Tsinghua Science and Technology

Volume 19 | Issue 1 Article 8

2014

Hierarchical Covering Algorithm

Jie Chen
Department of Computer Science and Technology and Key Lab of Intelligent Computing and Signal Processing, Anhui University, Hefei 230601, China

Shu Zhao
Department of Computer Science and Technology and Key Lab of Intelligent Computing and Signal Processing, Anhui University, Hefei 230601, China

Yanping Zhang
Department of Computer Science and Technology and Key Lab of Intelligent Computing and Signal Processing, Anhui University, Hefei 230601, China

Recommended Citation

Jie Chen, Shu Zhao, Yanping Zhang. Hierarchical Covering Algorithm. Tsinghua Science and Technology 2014, 19(1): 76-81.

TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 08/10 pp76-81
Volume 19, Number 1, February 2014

Hierarchical Covering Algorithm

Jie Chen, Shu Zhao, and Yanping Zhang*

Abstract: The concept of deep learning has been applied to many domains, but the definition of a suitable problem depth has not been sufficiently explored. In this study, we propose a new Hierarchical Covering Algorithm (HCA) method to determine the levels of a hierarchical structure based on the Covering Algorithm (CA). The CA constructs neural networks based on samples' own characteristics, and can effectively handle multi-category classification and large-scale data. Further, we abstract characters based on the CA to automatically embody the feature of a deep structure. We apply CA to construct hidden nodes at the lower level, and define a fuzzy equivalence relation $\overline{R}$ on upper spaces to form a hierarchical architecture based on fuzzy quotient space theory. The covering tree naturally becomes from $\overline{R}$. HCA experiments performed on MNIST dataset show that the covering tree embodies the deep architecture of the problem, and the effects of a deep structure are shown to be better than having a single level.

Key words: deep architecture; hierarchy; fuzzy equivalence relation; covering tree; MNIST dataset

1 Introduction

Over the past few years, there has been a significant interest in "deep" learning algorithms that learn layered, hierarchical representations of high-dimensional data[1,2]. These algorithms sometimes have the important goal of automatically discovering powerful features from raw input data that are independent of application domains. In many fields, traditional algorithms are replaced by deep learning, which achieves better results such as semantic analysis[3], speech recognition[4], and computer vision[5]. There are three advantages of deep architecture: (1) The abstraction is needed along with a deeper level, (2) The feature of the deeper-layer is described by the features of the lower-layer, and (3) The feature hierarchy is a sparse matrix. The human brain is one example of deep architecture in that the cognition process is hierarchical and is abstracted in layers[1]. However, there are also many problems related to deep learning which need to be improved. These problems are because of the shortage of deep levels. In 2008, Bengio[1] raised many open questions about deep learning in learning deep architectures for Artificial Intelligence (AI). One question sought to clarify whether there is a sufficient depth for the computations that are necessary for the AI tasks to approach human-level performance. In 2012, Bengio et al.[2] highlighted that one of the criticisms of artificial neural networks and deep learning algorithms is that the exploration of their configurations and architectures is an art.

Much of this work appears to have been motivated by the hierarchical organization of the cortex, and authors have frequently compared their algorithms' outputs with the oriented simple cell receptive fields. First, an unsupervised learning algorithm and a model with a deep architecture are chosen. Then, unsupervised learning is applied to all layers of the architectures. This is the same architecture used by a supervised train[6-8]. However, to the best of our knowledge, there has been no serious attempt to directly relate (such as through quantitative comparisons) the computations

• Jie Chen, Shu Zhao, and Yanping Zhang are with Department of Computer Science and Technology and Key Lab of Intelligent Computing and Signal Processing, Anhui University, Hefei 230601, China. E-mail: Zhangyp2@gmail.com.
∗ To whom correspondence should be addressed.
Manuscript received: 2013-09-16; revised: 2013-12-20; accepted: 2013-12-26

of these deep learning algorithms. Deep architectures appear to be the natural choice in hard AI tasks involving several sub-tasks, which can be coded into the layers of the architecture. However, the definition of a layer in hierarchical architecture needs to first be clarified.

In this study, the author proposes a new method to determine the levels of hierarchical structure based on the Covering Algorithm (CA). The CA was proposed by Zhang and Zhang[9, 10] as a constructive supervised learning algorithm for the McCulloch-Pitts neural model[11]. The basic idea of CA is to first map all samples in the data set to an n-dimensional sphere S^n. Then, the sphere neighborhoods are utilized to divide the samples. The CA can construct Neural Networks (NNs) based on the samples' own characteristics, and overcome some general drawbacks of traditional NNs. It is very straightforward and can effectively handle multi-category classification and large-scale data, while performing well in many real applications. We further abstract characters based on the CA to embody the feature of a deep structure. The hierarchy is then self-formed.

The author applied CA to construct hidden nodes in the bottom level, and then defined a fuzzy equivalence relation $\overline{R}$ on the upper spaces to form a hierarchical architecture based on fuzzy quotient space theory. The deep architecture comes into being naturally.

2 Covering Algorithm

Covering algorithm is a constructive supervised learning algorithm that maps all samples in the data set to an n-dimensional sphere S^n. The sphere neighborhoods are utilized to divide the samples.

Definition 1 Cover

Assume that the domain of input vectors is a bounded set D of an n-dimensional space. We define a transformation $T: D \to S^n, x \in D$; thus, all points of D are projected upward on S^n by transformation T. Notably, in this situation, a neuron (ω, φ) corresponds to a characteristic function of a "sphere neighborhood" on S^n with ω as its center and $\gamma(\varphi)$ as its radius. This sphere neighborhood can cover some input vectors that belong to the same class t. This sphere is therefore called a cover, $c(t)$.

Definition 2 Covering algorithm

Input: $K = (x_1, y_1), (x_2, y_2), \cdots, (x_m, y_t)$, where K is the set in an n-dimensional Euclidean space, X is the sample set, Y is the domain, m is the number of X, and t is the class number of Y.

Output: The set of covers $C(t)$.

(1) Let $\forall x^i$ be the center of cover C_i^j of class j, where $x^i \in P(j)$, $P(j) = \{x_l \mid \forall x_l \in X, y_l = j\}$, and $I(j) = \{l | \forall x_l \in X, y_l = j\}$;

(2) $d_1(i) = \min\limits_{k \notin I(j)} \{d(x^i, x_k)\}$, $d_2(i) = \max\limits_{k \in I(j)} \{d(x^i, x_k) | d(x^i, x_k) < d_1(i)\}$;

(3) $d(i) = (d_1(i) + d_2(i))/2, \alpha(i) = (d_1(i) - d_2(i))/2$;

(4) Then, C_i^j is the i-th cover of class j which is constructed by x^i and $d(i)$. The center of C_i^j is x^i, the radius is $d(i)$, and the classification interval is $\alpha(i)$;

(5) Finally, the points in C_i^j are deleted from X. The new center is selected to repeat Steps (1)-(4) until $X = \varnothing$.

After this, we can obtain a set of sphere neighborhoods (covers) such that $C(j) = \cup C_i^j$ covers every input $\forall x^l \in p(j)$ and does not cover any input $x^l \notin p(t)$. Therefore, $C(t) = \cup C(j), j = 1, \cdots, t$, which is the output of CA.

Finally, the cover sets obtained above are utilized to construct a three-layer feed forward NNs for classification purposes, as shown in Fig. 1.

The CA transforms the design of the NNs to calculate the set of covers. This algorithm constructs NNs based on the samples' own characteristics that are suitable for large sets of data, and avoids the selection of the structure of NNs and local minimum point. The training speed is rapid.

3 Hierarchy

The concept of a hierarchy has long been used in AI studies. It is a strategy used by humans to deal with complex problems. Deep learning divides the

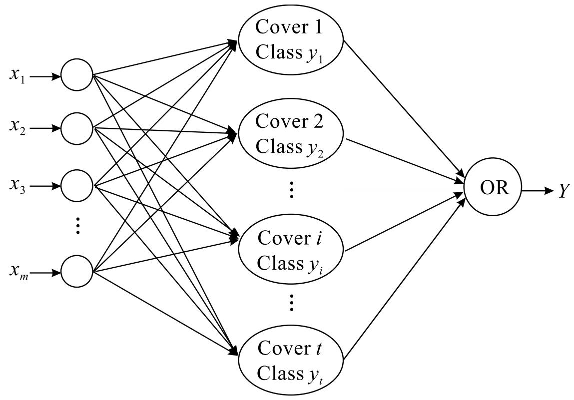

Fig. 1 A three-layer neural network constructed by CA.

problem into many layers which are then trained by unsupervised learning. The problem is abstracted and solved successively. Each layer shows the abstraction in each granular space. Quotient Space Theory (QST) researches the relationship among these quotient spaces (different granular spaces) and the reasoning in those spaces. The model that is built based on QST satisfies two principles: falsity preserving and truth preserving[12]. Truth (falsity) preserving emphasizes that granular transaction does not affect the existence of an answer. QST may therefore be introduced into deep learning to solve its structural problem.

Zhang and Zhang[12] introduced fuzzy set theory into QST to obtain the granular computing model of a fuzzy quotient space. This model builds multi-granular quotient spaces based on different thresholds. The metric distance is normalized to obtain the hierarchical quotient space sequence.

Definition 3 Fuzzy equivalence relation

Assume that $X * X$ is a product space.

For $\overline{R} \in F(X * X)$, if it satisfies

(1) $\forall x \in X, \overline{R}(x,x) = 1$,

(2) $\forall x, y \in X, \overline{R}(x,y) = \overline{R}(y,x)$, and

(3)$\forall x, y, z \in X$, have

$\overline{R}(x,z) \geqslant \sup_y(\min(\overline{R}(x,y), \overline{R}(y,z)))$,

$\overline{R}$ is called a fuzzy equivalence relation on X. The corresponding quotient space is denoted by $[X]$.

Definition 4 Metric function

$\overline{R}$ is a fuzzy equivalence relation on X. $[X]$ is a quotient space as given in Definition 3. Define $\forall a, b \in [X], d(a,b) = 1 - \overline{R}(x,y), \forall x \in a, y \in b$. Then, $d(a,b)$ is a metric (or distance) function on $[X]$.

Definition 5 Fuzzy quotient space

$\overline{R}$ is a fuzzy equivalence relation. Metric space $([X], d)$ defined in Definition 4 is called a quotient space with respect to $\overline{R}$.

Definition 6 Section relation

$\overline{R}$ is a fuzzy equivalence relation. Let

$$\overline{R}_\lambda = \{(x,y) | \overline{R}(x,y) \geqslant d\}, 0 \leqslant d \leqslant 1 \qquad (1)$$

$\overline{R}_d$ is an equivalence relation on X. It is called a section relation of $\overline{R}$.

Let $X(d)$ be a quotient space with respect to $\overline{R}$. From the definitions, we have a proposition below.

Proposition 1 Hierarchy

$0 \leqslant d_2 \leqslant d_1 \leqslant 1 \Leftrightarrow R_{d_1} \subset R_{d_2} \Leftrightarrow X(d_2)$ is a quotient set of $X(d_1)$.

A family of quotient space $\{X(d) | 0 \leqslant d \leqslant 1\}$ is an order-sequence under the inclusion relation of quotient sets. $X(d)$ forms a hierarchical structure with respect to X. Thus, given a fuzzy equivalence relation on X, we have a corresponding hierarchical structure on X.

Example 1: Given $X = \{0,1,2,3,4,5,6,7,8,9\}$ and a fuzzy equivalence relation $\overline{R}$, $\overline{R}$ is represented by a symmetric matrix as follows:

$$\begin{bmatrix}
1.00 & 0.36 & 0.51 & 0.54 & 0.41 & 0.60 & 0.61 & 0.50 & 0.51 & 0.50 \\
0.36 & 1.00 & 0.60 & 0.52 & 0.37 & 0.41 & 0.51 & 0.54 & 0.58 & 0.52 \\
0.51 & 0.60 & 1.00 & 0.59 & 0.52 & 0.41 & 0.59 & 0.59 & 0.69 & 0.59 \\
0.54 & 0.52 & 0.59 & 1.00 & 0.36 & 0.68 & 0.52 & 0.51 & 0.59 & 0.50 \\
0.41 & 0.37 & 0.52 & 0.36 & 1.00 & 0.32 & 0.52 & 0.60 & 0.55 & 0.70 \\
0.60 & 0.41 & 0.41 & 0.68 & 0.32 & 1.00 & 0.60 & 0.42 & 0.57 & 0.52 \\
0.61 & 0.52 & 0.59 & 0.52 & 0.52 & 0.60 & 1.00 & 0.40 & 0.53 & 0.49 \\
0.50 & 0.54 & 0.59 & 0.51 & 0.60 & 0.42 & 0.40 & 1.00 & 0.62 & 0.78 \\
0.51 & 0.58 & 0.69 & 0.59 & 0.55 & 0.57 & 0.53 & 0.62 & 1.00 & 0.73 \\
0.50 & 0.52 & 0.59 & 0.50 & 0.70 & 0.52 & 0.49 & 0.78 & 0.73 & 1.00
\end{bmatrix}.$$

Let $d(i,j) = 1 - \overline{R}(x,y) = 1 - r_{ij}$. Based on the distance, we construct the quotient space shown below.

$X(0) = \{\{0\},\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\},\{9\}\}$,

$X(0.25) = \{\{0\},\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7,9\},\{8\}\}$,

$X(0.3) = \{\{0\},\{1\},\{2\},\{3\},\{4,7,8,9\},\{5\},\{6\}\}$,

$X(0.35) = \{\{0\},\{1\},\{2,4,7,8,9\},\{3,5\},\{6\}\}$,

$X(0.39) = \{\{0,6\},\{1\},\{2,4,7,8,9\},\{3,5\}\}$,

$X(0.4) = \{\{0,3,5,6\},\{1,2,4,7,8,9\}\}$,

$X(0.41) = \{\{0,1,2,3,4,5,6,7,8,9\}\}$,

$X(1) = \{\{0,1,2,3,4,5,6,7,8,9\}\}$.

The hierarchical structure is as shown in Fig. 2.

4 Hierarchical Covering Algorithm (HCA)

The HCA is based on CA where the sample set is entirely covered by many covers. The upper space is constructed instead of all samples with cover centers. We define a fuzzy equivalence relation in the upper space to build a hierarchical quotient space.

The algorithm is described in Algorithm 1.

According to $\overline{R}$, the hierarchical structural for this

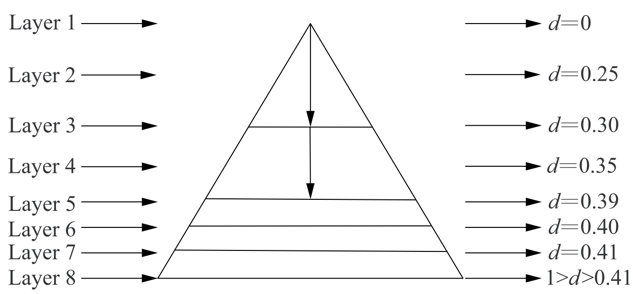

Fig. 2 Hierarchical structural for fuzzy quotient space.

Algorithm 1 Learning HCA

Given data: $X = \{x_1, x_2, \cdots, x_m\}$, where m is the number of samples X.

//Layer 1 training

Train the 1st layer on X using CA $\rightarrow C(j), j = 1, \cdots, t$, where t is the class number of $Y = \{y_1, y_2, \cdots, y_t\}$.

//Upper layers training

Assume $C(j) = C_i^j, i = 1, \cdots, m_j, j = 1, \cdots, t$, where m_j is the cover number of class j. x^i is the center of cover C_i^j.

For $j = 1$ to t do

  $C'_j = \sum_{i=1}^{m_j} C_i^j$, superimpose the centers of the same class covers as the criteria of the class samples.

End for

For $i = 1$ to t do

  For $j = 1$ to t do

    Inner$[i][j] = x^i \times x^j$

  End for

End for

For $i = 1$ to t do

  Ab-inner$[i][j]$ = Inner$[i][j]$/Inner$[i][i]$;

End for

For $i = 1$ to t do

  Sym-inner$[i][j] = \sqrt{\text{Ab-inner}[i][j] \times \text{Ab-inner}[j][i]}$

End for

Find fuzzy equivalence relation $\overline{R}$ on Sym-inner.

Based on Definition 6 and Proposition 1, we obtain the covering tree according to the relation $\overline{R}$.

End

fuzzy quotient space is clear and the covering tree of the samples is constructed. The deep architecture of the covering tree is very clear. Each layer is the abstraction of the lower layer and has the input of the upper abstractive layer.

For examples, assume $X = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ and the fuzzy equivalence relation $\overline{R}$ in Example 1, we can get a covering tree as shown in Fig. 3.

4.1 Adjustment of the unsymmetrical matrix

The matrix of the inner section is largely determined by the characteristics of the samples. For example, the value of sample "1" is clearly smaller than sample "8". This algorithm therefore transforms the inner matrix to an Ab-inner matrix, which is relative. However, the Ab-inner matrix is unsymmetrical. The relation between Ab-inner$[u][v]$ and Ab-inner$[v][u]$ is bidirectional in that it cannot be divided by the equivalence relation. Information mining on weighted bidirected graphs is an important research

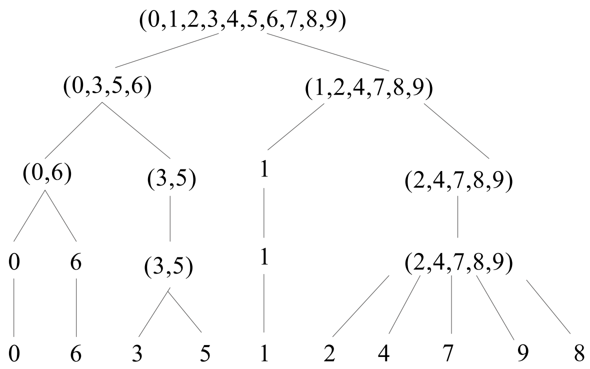

Fig. 3 Covering tree corresponding to Example 1.

topic, and it is important to utilize the weights and direction information of bidirected edges[13]. Therefore, we adjust the unsymmetrical matrix to a symmetrical matrix. Assume $\forall u, v \in Y$, and Ab-inner$[u][v] \neq$Ab-inner$[v][u]$, we define the symmetrical matrix Sym-inner$[u][v]$ = Sym-inner$[v][u]$ as follows:

$$\text{Sym-inner}[u][v] = \sqrt{\text{Ab-inner}[u][v] \times \text{Ab-inner}[v][u]} \tag{2}$$

This adjustment function has two advantages:

(1) Larger Ab-inner$[u][v]$ and Ab-inner$[v][u]$ result in a larger value of the adjustment function.

(2) As Ab-inner$[u][v]$ and Ab-inner$[v][u]$ become closer, the value of the adjustment function increases more quickly.

The adjusted Sym-inner matrix shows the degree of two classes that make contact with each other. The variation of the adjustment function is shown in Fig. 4.

4.2 Learning and testing

The HCA can be trained efficiently based on fuzzy quotient space theory. The first layer CA is trained in a standard CA, which deals with refused samples using the nearest border principle. We use Definition 3 to infer

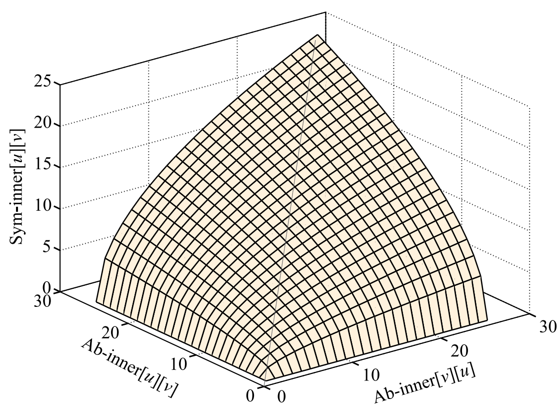

Fig. 4 Variation of the adjustment function.

the fuzzy equivalence relation $\overline{R}$ and the hierarchical structure based on Proposition 1. Algorithm 1 details this training algorithm.

5 Experiment

Here, we present experimental results for the MNIST handwritten digit dataset. The MNIST dataset[14] contains 60 000 training and 10 000 test images of 28×28 handwritten digits. We then use 60 000 images for training and 10 000 images for testing. The original pixel intensities were normalized to lie in the interval $[0, 1]$. In this study, the MNIST dataset was trained using the HCA. We define the fuzzy equivalence relation to obtain the hierarchical structure, as shown in Fig. 2. Then, the covering tree is constructed, as shown in Fig. 3. The error rates of the HCA and CA are shown in Table 1.

The recognition speed of the samples is upgraded through the covering tree. Assume that the layer number of the covering tree is L. The time complexity is given by $O(L) < O(N)$, where N is the number of covers (leaf nodes). The experiments also show that the error rate of the covering tree is lower than the CA. Although the error rate of the MNIST dataset is higher than the best result[14], the intension of the HCA is to construct the covering tree and reveal secrets of the deep structure to guide the description of structures of other problems.

6 Conclusions

In this study, we proposed a new HCA method to determine the levels of a hierarchical structure based on the CA. A fuzzy equivalence relation was defined to form a hierarchical architecture based on fuzzy quotient space theory. Then, we constructed a covering tree which naturally embodies the deep architecture. The HCA experiments on the MNIST dataset are better than those of the CA, and the experiments show that the precision decreases rapidly for lower layers, depending on the characteristics of the MNIST dataset employed. In future, we aim to improve the training precision and introduce other datasets to examine the effect of our proposed algorithm.

Table 1 Experimental results.

		Error rate (%)
CA		4.19
HCA	Layer 2	1.50
	Layer 3	1.59
	Layer 4	1.65
	Layer 5	3.01

Acknowledgements

This work was supported by the National Key Basic Research and Development (973) Program of China (No. 2007CB311003), the National Natural Science Foundation of China (Nos. 61073117 and 61175046), the Young Science Foundation of Anhui University (No. KJQN1118), the Outstanding Young Talents Higher Education Institutions of Anhui Province (No. 2011SQRL129ZD).

References

[1] Y. Bengio, Learning deep architectures for AI, *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1-127, 2009.

[2] Y. Bengio, A. Courville, and P. Vincent, Representation Learning: A review and new perspectives, Technical report, arXiv:1206.5538, 2012.

[3] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, Joint learning of words and meaning representations for open-text semantic parsing, presented at the 15th International Conference on Artificial Intelligence and Statistics, La Palma, Canary Islands, 2012.

[4] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription, presented at the 29th International Conference on Machine Learning, Edinburgh, Scotland, 2012.

[5] Y. Boureau, J. Ponce, and Y. Lehern, A theoretical analysis of feature pooling in vision algorithms, presented at the 27th International Conference on Machine Learning, Haifa, Israel, 2010.

[6] D. Ciresan, U. Meier, and J. Schmidhuber, Multi-column deep neural networks for image classification, Technical report, arXiv:1202.2745, 2012.

[7] G. E. Dahl, D. Yu, L. Deng, and A. Acero, Context dependent pre-trained deep neural networks for large vocabulary speech recognition, *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 20, no. 1, pp. 33-42, 2012.

[8] J. J. Kivinen and C. K. Williams, Multiple texture Boltzmann machines, presented at the 27th International Conference on Artificial Intelligence and Statistics, La Palma, Canary Islands, 2012.

[9] L. Zhang and B. Zhang, A forward propagation learning algorithm of multilayered neural networks with feed-back connections, (in Chinese), *Journal of Software*, vol. 8, no. 4, pp. 252-258, 1997.

[10] L. Zhang and B. Zhang, A geometrical representation of McCulloch Pitts neural model and its applications, *IEEE Transactions on Neural Networks,* vol. 10, no. 4, pp. 925-929, 1999.

[11] W. S. McCulloch and W. Pitts, A logical calculus of the ideas immanent in neurons activity, *Bulletin of Mathematical Biophy,* vol. 5, no. 4, pp. 115-133, 1943.

[12] L. Zhang and B. Zhang, *Theory and Applications of Problem Solving–The Quotient Space Granular Computation Theory and Applications*. Beijing, China: Tsinghua University Press, 2007.

[13] D. H. Qiu, Y. Li, and X. Xu, An approach to detecting clusters from weighted bidirected graphs, (in Chinese), *Journal of Chinese Computer Systems*, vol. 33, no. 7, pp. 1568-1571, 2012.

[14] The MNIST dataset, http://yann.lecun.com/exdb/mnist/, 2013.

Jie Chen is currently an assistant professor and a PhD candidate in the Department of Computer Science and Technology at Anhui University. She received her MS degree from Anhui University in 2006. Her main research interests include deep learning, quotient space theory, granular computing, and machine learning.

Shu Zhao received her PhD degree in computer science from Anhui University in 2007. She is currently an associate professor in the Department of Computer Science and Technology, Anhui University, China. Her current research interests include quotient space theory, granular computing, and machine learning.

Yanping Zhang received her PhD degree in computer science from Anhui University in 2003. She is currently a professor in the Department of Computer Science and Technology, Anhui University. Her primary research interests include computational intelligence, quotient space theory, artificial neural networks, and machine learning.