Unsupervised Classification of Multiple Attributes via Autoassociative Neural Network

Reina Kamioka†, Kouji Kurata††, Kazuyuki Hiraoka†, Taketoshi Mishima†

†Dept. of Information and Computer Science, Saitama University, 255 Shimo-Okubo, Saitama-shi, 338-8570, JAPAN
Tel.+81-48-858-3723, Fax.+81-48-858-3723
††Dept. of Mechanical System Engineering, Ryukyu University, 1 Senbaru, Nishihara, Nakagami, 903-0129, JAPAN

E-mail: {kamioka,hira,mishima}@me.ics.saitama-u.ac.jp

Abstract This paper proposes unsupervised classification of multiple attributes via five-layer autoassociative neural network with bottleneck layer. In the conventional methods, high dimensional data are compressed into low dimensional data at bottleneck layer and then feature extraction is performed (Fig.1). In contrast, in the proposed method, analog data is compressed into digital data. Furthermore bottleneck layer is divided into two segments so that each attribute, which is a discrete value, is extracted in corresponding segment (Fig.2).

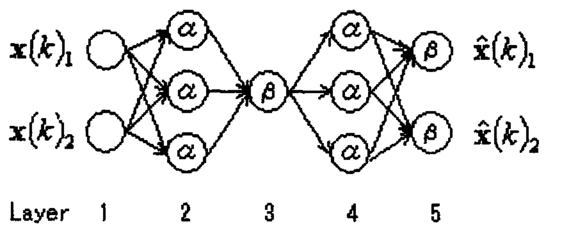

Fig.1. Conventional autoassociative neural network with bottleneck layer architecture. In the figure, α is sigmoid cells. β is linear or sigmoid cells.

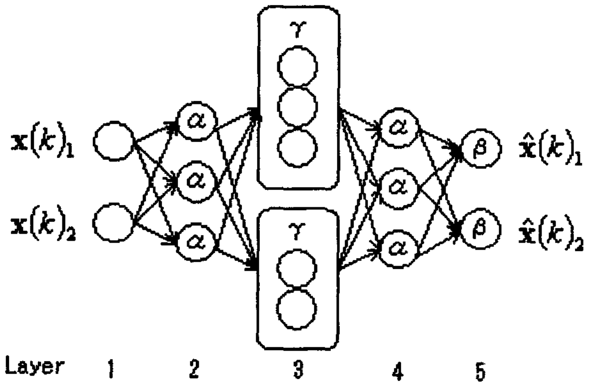

Fig.2. Proposed autoassociative neural network with bottleneck layer architecture. For details to Fig.1 about α and β. γ is softmax cells.

1. Introduction

Data in the real world have multiple attributes. For example, face image have multiple attributes, such as "person", "expression" and so on. It is easy for human to interpret these attributes naturally, while it is hard for computer since such structure cannot acquire by conventional clustering methods.

In the present study, the authors try to classify unlabeled data with multiple attributes. Namely, the proposed system receives unlabeled data with multiple attributes and it divides the data into groups according to estimated attributes.

It uses five-layer autoassociative neural networks with bottleneck layer. In the conventional research, it has been studied that feature extraction of input vector from expression of bottleneck layer. Practical training method to autoassociative neural network is that output and input become as equal as possible. In the conventional methods, high dimensional data is compressed into low dimensional data at bottleneck layer (Fig.1). For detail of the proposed method is abstract.

2. Problem setup

Unlabeled data

$$\mathbf{x}(k) \in R^N (k = 1, \cdots K) \quad (1)$$

are presented to the learning system.

Though each datum $\mathbf{x}(k)$ has M attributes, which are not presented. In this paper, the case M=2 is studied and these two attributes are represent as $s(k)$ and $c(k)$.

$$s(k) \in \tilde{S} = \{1,\ldots,S\}, c(k) \in \tilde{C} = \{1,\ldots,C\} \quad (2).$$

Extraction to general M is straightforward. Of course, this is impossible in general because the information on $s(k)$ and $c(k)$ are not given at all. In the present study, it is assumed that distribution of $\mathbf{x}(k)$ has a certain structure which reflects the hidden attributes $s(k)$ and $c(k)$ (Fig.3).

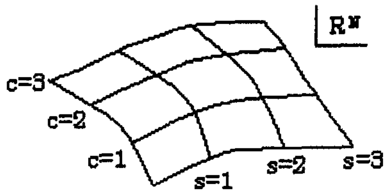

Figure 3. Conceptual diagram of the assumed distribution of $\mathbf{x}(k)$. $s(k)$ and $c(k)$ represent hidden attributes of $\mathbf{x}(k)$.

3. Proposed method

3.1 Approach

Let us consider a massage passing game between two players A and B.

- The procedure of this game is as follows.

(1) Target vector $\mathbf{x}(k)$ is presented to A (Fig.4).

(2) A is allowed to tell only two words $\hat{s}(k)$ and $\hat{c}(k)$ to B (Fig.5).

(3) B is required to guess $\mathbf{x}(k)$ (Fig.6).

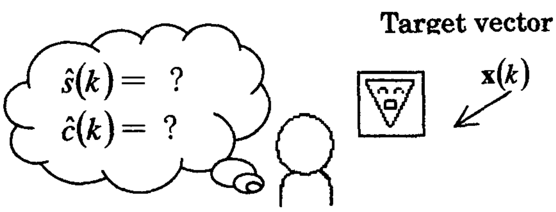

Fig.4. Presentation of target vector $\mathbf{x}(k)$ to A.

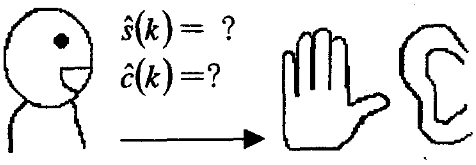

Fig.5. Transmission of restricted information from A to B

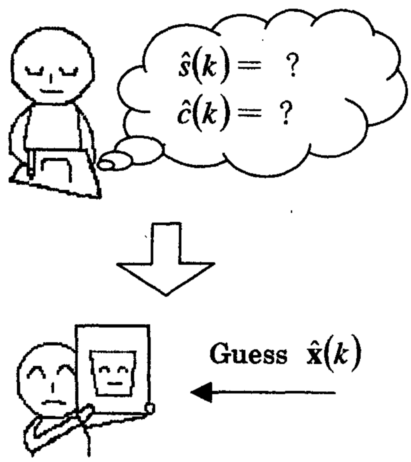

Fig.6. Extrapolation of $\mathbf{x}(k)$ by B.

This game is performed for the data $\mathbf{x}(1),\ldots,\mathbf{x}(K)$.

We expect that two words $\hat{s}(k)$ and $\hat{c}(k)$ correspond to hidden attributes $s(k)$ and $c(k)$ when training of A and B is completed. Note that it does not mean $s(k) = \hat{s}(k)$ and $c(k) = \hat{c}(k)$. The result is successful if the correspondence between $s(k)$ and $\hat{s}(k)$ ($c(k)$ and $\hat{c}(k)$) is one-to-one, because they are equivalent in the sense that same grouping of data is obtained from them.

3.2 Implementation by neural network

We use a five-layer autoassociative feed-forward neural network (Fig.7). Black squares in Fig.7 are competitive layer and softmax function. Softmax is defined by

$$v_i = \frac{\exp \beta \mathbf{x}_i}{\sum_j \exp \beta \mathbf{x}_i} \quad (i, j = 1 \cdots k) \quad (3)$$

for input $\mathbf{x}$ and output $\mathbf{v}$.

$s(k)$ corresponds to $\hat{s}(k)$ and $c(k)$ corresponds to $\hat{c}(k)$, respectively. $\mathbf{v} = (v_1 \cdots v_L)$, where L = S or C, and β is a positive constant. This network is trained so that output and input are as equal as possible. After training, the presented data $\mathbf{x}(1),\ldots,\mathbf{x}(K)$ are divided according to $\hat{s}(k)$ and $\hat{c}(k)$. We expect that this classification corresponds to true attributes $s(k)$ and $c(k)$.

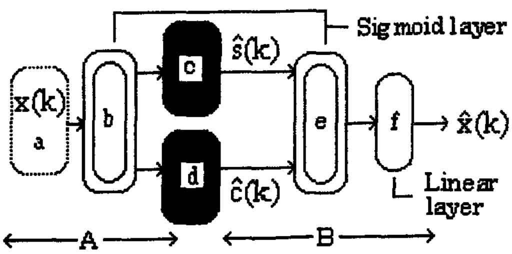

Fig.7. Five-layer autoassociative feed forward neural network with bottleneck layer architecture.

3.3 Correspondence to the analogy of message passing game

The former half of Fig.7 corresponds to the player A of massage passing game. The latter half corresponds to B.

(1) The target vector $\mathbf{x}(k)$ is input to the first layer.
(2) Two message words are obtained from competitive layers c and d.
(3) The estimated data $\hat{\mathbf{x}}(k)$ is obtained from last layer f.

3.4 Training procedure to avoid spurious solution

Thanks to adoption of softmax function (3) instead of archetypal hardmax in the competitive layers, standard routine of backpropagation [5] can be use for training of the proposed network.

In this training, two types problem occurred. First problem is premature convergence. It can be coped with by brute force approach. In fact, we repeat trials with different initial weight of neurons and select the best result. "Best" is measured based on the true $s(k)$ and $c(k)$. The other problem is spurious solutions, and it is more awkward. In some trials, representation error ($\sum_{k=1}^{K}\|\hat{\mathbf{x}}(k)-\mathbf{x}(k)\|^2$) is almost zero while acquired division is far from the desired division based on the true $s(k)$ and $c(k)$. The cause is softmax function (3). In spurious solutions, analog value of (3) is used subtly and small representation error is archived. This is illegal in our message passing game. And then, we use two steps of leaning method in order to eliminate spurious solutions.

- Two steps of leaning method

(step1) All of neural network learned

(step2) The former half of neural network is fixed and then softmax of competitive layer is changed into hardmax and neural network learns only the latter half of neural network.

We repeat trials with different initial weight of neurons. Then the best result in last result of step2 is adopted. Result obtained from step2 is not spurious solutions. This is because digital values of $\hat{s}(k)$ and $\hat{c}(k)$ forced by hardmax. In constant, if expression obtained from step1 is appropriate, the trial will also obtain good expression from step2.

4. Experimental verification of the proposal technique

4.1 Target inputs and parameters

We have applied the proposed method to a small basic task of $K=6$ input data on $N=2$ dimensional space (Fig.8). It is natural for human to divide these data into three groups according to $\mathbf{x}_1$, and into two groups according to $\mathbf{x}_2$.

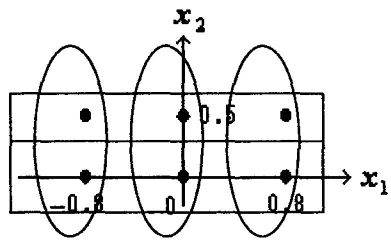

Fig.8. Small basic task of $K=6$ input data on $N=2$ dimensional space.

learning rate	0.1
momentum coef.	0.9
number of epochs of training	20000
number of neurons of b layer	3
number of neurons of e layer	2

Table.1. Parameter setup of proposed learning method step1. For details is section 3.4 and Fig.7.

learning rate	0.1
momentum coef.	0.9
number of epochs of training	5000
number of neurons of b layer	3
number of neurons of e layer	2
weight of c and d layers	150

Table.2. Parameter setup of step2. For details is section 3.4 and Fig.7.

4.2 Experimental result

By the proposed method, 3200 trials have been performed (Table.3). We have adopted trial case when representation error has been minimum (minimum value = 0.027). As a result, these natural classifications have been obtained automatically.

As a problem of this proposed method, it has been proved that a huge number of trials are necessary. Results of trials that have been obtained natural classification about the other attribute are described below. The rate of a correct answer of classification about $s(k)$ is 3.7%, about $c(k)$ is 7.1%. However, the rate of a correct answer of classification about both attributes ($s(k)$ and $c(k)$) is 0.3%.

As the measure to this, appropriate quasi-optimization such as genetic algorithm or taboo search may help reduce the problem. Also combing correct trials about the other attribute may help reduce the problem.

number of emsemble	3200
number of correct answer about *s(k)*	118
number of correct answer about *s(k)*	228
number of correct answer about *s(k),c(k)*	1

Table.3. Experimental result by the proposed method.

5. Conclusion

In the present study, we have proposed a method to classify two kinds of attributes through unsupervised learning by autoassociative neural network. And the natural classification has been obtained in the experiment to a small basic task of $K=6$ input data on $N=2$ dimensioned space.

A future subjects are measure against partial solution and the reducing of trial. At the present, the proposed method has a serious problem that a huge number of trials are necessary because of premature convergence. As the measure to this, appropriate quasi-optimization such as genetic algorithm or taboo search may help reduce the problem. Also combing correct trials about the other attribute may help reduce the problem.

References

[1] HIRAOKA Kazuyuki, SHIGEHARA Takaomi, MIZOGUCHI Hiroshi, MISHIMA Taketoshi, and YOSHIZAWA Shuji: "On The Overfitting of The Five-Layered Bottleneck Network", Proceedings of 6th International Conference on Neural Information Processing, 234-239, 1999.

[2] T. Kurita, H. Shimai, T. Mishima and T. Takahashi: "Self-Organization of Viewpoint Dependent Face Representation by the Self-Supervised Learning and Viewpoint Independent Face Recognition by the Mixture of Classifiers", Proceedings of IAPR Workshop on Machine Vision Applications (MVA2000), pp.319-322, 2000.

[3] Malthouse, E. C: "Limitations of Nonlinear PCA as Performed with Generic Neural Networks", IEEE Trans. on Neural Networks, Vol. 9, No. 1, pp. 165-173, 1998.

[4] M.A.Kramer: "Nonlinear principal component analysis using neural networks", AIChE J., vol.37, no2, pp.233-243, 1991.

[5] D.E Rumelhart, G.E. Hinton, R.J. Williams, "Learning internal representations by error propagation", D. Rumelhart and J. McClelland, editors. Parallel Data Processing, Vol.1, Chapter 8, the M.I.T. Press, Cambridge, MA 1986 pp. 318-362.

This work has been partly supported by JSPS (Japan Society for the Promotion of Science), 14580405.