, extensive experiments are conducted. Through the experimental results, we find that the efficiency of random projection can be largely improved with comparable performance so that the proposed method can be used for rather large data sets. In this section, some experiments are first conducted to prove that the feature with a larger output weight has a higher probability to generate better performance. Then, the performance and efficiency of the proposed method is evaluated on both classification and regression tasks.

A. Why the Proposed Method is Working?

Empirically, features having larger weights tend to generate better performance since large weights indicate the significance of features. To prove this idea, several experiments are conducted, including classification and regression tasks.

In particular, we randomly generate some 1-D features and calculate their corresponding output weight and accuracy/MSE. Then these features are ranked by the value of their output weights. The relationship of the output weight and performance is thus shown in Fig. 2. In this figure, the classification accuracy is proportional to the output weight and the regression error is inversely proportional to the weight which indicates that the larger weight has higher probability to yield better performance in both classification and regression tasks.

To further confirm the effectiveness of feature selection, the performance of the original algorithm is compared with the performance of the algorithm with feature selection. In detail, the original algorithm is compared with the algorithm with the same number of features, which have rather higher output weights selected from a feature pool of 500 features. The experimental result is shown in Fig. 3 in which the classification accuracy of algorithm with features selected from the feature pool is higher than the algorithm with randomly generated features. The experimental result indicates that the algorithm with selected features always has a better performance.

TABLE II

COMPARISON OF DIFFERENT METHODS ON CLASSIFICATION TASKS. THE BEST RESULTS ARE INDICATED IN **BOLD**

	SVM		Adaboost		ELM		Ours	
	accuracy	testing time	accuracy	testing time	accuracy	testing time	accuracy	testing time
dig	**97.96%**	0.039	95.34%	0.088	96.98%	0.053	97.20%	**0.028**
msra	99.85%	0.093	99.27%	0.142	99.61%	0.130	**99.94%**	**0.057**
palm	99.60%	**0.020**	96.50%	0.060	99.00%	0.049	**99.80%**	**0.02**
tdt	97.57%	1.283	97.71%	1.193	98.21%	0.033	**99.17%**	**0.030**
text	**96.92%**	5.464	94.60%	0.764	93.93%	2.496	95.29%	**1.676**
usps	94.95%	0.997	91.84%	0.358	**96.74%**	0.334	96.51%	**0.172**

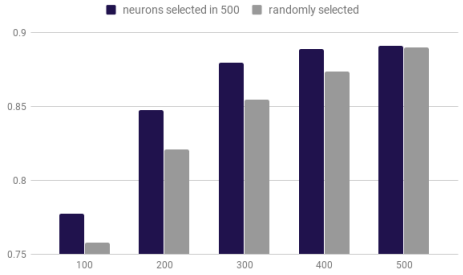


Fig. 3. Comparison of single-layer neural networks with or without feature selection on classification task. The horizontal axis is the number of features randomly generated or selected from a feature pool. The vertical axis is classification accuracy.

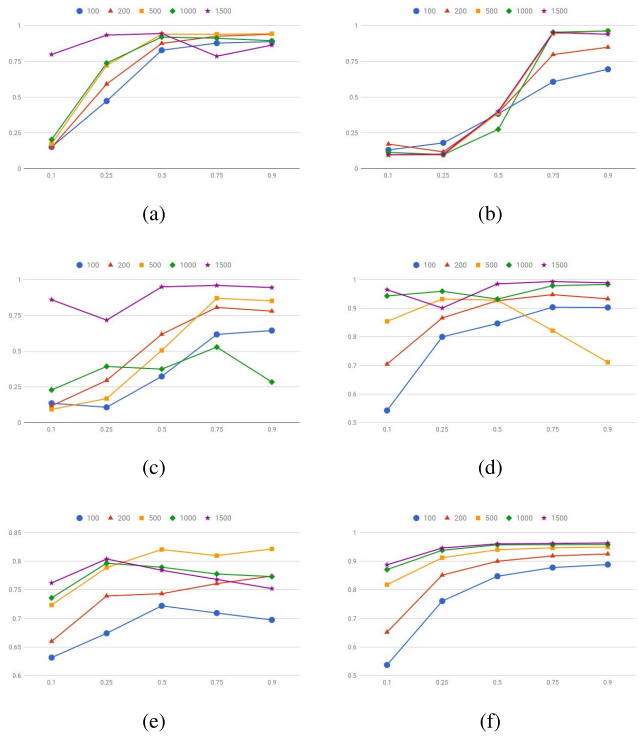


Fig. 4. Experimental results regarding to how many features should be retained in each selection procedure on different data sets. In this figure, horizontal axis is the ratio of remained features in the selection procedure. The vertical axis is accuracy. Different lines represent the performance of FSRP with different number of final remained features. (a) dig. (b) msra. (c) palm. (d) tdt10. (e) text. (e) usps.

B. Random Projection With Feature Selection for Classification

Classification is one of the fundamental problems in artificial intelligence. To assess the efficiency of the proposed method, three popular classifiers (ELM, support vector machine, and Adaboost) are included for comparison on several widely used benchmarks (dig [24], msra [25], palm [26], tdt10 [27], text [28], and usps [29]) and the detailed information is concluded in Table I.

We first investigate three hyper-parameters that should be considered: 1) how many features should be remained when we select useful features and 2) how many features should be remained at last and the number of layers in hierarchical scheme. The first parameter is the number of features selected each time. Based on

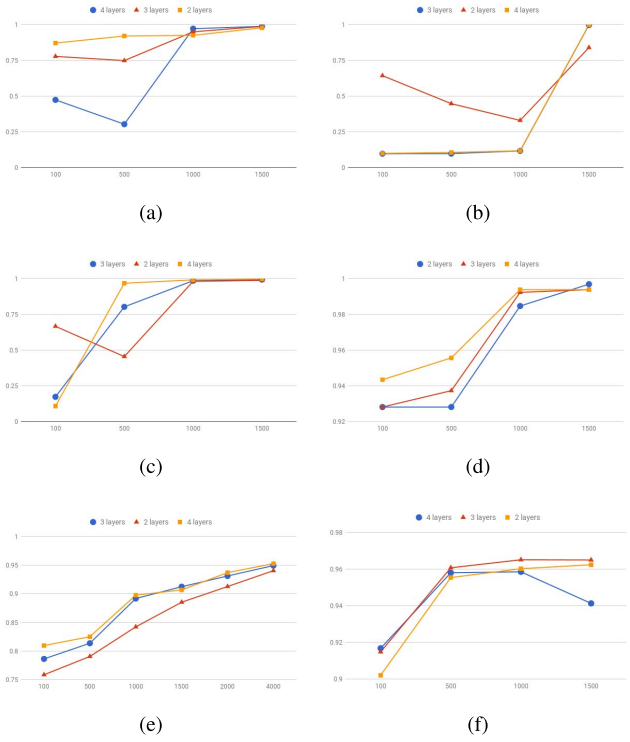


Fig. 5. Experimental results about the number of layers in the selection scheme and the size of features remained at last on different data sets. In this figure, the horizontal axis is the number of final remained features and the vertical axis is accuracy. Different lines represent the performance of FSRP with different selection layers. (a) dig. (b) msra. (c) palm. (d) tdt10. (e) text. (e) usps.

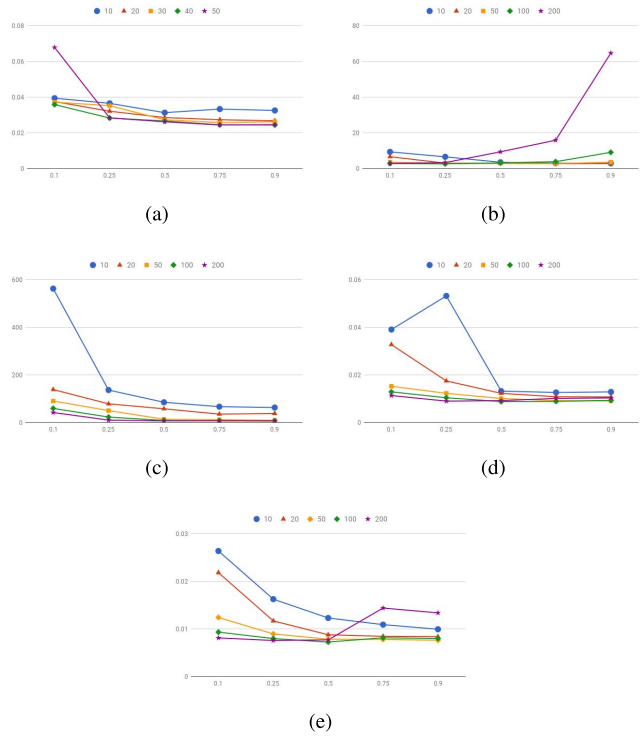
TABLE III

DETAILED INFORMATION OF DATA SETS FOR REGRESSION

	size of dataset	dimension
NWPU_Cong	5585	10
abalone	4177	8
cpusmall	8192	12
mg	1385	6
space_ga	3107	6

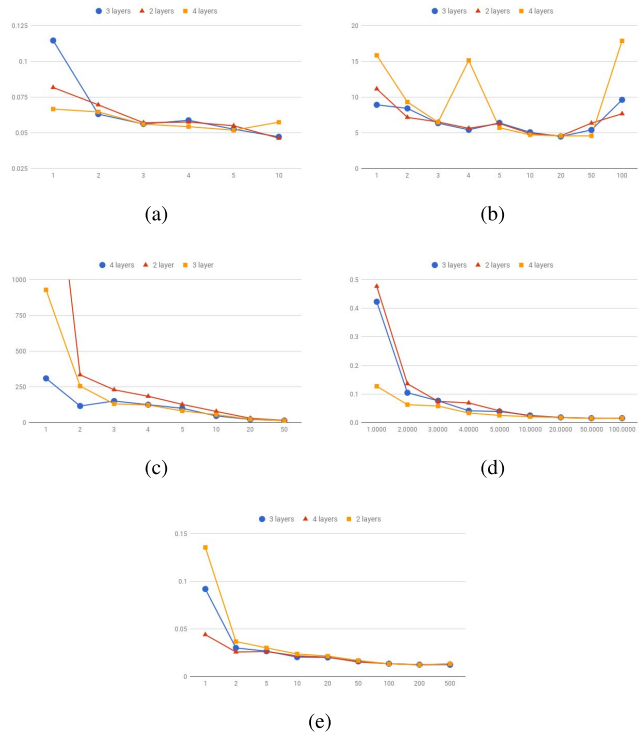
TABLE IV
COMPARISON OF DIFFERENT METHODS ON REGRESSION TASKS. THE BEST RESULTS ARE INDICATED IN **BOLD**

	Adaboost		OPELM		ELM		Ours	
	MSE	testing time	MSE	testing time	MSE	testing time	MSE	testing time
NWPU_Cong	**0.021**	0.038	0.026	0.019	0.029	0.026	0.029	**0.014**
abalone	7.864	0.020	5.610	0.010	5.250	0.012	**4.482**	**0.006**
cpusmall	13.75	0.041	17.27	0.02	11.99	0.02	**11.52**	**0.02**
mg	0.024	0.020	**0.015**	0.0064	0.0153	0.0060	**0.015**	**0.004**
space_ga	0.019	0.024	0.013	0.005	**0.012**	0.005	**0.012**	**0.004**



(a) (b) (c) (d) (e)

Fig. 6. Experimental results about how many features should be remained on different data sets. In this figure, the horizontal axis is the ratio of remained features in selection procedure and the vertical axis is accuracy. Different lines represent the performance of FSRP with different number of final remained features. (a) NWPU_Cong. (b) abalone. (c) cpusmall. (d) mg. (e) space_ga.



(a) (b) (c) (d) (e)

Fig. 7. Experimental results about how many layers the selection procedure should include on different data sets. In this figure, the horizontal axis is the number of final remained features and the vertical axis is accuracy. Different lines represent the performance of hierarchical feature selection with FSRP with different selection layers. (a) NWPU_Cong. (b) abalone. (c) cpusmall. (d) mg. (e) space_ga.

the aforementioned declaration that it is the tradeoff between the performance and efficiency, the performance of single-layer neural network with a different compression level is compared, to find out a proper number that can maximumly compress the network without much performance decrease. The experimental results are shown in Fig. 4 in which we can see that the performance can be guaranteed with a rather high compression degree when the ratio equals to 0.5 over most of the data sets. Thus, the ratio is set to 0.5 in the following experiments.

The number of final remained features and the number of layers are explored at the same time. The proposed algorithms with different layers and remained features are trained and compared in the experiment. The results are shown in Fig. 5. In this figure, we can see that a two-layer selection scheme cannot achieve the best or comparable results than a three- or four-layer selection scheme. And a four-layer selection method may generate unstable performance on some data sets such as dig or may cause overfitting on usps. Thus, the number of layers of hierarchical selection scheme is set to three in the following experiments.

Table II shows the experimental results regarding to the comparison of some widely used classification algorithms. The proposed method is the most efficient one among all compared algorithms with comparable classification performance since the hierarchical feature selection can prune redundant and noisy features effectively and generate more compact networks. Comparing the proposed algorithm with the algorithm without feature selection, better performance can be obtained with more compact networks and faster testing speed, which proves the effectiveness of the proposed selection method.

C. Random Projection With Feature Selection for Regression

In this section, the performance and efficiency of the proposed method is evaluated on regression tasks. Specifically, the proposed algorithm (FSRP) is compared with Adaboost, optimal ELM (OPELM), and ELM. The detailed information of data sets (abalone [30] cpusmall,[1] mg [31], and space_ga [32]) can be seen in Table III. Most of these data sets come from University of California, Irvine, repository and the NWPU_Cong data set [33] is a real-world congestion detection data set. Note that, the MSE which is popularly used in many tasks [34], [35] is utilized to evaluate the performance of regression.

Similar to the experiments in classification tasks, the parameters are first exploited. We find that the ratio can be set to 0.5 over both

[1] http://www.cs.toronto.edu//delve/data/datasets.html

classification and regression tasks, as shown in Fig. 6. A smaller ratio may cause worse performance on most of the data set while a larger ratio may cause overfitting on abalone and space data sets. The number of layers can be set as two or three based on the results shown in Fig. 7 and only four-layer scheme may cause overfitting on abalone data set. Since the size of regression data sets is rather small, a four-layer selection scheme is far enough to fit the training data. However, overfitting occurs on only one data set which confirms the proposed method is very robust to avoid overfitting.

The experimental results are summarized in Table IV. In most cases, the proposed method achieves superior performance than others. What is more, the testing time is always shorter than others which confirmed the proposed selection criterion is very effective to compress network by avoiding redundant and noisy features.

VI. Conclusion and Future Works

In this paper, we propose a novel criterion to rank the importance of neurons for the compression of neural networks, which establishes a new way for network architecture design. To prune redundant and noisy features, a hierarchical feature selection method is proposed. The selection method which can be used to select useful features from feature pool is based on a simple but effective criterion regarding to the value of the learned weight. Based on the selection criterion, a hierarchical selection scheme is proposed to accelerate the efficiency of the algorithm. Utilizing hierarchical feature selection in single-layer neural networks can generate very compact networks which have comparable performance with several widely used algorithms in classification tasks and better results than traditional methods in regression tasks. And the testing speed has shown its priority in both classification and regression tasks.

Although the proposed algorithm is effective for random projection, it can be further improved by automatically estimating hyper-parameters such as the layer numbers. In the future, adaptive parameter selection methods should be exploited.

References

[1] J. Yang, D. Chu, L. Zhang, Y. Xu, and J. Yang, "Sparse representation classifier steered discriminative projection with applications to face recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 7, pp. 1023–1035, Jul. 2013.

[2] C. Yan, H. Xie, D. Yang, J. Yin, Y. Zhang, and Q. Dai, "Supervised hash coding with deep neural network for environment perception of intelligent vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 284–295, Jan. 2018.

[3] C. Yan, H. Xie, S. Liu, J. Yin, Y. Zhang, and Q. Dai, "Effective uyghur language text detection in complex background images for traffic prompt identification," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 220–229, Jan. 2018.

[4] C. Yan *et al.*, "An effective Uyghur text detector for complex background images," *IEEE Trans. Multimedia*, to be published, doi: 10.1109/TMM.2018.2838320.

[5] C. Yan *et al.*, "Efficient parallel framework for HEVC motion estimation on many-core processors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 12, pp. 2077–2089, Dec. 2014.

[6] C. Yan *et al.*, "A highly parallel framework for HEVC coding unit partitioning tree decision on many-core processors," *IEEE Signal Process. Lett.*, vol. 21, no. 5, pp. 573–576, May 2014.

[7] Q. Wang, Z. Qin, F. Nie, and X. Li, "Spectral embedded adaptive neighbors clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2018.2861209.

[8] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 809–821, Apr. 2016.

[9] C. Yao, Y.-F. Liu, B. Jiang, J. Han, and J. Han, "LLE score: A new filter-based unsupervised feature selection method based on nonlinear manifold embedding and its application to image recognition," *IEEE Trans. Image Process.*, vol. 26, no. 11, pp. 5257–5269, Nov. 2017.

[10] Q. Wang, F. Zhang, and X. Li, "Optimal clustering framework for hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2018.2828161.

[11] G.-B. Huang and L. Chen, "Enhanced random search based incremental extreme learning machine," *Neurocomputing*, vol. 71, nos. 16–18, pp. 3460–3468, Oct. 2008.

[12] Y. Yang, Y. Wang, and X. Yuan, "Bidirectional extreme learning machine for regression problem and its learning effectiveness," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 9, pp. 1498–1505, Sep. 2012.

[13] H.-J. Rong, Y.-S. Ong, A.-H. Tan, and Z. Zhu, "A fast pruned-extreme learning machine for classification problem," *Neurocomputing*, vol. 72, no. 1, pp. 359–366, 2008.

[14] L. Ying and L. Fan-Jun, "A pruning algorithm for extreme learning machine," in *Proc. Int. Conf. Intell. Data Eng. Automated Learn.*, 2013, pp. 1–7.

[15] J. Luo, C.-M. Vong, and P.-K. Wong, "Sparse Bayesian extreme learning machine for multi-classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 4, pp. 836–843, Apr. 2014.

[16] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, and A. Lendasse, "OP-ELM: Optimally pruned extreme learning machine," *IEEE Trans. Neural Netw.*, vol. 21, no. 1, pp. 158–162, Jan. 2010.

[17] M. Luo, F. Nie, X. Chang, Y. Yang, A. G. Hauptmann, and Q. Zheng, "Adaptive unsupervised feature selection with structure regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 944–956, Apr. 2017.

[18] C. Wang, Q. Hu, X. Wang, D. Chen, Y. Qian, and Z. Dong, "Feature selection based on neighborhood discrimination index," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 2986–2999, Jul. 2018.

[19] J. Sun, A. Zhou, S. Keates, and S. Liao, "Simultaneous Bayesian clustering and feature selection through student's t mixtures model," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 1187–1199, Apr. 2018.

[20] P. A. Kowalski and M. Kusy, "Sensitivity analysis for probabilistic neural network structure reduction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1919–1932, May 2018.

[21] X. Zhu, S. Zhang, R. Hu, Y. Zhu, and J. Song, "Local and global structure preservation for robust unsupervised spectral feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 3, pp. 517–529, Mar. 2018.

[22] R. Shang, W. Wang, R. Stolkin, and L. Jiao, "Non-negative spectral learning and sparse regression-based dual-graph regularized feature selection," *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 793–806, Feb. 2018.

[23] N. Wang, M. J. Er, and M. Han, "Parsimonious extreme learning machine using recursive orthogonal least squares," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 10, pp. 1828–1841, Oct. 2014.

[24] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 3, pp. 418–435, May/Jun. 1992.

[25] S. A. Nene *et al.*, "Columbia object image library (COIL-20)," Tech. Rep., 1996.

[26] C. Hou, F. Nie, C. Zhang, and Y. Wu, "Learning an orthogonal and smooth subspace for image classification," *IEEE Signal Process. Lett.*, vol. 16, no. 4, pp. 303–306, Apr. 2009.

[27] F. Wang, C. Tan, P. Li, and A. C. König, "Efficient document clustering via online nonnegative matrix factorizations," in *Proc. Int. Conf. Data Mining*, 2011, pp. 908–919.

[28] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2005, pp. 57–64.

[29] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning* (Springer Series in Statistics), vol. 1. New York, NY, USA: Springer, 2001.

[30] D. Dheeru and E. K. Taniskidou, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, Irvine, CA, USA, Tech. Rep., 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[31] G. W. Flake and S. Lawrence, "Efficient SVM regression training with SMO," *Mach. Learn.*, vol. 46, nos. 1–3, pp. 271–290, 2002.

[32] R. K. Pace and R. Barry, "Quick computation of spatial autoregressive estimators," *Geograph. Anal.*, vol. 29, no. 3, pp. 232–247, 1997.

[33] J. Wan, Y. Yuan, and Q. Wang, "Traffic congestion analysis: A new perspective," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2017, pp. 1398–1402.

[34] Y. G. J. H. Sicheng Zhao and G. Ding, "Approximating discrete probability distribution of image emotions by multi-modal features fusion," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 4669–4675.

[35] Y. Guo, G. Ding, and J. Han, "Robust quantization for general similarity search," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 949–963, Feb. 2018.