Neurocomputing 173 (2016) 744–750

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Automatic feature extraction based structure decomposition method for multi-classification

Liping Xie [a,b], Haikun Wei [a,b,*], Junsheng Zhao [c], Kanjian Zhang [a,b]

[a] School of Automation, Southeast University, Nanjing 210096, China
[b] Key Laboratory of Measurement and Control of CSE, Ministry of Education, School of Automation, Southeast University, Nanjing 210096, China
[c] School of Mathematical Science, Liaocheng University, Liaocheng 252059, China

ARTICLE INFO

Article history:
Received 5 August 2014
Received in revised form
18 July 2015
Accepted 10 August 2015
Available online 30 August 2015

Keywords:
Neural networks
SDBSkeletonization
Classification
Automatic feature extraction

ABSTRACT

For years, researchers in neural network (NN) area have been carried out much productive research in improving the generalization ability of NNs. In this paper, a novel neural network design algorithm is presented for solving multi-class problems, structure decomposition based on Skeletonization (SDBSkeletonization), which is to simplify NNs further. The proposed method decomposes a complex multi-class problem into a set of two-class problems, each of which can be regarded as an individual problem. After learning all these individual problems in parallel with Skeletonization algorithm, we then integrate these results to final decision. In addition, Skeletonization solves the classification problem based on automatic feature extraction. This perspective gives a broader range of application of our method. Our experimental results on Waveform and Handwritten Digits database demonstrate that SDBSkeletonization improves the overall classification performance.

1. Introduction

Neural networks [1–3] have shown good abilities to learn and reconstruct complex nonlinear problems, so they have been rapidly applied into the fields of function approximation [4], pattern recognition [5–7], optimization [8], various predictions and control systems [9–11].

As for learning systems, a pretty important and efficient application area of neural networks is pattern classification problem. Generalization ability [12] is the most useful feature when we evaluate the performance of NNs. In recent years, much deep and productive research in this area has been carried out [13]. A rule of thumb for gaining good generalization of NNs is that we need to use the smallest structure which can fit all the training samples [14]. Up to now, a variety of techniques to simplify the NNs have been brought up [15]. They can be roughly categorized into two groups:

(1) *Pruning algorithms* [16]: The objective of all these algorithms is to train a network which is obviously larger than necessary and then prune the components one by one which is not needed. Among them, the most classic are Weight-Elimination [17,18], Skeletonization [19] and Correlation Pruning algorithms [20]. A limitation of these pruning algorithms is the uncertainty of the initial size of networks. Too complex a network is slow and inefficient while too simple a network will not learn well.

(2) *Constructive algorithms* [21]: Contrary to pruning algorithms, constructive algorithms start by a minimal network, then automatically trains and adds new components one by one, such as Cascade-Correlation [22] and Resource-Allocating algorithm [23]. The main problem of these networks is that there is no good criterion to decide when to stop.

Inspired by the aforementioned ideology, we come up with a novel algorithm to simplify the network further. We combine a structure decomposition method and integration approach to solve the complex multi-class problems more effectively and efficiently. Firstly, we decompose a complex problem into a set of two-class problems, each of which can be regarded as an individual problem, then they can work in parallel. After that, we integrate these results to final decision.

The Resilient Backpropagation (RPROP) learning algorithm is a common method of training neural networks, which has been employed to solve the most wide variety of applications both in engineering and science. The efficiency and simpleness in solving learning problems lead to its success. However, from the point of view of optimization and flexibility, RPROP is not very desirable because the numbers of input, and hidden nodes are fixed. The problem of information redundancy is inevitable and further simplification is impossible. To solve the problem, we adopt Skeletonization algorithm [24]. Using the knowledge in a network to determine the relevance of individual nodes, the relevance metric

* Corresponding author.
E-mail address: hkwei@seu.edu.cn (H. Wei).

http://dx.doi.org/10.1016/j.neucom.2015.08.025
0925-2312/

can identify which input or hidden units are most critical to the performance of the network. The least relevance nodes can then be trimmed to construct a skeleton version of the network. In other words, Skeletonization can solve the classification problem by reducing the complexity of the neural networks. This perspective gives a broader range of application of our method. This approach is more applicable and adaptive. The algorithm does not need to be modified in a different problem area.

Our experimental results on Waveform and Handwritten Digits database [19] demonstrate that our method is effective and improves the overall classification performance.

The rest of this paper is organized as follows. In Section 2, we prove the decomposition principle and introduce some relevant theories. The overview of our proposed method is shown in Section 3. In Section 4, several experimental results are given. Finally, conclusions are presented in Section 5.

2. Relevant theories

2.1. Decomposition principle

Theorem 1. *The multiple-input–multiple-output system of feedforward neural networks with the structure of $N_I - N_H - N_O$ can be decomposed to N_O feedforward subnets of multiple-input–single-output. The input–output characteristic remains unchanged.*

As illustrated in Fig. 1, to ensure equivalence of a three-class neural network after structure decomposition, we design three single-output subnets with the structure of $N_I - N_H - 1$ to form a system. The input weight matrices of all the subnets are all identical to the original network. And the output weight matrix (a vector) of the ith subnet is identical to the ith column of the matrix W^2, which is the output weight matrix of the three-class neural network . All the outputs of the subnets are then combined to generate three-dimensional column vector. Therefore the decomposed subnets have the exact same input–output characteristics with the multiple-input–multiple-output system of $N_I - N_H - N_O$.

The multiple-input–multiple-output feedforward neural networks with the structure of $N_I - N_H - N_O$ can be decomposed to N_O feedforward subnets of multiple-input–single-output when the condition of same approximating precision is met. And the number of hidden nodes of each subnet is less than N_H, which means after structure decomposition, the network can be simplified further.

We know from Theorem 1 that, the structure decomposed system, in which every subnet has the structure of $N_I - N_H - 1$, and the original network are completely equal. If the structure of the subnets can be further simplified with the condition of same approximating precision, then the number of the hidden nodes of the subnets will be even smaller.

Actually, each single-output subnet just needs to be able to recognize the samples corresponding to their own class. While the original multiple output system needs to recognize all the multi-class samples. As for each subnet, the complexity of the task it needs to accomplish is decreased. And the decomposed system has better generalization ability than the original network. In conclusion, it can be held that better generalization ability can be acquired after decomposing the multiple output system to multi single-output networks.

2.2. Computational complexity

The structural complexity of neural networks refers to the size or capacity of the networks. For the linear-threshold neural network model, structural complexity generally means VC dimension. The complexity theory can be viewed as a refinement of the theory of recursive functions, which is developed after Goedel's paper about the imperfectness of formal systems [25]. The theory separates what is computable from what is not computable. Roughly speaking, the aim of complexity theory can be summarized as follows: given a model or a computational problem, to determine how much computational power (in terms of used resources) we need to solve the problem [26].

There is a valuable conclusion given by Moody [27](so-called Moody criterion): For the neural networks of function approximation: when we use the regularization method in neural network learning systems, the number of valid parameters does not equal the number of free parameters. Regarding the expected error on the testing set as the generalization error, Moody has done much research on the relationship between the expected error on the testing set and the training set, and come to the following formula:

$$\langle \varepsilon_{test}(\lambda)\rangle_{\xi\xi'} \approx \langle \xi_{train}(\lambda)\rangle_{\xi} + 2\sigma_{eff}^2 \frac{p_{dff}(\lambda)}{N} \tag{1}$$

where λ is the regularization parameter, $p_{eff}(\lambda)$ and σ_{eff}^2 refer to the number of valid parameter and the valid noise variance of output variable, respectively. $p_{eff}(\lambda)$ does not mean the number of free parameter p, it depends on the model deviation, nonlinear degree, regularization parameter and the form of the regularization item. The above conclusion also confirms the structure minimum principle of the neural networks: for the network which has reached the given training accuracy, the less the valid parameter is, the better the generalization ability will be.

When we construct a neural network, the key issue which is most intricacy and momentous is how to determine the number of the hidden nodes. This problem has not yet been satisfactorily solved though it has been explored for so many years. But we know that the number of hidden nodes depends largely on the complexity of the task, including the size of the training set and the number of classes. That is, the more training samples and the number of classes are, the more hidden nodes are needed. Otherwise, the too few parameters will lead to no convergence.

As for the multiple classification problems, after we first decompose the task, the number of hidden nodes will be much less due to the simplified complexity of the task for each subnet. Then we use the Skeletonization algorithm, after automatic feature extraction, the

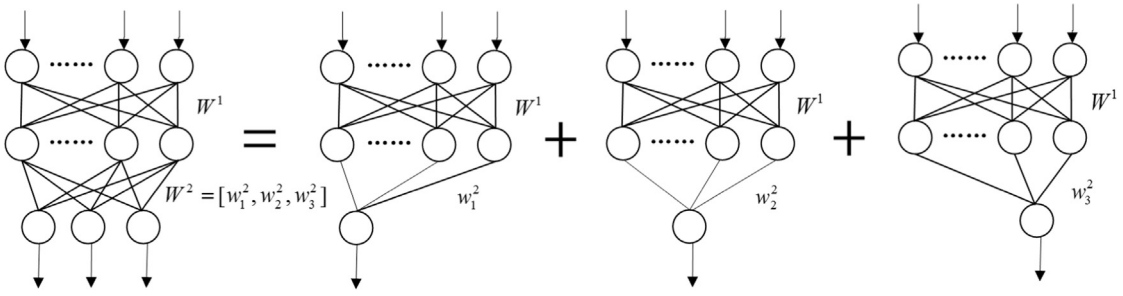


Fig. 1. Illustration of the structure decomposition of a three-class problem.

redundant information is trimmed and the structure of the network is simplified further. Based on the analysis of computational complexity problem, our proposed method SDBSkeletonization will greatly simplify the complex classification problem. This scheme will have better generalization ability and hasten the learning process.

2.3. Skeletonization

Skeletonization [24], proposed by Mozer and Smolensky, is a technique for trimming the fat from a network via relevance assessment. The most critical input or hidden nodes to the performance of the networks are reserved.

The way of considering whether a node needs to be eliminated or not is to think how well does the network perform with the node versus without it. For node i, the measure of the relevance, ρ_i is

$$\rho_i = E_{without\ node\_i} - E_{with\ node\_i} \tag{2}$$

where E is the error of the network on the training set. This formula is used to compute the error with a given node removed, a complete pass must be made through the training set. The key to the Skeletonization technique is to find a fast computation of ρ_i.

(1) *Approximation method of* ρ_i: To find a good approximation to ρ_i, Mozer and Smolensky associated with each input or hidden node i a coefficient α_i, which represents the attentional strength of the node (see Fig. 2). This coefficient is not a parameter of the network, it can be thought of as gating the flow of activity from the node:

$$o_j = f\left(\sum_i w_{ji}\alpha_i o_i\right) \tag{3}$$

where o_j is the activity of unit j, w_{ji} the connection strength to j from i, and f the sigmoid squashing function. If $\alpha_i = 0$, node i has no influence on the rest of the network; if $\alpha_i = 1$, node i is a conventional unit. In terms of α, the relevance of node i can then be rewritten as

$$\rho_i = E_{\alpha_i=0} - E_{\alpha_i=1} \tag{4}$$

(2) Computation of ρ_i: ρ_i is approximated by using the derivative of the error with respect to α:

$$\lim_{\gamma\to 1}\frac{E_{\alpha_i=\gamma} - E_{\alpha_i=1}}{\gamma - 1} = \left.\frac{\partial E}{\partial \alpha_i}\right|_{\alpha_i=1} \tag{5}$$

Assuming that this equality holds approximately for $\gamma = 0$:

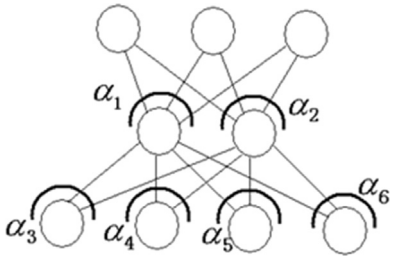


Fig. 2. A network with attentional coefficients on the input and hidden nodes.

$$\frac{E_{\alpha_i=0} - E_{\alpha_i=1}}{-1} \approx \left.\frac{\partial E}{\partial \alpha_i}\right|_{\alpha_i=1} \tag{6}$$

Then the approximation for ρ_i is

$$\rho_i = -\left.\frac{\partial E}{\partial \alpha_i}\right|_{\alpha_i=1} \tag{7}$$

This derivative can be computed using an error propagation procedure very similar to that used in adjusting the weights with back propagation. Additionally, note that because the approximation assumes that α_i is 1, the α_i never needs to be changed. Thus, the α_i is not actual parameter of the system, just a bit of notational convenience used in estimating relevance. The error we used when we compute ρ_i is

$$E^l = \Sigma|t_{pj} - o_{pj}| \tag{8}$$

Compared to

$$E^q = \Sigma(t_{pj} - o_{pj})^2 \tag{9}$$

when the total error is small, absolute error can also leads to good approximation of ρ_i. Also we can use E^l to compute ρ_i and use E^q to train the weights.

3. SDBSkeletonization

Based on Skeletonization technique, SDBSkeletonization algorithm consists of two steps: structure decomposition is the first step, and then to integrate all the output values of individual networks to final decision.

3.1. Structure decomposition

Assuming that T represents the training set for a K-class problem, then

$$T = \{(U_n, Y_n)\}_{n=1}^{N} \tag{10}$$

where $U_n \in R^p$ is the input value, $Y_n \in R^K$ is the desired output value for U_n, and N is the total number of the training samples.

Following the structure decomposition method, a K-class problem is decomposed into K two-class problems, each of which can be represented as

$$T_k = \{(U_{n_k}, Y_{n_k})\}_{n_k=1}^{N_k}, \quad k = 1, \ldots, K \tag{11}$$

$$N = \sum_k N_k \tag{12}$$

where N_k is the number of the training samples from class C_k, $Y_k \in R^1$ is the desired output value which is defined by

$$Y_{n_k} = \begin{cases} 1 - \partial, & \text{if } U_{n_k} \text{ belongs to class } C_k \\ \partial, & \text{if } U_{n_k} \text{ belongs to class } \overline{C_k} \end{cases} \tag{13}$$

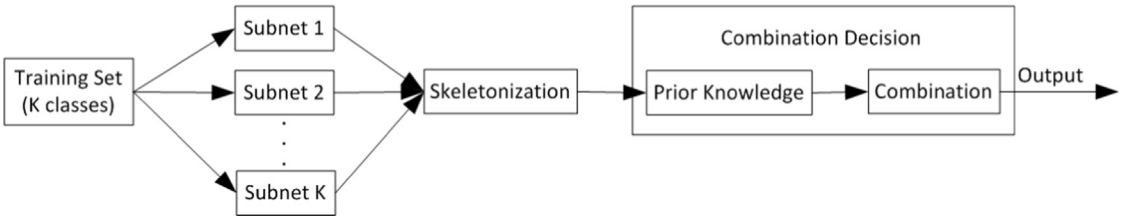


Fig. 3. The flowchart of SDBSkeletonization.

where ∂ is a low positive-real parameter, $\overline{C_k}$ denotes all the classes except C_k. As shown in Fig. 1, a three-class problem can be divided into three two-class ones.

3.2. Combination decision

When we integrate all the output values of individual networks to final decision, we distribute the different weights for different subnets. The criterion has been worked out with reference to the reliability of each subnet. In the following section, we first introduce the way of computing the reliability, then we finally integrate all the subnets.

3.2.1. Prior knowledge

For the training set, we use the confusion matrix [28] to describe the errors of all the classifiers, which can be written as

$$PT = \begin{pmatrix} n_{11} & n_{12} & \cdots & n_{1K} \\ n_{21} & n_{22} & \cdots & n_{2K} \\ \cdots & \cdots & \cdots & \cdots \\ n_{K1} & n_{K2} & \cdots & n_{KK} \end{pmatrix} \tag{14}$$

where each column j denotes the output result for the subnet C_i and each row i corresponds to class C_i. So the element n_{ij} means that n_{ij} training samples from the class C_i have been recognized as the ones from the class C_j.

The confusion matrix PT, obtained from all the subnets with the training set, reflects the reliability of all the subnets. In another word, PT gives us the distribution of pattern space. Following from (14), we can acquire the total number of the training set

$$N = \sum_i \sum_j n_{ij} \tag{15}$$

In addition, the number of the training samples from class C_i is

$$N_i = n_{i.} = \sum_j n_{ij}, \quad i = 1, \ldots, K \tag{16}$$

and the number of the training samples that have been recognized as the ones from the class C_j is

$$M_j = n_{.j} = \sum_i n_{ij}, \quad j = 1, \ldots, K \tag{17}$$

Based on the above analysis, for each subnet, we can obtain its uncertainty (i.e. its reliability of the output result).

With the knowledge of the confusion matrix PT, the uncertainty could be regarded as the conditional probabilities, that is

$$P_{ij} = P(U_n \in C_i | e_k(U_n) = j) = \frac{n_{ij}}{n_{.j}} = \frac{n_{ij}}{\sum_i n_{ij}}, \quad i = 1, \ldots, K \tag{18}$$

$$P = \begin{pmatrix} P_{1.} \\ P_{2.} \\ \cdots \\ P_{K.} \end{pmatrix} = \begin{pmatrix} P_{11} & P_{12} & \cdots & P_{1K} \\ P_{21} & P_{22} & \cdots & P_{2K} \\ \cdots & \cdots & \cdots & \cdots \\ P_{K1} & P_{K2} & \cdots & P_{KK} \end{pmatrix} \tag{19}$$

where $e_k(U_n)$ denotes the output value of the kth subnet for the input value U_n.

From another point of view, the confusion matrix PT, obtained from the training set, could be treated as the prior knowledge of each subnet. We can use this knowledge to distribute the weights among the subnets.

3.2.2. Combination

Generally, the neural network we adopted for each subnet has only one output node, which generates the output value O_{i^k} to be in the range of [0,1] (i.e. $O_{i^k} \in [0, 1)$). Where O_{i^k} denotes the output of the kth classifier on ith sample from the testing set. The values of O_{i^k} can be regarded as a posterior-probility with which an input U_n belongs to the class C_i, namely:

$$p_i^k(C_i|U_n) = O_{i^k} \tag{20}$$

Then the new estimate of the posterior-probility with which U_n belongs to C_i can be determined by

$$P_k(C_i|U_n) = \sum_i P_{ik} p_i^k(C_i|U_n) = \sum_i P_{ik} O_{i^k} \tag{21}$$

where P_{ik} satisfy

$$\sum_i P_{ik} = 1 \tag{22}$$

Then the final decision can be given by

$$U_n \in C_{k'}, \quad \text{if } k' = \text{argmax}_k\{P_k(C_i|U_n)\} \tag{23}$$

Fig. 3 shows the flowchart of our proposed method, SDBSkeletonization.

3.3. The algorithm of the proposed method

As described above, our proposed method, SDBSkeletonization consists of several sections. The overall algorithm can be described as follows:

Step 1. Decompose a K-class problem into K two-class problems.
Step 2. Select a relatively large architecture of each individual network and initialize the learning parameters.
Step 3. Train all the individual networks in parallel based on Skeletonization algorithm.
Step 4. Get the confusion matrix from the training set and obtain the prior knowledge.
Step 5. Combine K two-class output values to obtain the final decision based on the prior knowledge.

4. Experiments

In the simulations, we use the Waveform and Handwritten Digits database (Opt-digits) to evaluate the performance of SDBSkeletonization method. Under the same simulation conditions, RPROP algorithm and Skeletonization algorithm are used for comparison. It has to be noted that all the results are the average of 100 times.

4.1. Waveform classification

The Waveform problem (Version 2) [19] was brought forward by Breiman, Friedman, etc. (1984). This database consists of three classes of waves, each of which was generated from a combination of 2 of 3 “base” waves. The “base” waves can be seen in Fig. 4. In addition, noise was added (mean 0, variance 1) in each instance. In our experiment, we choose 5000 samples to divide into two parts, 300 samples are used for training while the other are used for testing. In our experiment, we have converted it to three one-output subnets. The ith subnet is used to judge if the sample belongs to ith class or not.

The experimental results are summarized in Tables 1–3. From these experimental results, it is easily noted that our method is effective for classification. Although the result is not completely comparable due to a different experiment context, the generalization ability of SDBSkeletonization has improved greatly compared to most other methods. In addition, references [36,37] adopting different versions of Waveform database, used more difficult experimental setups than ours. As shown in Table 1, the input numbers in our method are only half of their original ones in each subnets and the

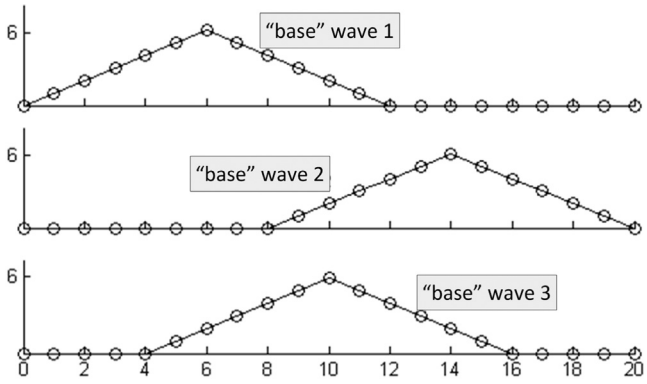


Fig. 4. Three base wave of the Waveform classification problem.

Table 1
Performance comparison of RPROP, Skeletonization and the proposed SDBSkeletonization network on the Waveform classification problem.

Classifiers	LeftHiddenNum	Number of SNets	LeftInNum	Error ratio of recognition	
				Training samples	Testing samples
RPROP	50	1	40	0.0	15.23
Skeletonization	49	1	32	9.47	15.10
Ours(1)	–	3	–	9.36	14.11
Ours(2)	–	3	–	9.13	13.20

Table 2
Illustration of the parameters of the three subnets on the Waveform classification problem.

Criterion	SNet 1	SNet 2	SNet 3
TestErrorRate	11.90	10.23	9.65
LeftHiddenNum	43	42	43
LeftInNum	17	24	24

Table 3
Comparison of accuracy with different approaches on Waveform database.

Database	[29]	[30]	[31]	[32]	[33]	[34]	Ours(2)
Waveform	82.94	83.12	81.50	85.00	89.37	90.45	86.80

Table 4
Performance comparison of RPROP, Skeletonization and the proposed SDBSkeletonization network on the optical recognition of Handwritten Digits database.

Classifiers	LeftHiddenNum	Number of SNets	LeftInNum	Error ratio of recognition	
				Training samples	Testing samples
RPROP	50	1	64	0.0	3.51
Skeletonization	48	1	42	0.37	3.62
Ours(1)	–	10	–	0.16	2.84
Ours(2)	–	10	–	0.13	2.36

hidden unit numbers are also much less. Which means the structure of the subnets is much simplified further. Actually, from Fig. 2, we can find that the first and last input value is entirely random, also the other ones. In addition, the more marginal the data is, the more random it will be and the possibility to be deleted is much bigger. Because for each subnet, the complexity of the task is decreased, and the feature extraction of our method is more obvious and effective.

In Table 1, Ours(1) and Ours(2) denote the algorithms with and without considering the prior knowledge, respectively. It is obvious that the prior knowledge has great contribution to the performance and the recognition rate of Ours(2) is higher than Ours(1). In fact, the classifier of neural networks supplies K values of probabilities for each possible label, the final label j is the result of maximum selection from the K values and this selection certainly discards some information that is considered useless for the final output when there is only a single classifier. However such discarded information may be useful for multiclassifier combination. With the prior knowledge, we make good use of all the information and combine the classification results by all labels. It can be seen that the combined results are more reliable. Table 3 compares our method with several state-of-the-art algorithms in terms of the accuracy. We can find that the proposed method is effective and competitive.

4.2. Optical recognition of Handwritten Digits

The Handwritten Digits problem [18] is a real problem, which was collected from a total of 43 people. All the instances of 10 species include an input matrix of 8×8 where each element is an integer in the range [0, 16]. The database consists of 3823 training samples and 1797 testing ones. Similarly, Table 3 shows the classification error ratio, which imply that SDBSkeletonization has a better generalization ability.

The experimental process is adopted similar to Section 4.1. And Tables 4–6 give the experimental results, which imply that our method can still achieve promising recognition accuracy, which implies its effectiveness. Also, compared to several state-of-the-art methods, the recognition rate obtained by the proposed method is significantly higher than the others. In addition, the structure of the subnets is much simpler.

4.3. Two-dimensional Gaussian distribution

During the process of experiment, we notice that the higher the overlapping of the data set, the more effective our proposed method can be. To prove this conjecture, we generate three data set of two-dimensional Gaussian distribution with a different coupling degree as shown in Fig. 5. The coupling degree of each two classes of Gaussian data in one data set is the same. In our

Table 6
Comparison of accuracy with different approaches on Handwritten Digits database.

Database	[33]	[34]	[35]	LDA	LPP	PCA	Ours(2)
Opt-digits	93.88	96.5	96.27	95.7	95.4	95.3	97.64

Table 5
Illustration of the parameters of the three subnets on the optical recognition of Handwritten Digits database.

Criterion	SNet 1	SNet 2	SNet 3	SNet 4	SNet 5	SNet 6	SNet 7	SNet 8	SNet 9	SNet 10
TestErrorRate	0.17	1.09	0.25	1.20	0.64	0.76	0.36	1.05	1.56	1.31
LeftHiddenNum	29	28	26	29	28	29	29	28	36	32
LeftInNum	35	34	35	31	35	32	34	32	35	31

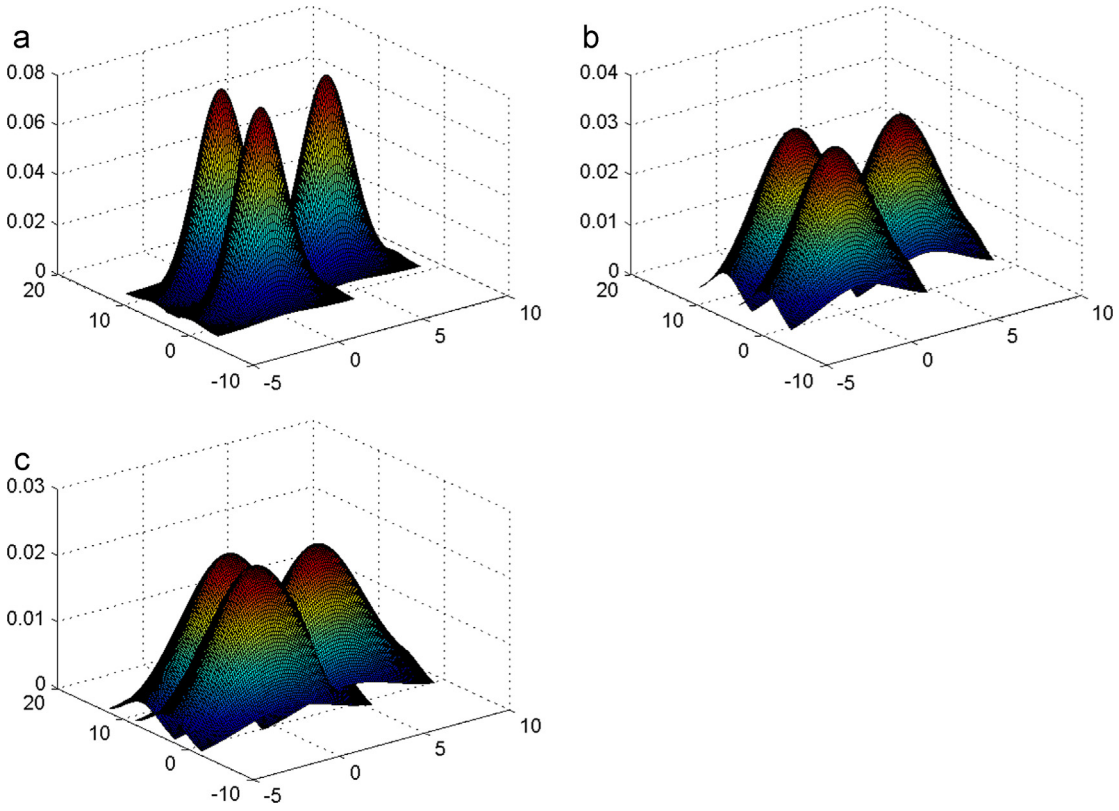


Fig. 5. Three data set of two-dimensional Gaussian distribution with a different coupling degree: (a) coupling degree = 0.029, (b) coupling degree = 0.074, (c) coupling degree = 0.219.

Table 7
Performance comparison of RPROP, Skeletonization and the proposed SDBSkeletonization network on the Gaussian distribution problem.

Fig. 5	Coupling degree	RPROP	Skeletonization	Ours(2)	Improvement of recognition	Relative improvement
(a)	0.029	5.17	4.47	3.42	1.05	30.70
(b)	0.074	17.23	17.19	15.03	2.16	14.37
(c)	0.219	35.06	34.56	30.88	3.68	11.92

experiment, each data set consists of three classes of Gaussian data. 3300 samples are generated, 300 samples are used for training while the other are used for testing.

The coupling degree is computed as follows:

$$Coupling\ degree = \frac{1}{(u_1 - u_2)'(\sum_1^{-1} + \sum_2^{-1})(u_1 - u_2)} \quad (24)$$

where $\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma\sigma_1\sigma_2 \\ \sigma\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$, u represents the mean value, and σ denotes the variance.

The experimental result in Table 7 has proven the validity of our conjecture. The higher the overlapping of the data set, the more effective our proposed method can be. However, for the relative improvement, the opposite is true. More studies are still needed to clarify the internal relation of overlapping and the effectiveness.

5. Conclusions and discussion

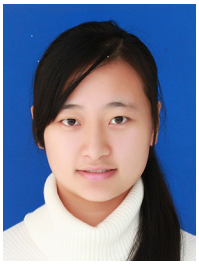
This paper proposed a novel neural network design algorithm to solve multi-class problems. Structure decomposition based on Skeletonization (SDBSkeletonization) is adopted, which is to simplify NNs further after automatic feature extraction. The proposed method decomposes a complex multi-class problem into a set of two-class problems, each of which can be regarded as an individual problem. After learning all these individual problems in parallel with Skeletonization algorithm, we then integrate these results into final decision. Generally speaking, our algorithm divides a complex task with supervised label into some subtasks and then assigns these subtasks to several experts. Finally by integrating, all results of these experts are integrated to produce the solution of the complex problem. In this way, we partition the whole sample space into several subspaces and largely reduce the complexity of the problem, and thereby quicken the learning process and improve the performance. Our experimental results on Waveform and Handwritten Digits database demonstrate that the decomposed method reduces the training time and improves the overall classification performance.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 61374006, Major Program of National Natural Science Foundation of China under Grant 11190015, the Scientific Research Foundation of Graduate School of Southeast University YBJJ1518, and the Program Sponsored for Scientific Innovation Research of College Graduates in Jiangsu Province KYLX15_0113.

References

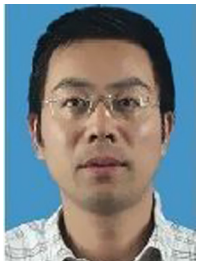
[1] Y. Fan, N.W. Duncan, M. de Greck, et al., Is there a core neural network in empathy? An fMRI based quantitative meta-analysis, Neurosci. Biobehav. Rev. 35 (3) (2011) 903–911.
[2] N. Wang, M. Han, N. Dong, et al., Constructive multi-output extreme learning machine with application to large tanker motion dynamics identification, Neurocomputing 128 (1) (2014) 59–72.
[3] I. Sadeghkhani, A. Ketabi, R. Feuillet., An intelligent switching overvoltages estimator for power system restoration using artificial neural network, Int. J. Innov. Comput. Inf. Control 10 (5) (2014) 1791–1808.
[4] G.A. Anastassiou, Multivariate sigmoidal neural network approximation, Neural Netw. 24 (4) (2011) 378–386.
[5] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, NIPS 1 (2) (2012) 4.
[6] L. Wu, Z. Feng, W. Zheng, Exponential stability analysis for delayed neural networks with switching parameters: average dwell time approach, IEEE Trans. Neural Netw. 21 (9) (2010) 1396–1407.
[7] L. Xie, H. Wei, K. Zhang, Behavioral modeling of nonlinear RF power amplifiers using ensemble SDBCC network, Neurocomputing 154 (2015) 24–32.
[8] J. Martens, I. Sutskever. Learning recurrent neural networks with hessian-free optimization, in: Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011, pp. 1033–1040.
[9] H.R. Maier, A. Jain, G.C. Dandy, et al., Methods used for the development of neural networks for the prediction of water resource variables in river systems: current status and future directions, Env. Model. Softw. 25 (8) (2010) 891–909.
[10] D.A. Asfani, Syafaruddin, M.H. Purnomo, T. Hiyama, Neural network based real time detection of temporary short circuit fault on induction motor winding through wavelet transformation, Int. J. Innov. Comput. Inf. Control 10 (6) (2014) 2277–2293.
[11] X. Su, Z. Li, Y. Feng, L. Wu, New global exponential stability criteria for interval-delayed neural networks, in: Proc. Inst. Mech. Eng. – Part I: J. Syst. Control Eng. 225 (2011) 125–136.
[12] M.T. Musavi, K.H. Chan, D.M. Hummels, K. Kalantri, On the generalization ability of neural network classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 16 (6) (1994) 659–663.
[13] J.M. Valls, I.M. Galvan, P. Isasi, Improving the generalization ability of RBNN using a selective strategy based on the Gaussian kernel function, Comput. Inform. 25 (2006) 1–15.
[14] R. Reed, Pruning algorithms – a survey, IEEE Trans. Neural Netw. 4 (5) (1993) 740–747.
[15] A.P. Engelbrecht, I. Cloete, A sensitivity analysis algorithm for pruning feed-forward neural networks. in: IEEE International Conference on Neural Networks, 1996.
[16] M.G. Augasta, T. Kathirvalavakumar, Pruning algorithms of neural networks – a comparative study, Cent. Eur. J. Comput. Sci. 3 (3) (2013) 105–115.
[17] F. Goksu, F. Ince, I. Onaran, Sparse common spatial patterns with recursive weight elimination, in: Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR), 2011.
[18] C.M. Ennett, M. Frize., Weight-elimination neural networks applied to coronary surgery mortality prediction, IEEE Trans. Inf. Technol. Biomed. 7 (2) (2003) 86–92.
[19] C.J. Merz, P.M. Murphy, UCI Repository of Machine Learning Databases. Univ. California, Dept. Inform. Comput. Sci, Irvine, CA, 1996.
[20] G. Castellano, A.M. Fanelli, M. Pelillo, An iterative pruning algorithm for feedforward neural networks, IEEE Trans. Neural Netw. 8 (3) (1997) 519–531.
[21] T. Kwok, D. Yeung, Constructive algorithms for structure learning in feedforward neural networks for regression problems, IEEE Trans. Neural Netw. 8 (3) (1997) 630–645.
[22] M.J.W. Riley, K.W. Jenkins, C.P. Thompson, A study of early stopping, ensembling, and patchworking for cascade correlation neural networks, IAENG Int. J. Appl. Math. 40 (4) (2010).
[23] M.L. Corradini, V. Fossi, A. Giantomassi, G. Ippoliti, S. Longhi, G. Orlando, Minimal resource allocating networks for discrete time sliding mode control of robotic manipulators, IEEE Trans. Ind. Inform. 8 (4) (2012) 733–745.
[24] M.C. Mozer, P. Smolensky., Skeletonization: a technique for trimming the fat from a network via relevance assessment, Adv. Neural Inf. Process. Syst. I (1989) 107–115.
[25] K. Goedel, Uber formal unentscheidbare satze der principia matematica und verwandter systeme I, Mon. Mat. Phys. 38 (1931) 173–198.
[26] A. Bertoni, B. Palano. Structural Complexity and Neural Networks [M] Neural Nets, Springer, Berlin, Heidelberg, 2002, pp. 190–216.
[27] J.E. Moody, The effective number of parameters: an analysis of generalization and regularization in nonlinear learning system, NIPS 4 (1992) 847–854.
[28] L. Xu, A. Krzyzak, C.Y. Suen, Methods of combining multiple classifiers and their applications to handwriting recognition, IEEE Trans. Syst. Man Cybern. 22 (3) (1992) 418–435.
[29] J. Bailey, T. Manoukian, K. Ramamohanarao, Fast algorithms for mining emerging patterns, Princ. Data Min. Knowl. Discov. (2002) 39–50.
[30] E. Frank, M. Hall, B. Pfahringer, Locally weighted Naive Bayes, in: Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence, 2002, pp. 249–256.
[31] J. Gama , R. Rocha, P. Medas. Accurate decision trees for mining high-speed data streams, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003, pp. 24–27.
[32] W. Pedrycz, D.J. Lee, N.J. Pizzi, Representation and classification of high-dimensional biomedical spectral data, Pattern Anal. Appl. 13 (4) (2010) 423–436.
[33] E.K. Tanga, P.N. Suganthana, X. Yaob, A.K. Qin, Linear dimensionality reduction using relevance weighted LDA, Pattern Recognit. 38 (4) (2005) 485–493.
[34] B. Xu, K. Huang, C.L. Liu., Maxi–Min discriminant analysis via online learning, Neural Netw. 34 (2012) 56–64.
[35] N. Liu, H. Wang., Weighted principal component extraction with genetic algorithms, Appl. Soft Comput. 12 (2) (2012) 961–974.
[36] S.S. Keerthi, K.B. Duan, S.K. Shevade, A.N. Poo, A fast dual algorithm for kernel logistic regression, Mach. Learn. 61 (1–3) (2005) 151–165.
[37] G. Valentini, T.G. Dietterich, Low bias bagged support vector machines, ICML (2003) 752–759.



Liping Xie received the B.S. degree in the Department of Automation, Southeast University, China in 2011. After that, she has been working toward the Ph.D. degree in the School of Automation, Southeast University, China. Her main research interest is neural networks and video-based facial expression recognition.

Haikun Wei received the B.S. degree in the Department of Automation, North China University of Technology, China in 1994, and the M.S. and Ph.D. degrees in the Research Institute of Automation, Southeast University, China in 1997 and 2000, respectively. He was a visiting scholar in RIKEN Brain Science Institute, Japan from 2005 to 2007. He is currently a professor in the School of Automation, Southeast University. His research interest is real and artificial in neural networks and industry automation.

Junsheng Zhao received the B.S. degree in the Department of Mathematics, Liaocheng University, China, in 2003, and the M.S. degree in School of Mathematics, Qufu Normal University, China, in 2006. He has been working in Liaocheng University, China, since 2006 and he has been working toward the Ph.D. degree in the School of Automation, Southeast University, China, since 2011. His main research is in singular learning dynamics of feedforward neural networks.



Kanjian Zhang received the B.S. degree in Mathematics from Nankai University, China in 1994, and the M.S. and Ph.D. degrees in Control Theory and Control Engineering from Southeast University, China in 1997 and 2000, respectively. He is currently a professor in the School of Automation, Southeast University. His research is in nonlinear control theory and its applications, with particular interest in robust output feedback design and optimization control.