IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 27, 2018, accepted December 4, 2018, date of publication December 12, 2018,
date of current version January 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2886425

Application of Clustering Analysis in Brain Gene Data Based on Deep Learning

YINA SUO[1], TINGWEI LIU[2], XUEYONG JIA[3], AND FUXING YU[1]

[1]Information Engineering Institute, North China University of Science and Technology, Tangshan 063000, China
[2]School of Mathematics, Shandong University, Jinan 250100, China
[3]College of Electrical Engineering, North China University of Science and Technology, Tangshan 063000, China

Corresponding author: Fuxing Yu (yufuxing626@126.com)

This work was supported by the National Natural Science Foundation of China under Grant 51504080.

ABSTRACT In the current research, cluster analysis has become a very good way to obtain biological information by analyzing the brain gene expression data. In recent years, many experts have used improved traditional clustering algorithm and a new clustering algorithm to mine brain gene expression data. First, the random Forest method is used to preprocess high-dimensional and high-complexity brain gene expression data. Then, a clustering model based on deep learning is proposed, and a clustering algorithm is implemented by using deep belief network (DBN) and fuzzy c-means algorithm (FCM). This model makes full use of the generality of unsupervised learning of deep learning and clustering technology, combines the advantages of deep learning with clustering, and makes clustering effect better and more convenient for clustering high-dimensional data.

INDEX TERMS Deep belief network, fuzzy c-means algorithm, unsupervised learning, brain gene data clustering.

I. INTRODUCTION

Since the 1980s, with the rapid increase of biological data and genome sequencing data, a new discipline, bioinformatics, has emerged. Bioinformatics refers to an interdisciplinary subject formed by the interpenetration and interaction of information science, computer science and biology [1]. Although it involves many disciplines, its scope is very clear. It is accompanied by genome research. Bioinformatics, on the one hand, is the collection, collation and service of massive data. On the other hand, it discovers new rules from the analysis of data [2], so as to use these data to achieve the purpose of research. The study of the nature of evolution is of great significance and opens up a new way for the diagnosis and prevention of human diseases.

There are many studies on brain gene databases at home and abroad. Liu Qing and Yang Xiaotao (2005) [3] used the signal-to-noise ratio (SNR) method in the stage of feature selection to classify the samples containing the first 10, 20, 30, 40, 50 and all brain genes by support vector machine, and then classify the samples after feature selection. The accuracy rate is significantly higher than that of direct classification of all original data samples. Youwei et al. (2010) [4] speeded up the operation efficiency of feature selection of SVM-RF method. Based on SVM-RF, SFS method was combined with sequence forward selection. The empirical results show that SVM-RF&SFS has lower average test error rate and less time-consuming in feature selection. Xialin (2018) and others used HPV gene chip detection technology to detect HPV genotypes in cervical squamous cell carcinoma tissues, which promoted the establishment of HPV genotype database in cervical cancer tissues. Wang Wei and Liu Hong (2010) [5] combined support vector machine (SVM) with genetic algorithm (GA) to analyze brain gene data and improve the classification ability of SVM.

In recent years, breakthroughs have been made in the field of machine learning. In 2006, Professor Geoffrey Hinton and his student Ruslan Salakhutdinov of the University of Toronto in Canada put forward the concept of deep learning in Science, which initiated the research upsurge of deep learning. Deep belief network (DBN) is a kind of deep learning algorithm. Deep learning (DL) refers to the use of a multi-layer network architecture model for data feature calculation, signal transformation, pattern classification and so on. Liu Nian *et al.* [6] studied the process of leaf image recognition, and used deep belief network training algorithm to establish classification framework, which improved the characteristics of network recognition, such as shorter time, stronger stability and higher accuracy. In this paper, based on

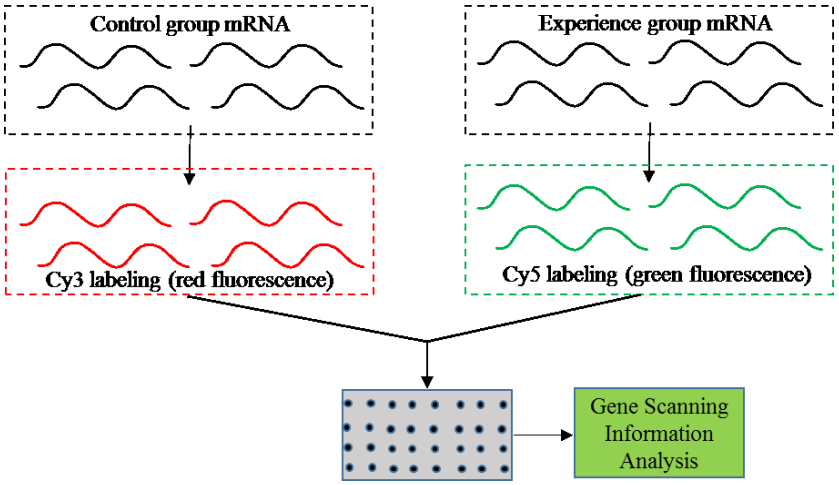


FIGURE 1. Gene chip technology.

deep belief network (DBN), the parameters of fuzzy C-means are optimized to cluster high-dimensional brain gene expression data to improve the accuracy of clustering.

II. BRAIN GENE DATA

A. A SUMMARY OF THE CHARATERISTICS OF BIOLOGICAL BRAIN GENE DATABASE

Biological brain gene database is a collection of data and information records for the results of biological gene research when carrying out biological gene research. In the process of biotechnology research, because of the complexity of biological gene itself and the composition and sequence of biological gene, the types and types of databases for collecting and recording biological gene research data also have great complexity. Gene research data information recorded and preserved in the biological brain gene database includes not only the measurement and research data information for each segment of the biological gene [7], but also the arrangement information for different segments of the brain gene. Normally, in the biological brain gene database, the number of recorded brain gene field data information is between 20 and 60, while the number of recorded genome composition and arrangement data information is usually more than 10,000. Therefore, according to the data information stored in the biological brain gene database, the data information of biological gene research in the biological brain gene database not only comes from a wide range of sources [8], but also has a wide variety of types of data information, and the record format of brain gene data information in the database is also diversified. The amount of information stored in brain gene research data is large.

B. BRAIN GENE CHIP TECHNOLOGY

Gene chip [9], [10], or DNA microarray technology, oligonucleotide chip, is a new biotechnology developed with the development of computer technology and genome sequencing technology in the 1990s. It can detect the expression level of tens of thousands of brain gene transcripts with high throughput, so as to systematically detect the surface of intracellular m RNA molecules. It is possible to reach the state and speculate on cell function [11]. DNA microarray technology has a wide range of applications, including discovery of new brain genes, analysis of temporal and spatial characteristics of brain gene expression, detection of differential brain gene expression, diagnosis and treatment of diseases, drug research and so on.

The nucleotide sequences called probes are arranged densely on solid-phase carriers such as silicon, glass, polypropylene or nylon membranes. When the fluorescent labeled nucleic acid sequences in solution complement the corresponding nucleic acid probes on the brain gene chip, a set of sequences is obtained by determining the position of the probe with the strongest fluorescence intensity. The whole process includes chip preparation, sample preparation, hybridization reaction, signal detection and result analysis, so as to obtain important information about brain gene expression in samples and form brain gene expression profiles [12]. Gene expression data can be represented by a matrix or vector, in which the numerical size of matrix or vector elements represents the expression level of the brain gene. Figure 1 shows the process of measuring the expressed data:

III. BRAIN GENE DATA PREPROCESSING BASED ON RANDOM FOREST ALGORITHMS

Random forest is a very popular and efficient algorithm in machine learning. It can be used not only to solve regression problems, but also to classify and discriminate. It can also calculate the importance of variables of each feature to screen features. Random forest is proposed by Breiman, which is essentially composed of many decision trees [13]. Compared with single decision tree, its prediction accuracy has been greatly improved. Random forests also have many outstanding advantages, such as: not easy to over-fit; not affected by

data dimensions, can handle high-dimensional data, and so on, so it has been widely used in practice.

A. DECISIO TREE

Decision tree is a kind of tree structure decision graph with many branches [14]. It consists of a root node, many intermediate interior nodes, many directed edges and many leaf nodes. Figure 2 below is a simple decision tree:

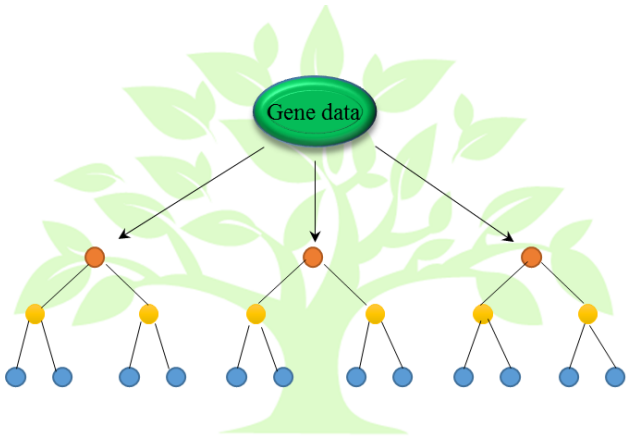


FIGURE 2. Simple decision tree.

The decision tree adopts the rule of joint distribution of multi-part nodes to unify the management of complex problems, and realizes the integration and dimensionality reduction of gene data.

B. RANDOM FOREST

Random forest refers to the construction of multiple decision trees to classify the samples [15], [16]. The classification results of each decision tree are recorded as a vote. Finally, the category of the most votes is found. Random forest classifies the samples into that category.

The concrete process of random forest construction is to establish many decision trees. Suppose the training set S, the test set N, and the number of features of each sample is F. The bootstrap method is used to randomly extract S samples from the training set S when building each decision tree. Because of the random sampling, some samples are repeatedly extracted, some samples are not extracted, and the samples not extracted are recorded as OOB (out of bag), which is mainly used to calculate the importance of features. The samples extracted by bootstrap method are used as training samples of decision tree. The feature used for splitting decision tree is also extracted randomly from all feature F. At this time, the extraction method is in the form of no-return, and the number of features extracted is mtry<F. That is to say, the feature of splitting decision tree each time is to select the optimal feature from the mtry feature. Repeat the above operations to get random forest.

The main parameters of random forest are characteristic number mtry and decision tree number ntree. The parameter mtry is generally taken as $\sqrt{F}$, $\frac{1}{2}\sqrt{F}$ and $2\sqrt{F}$ in experience (where F is the characteristic number of samples), and the parameter ntree generally chooses larger values in experience.

The main differences between random forest and decision tree are as follows: on the one hand, when the decision tree in random forest chooses the best feature to split in the splitting process [17], it chooses the best feature to split from the mtry feature extracted without playback. In this way, the difference between decision trees in random forests can be increased, and the diversity of the system can be improved, so as to improve the classification performance. On the other hand, random forests do not have pruning steps, which does not lead to over-fitting. This is due to the randomness of sample selection and feature selection in the establishment of each decision tree, and it contains many decision trees.

The bootstrap method is used to extract samples from random forests when building decision trees. The samples that are not extracted are recorded as OOB (Samples sampled by bootstrap method), which is used to calculate the importance of each feature. The process of calculating the characteristic importance of stochastic forests is as follows. For a given feature V:

(1) Using the OOB data of each decision tree to calculate the error of out-of-bag data of each decision tree: $error1_i$, $i = 1, 2, \ldots, ntree$.

(2) The characteristic V of the corresponding out-of-pocket data of each class decision tree is changed into random noise, and the out-of-pocket error of each decision tree is calculated by repeating step 1: $error1_i$, $i = 1, 2, \ldots, ntree$.

(3)Importance of computing feature V: $VI = \frac{1}{ntree}\sum_{i=1}^{ntree}(error1_i - error2_i)$.The bigger the importance VI, the smaller the change of the feature, which makes the error rate change greatly. Therefore, the more important this variable is for correct classification.

The random forest method is used to pre-process the brain genes, so that the useless information in the brain gene data is filtered, and then the effective brain gene data in the brain gene database is extracted and characterized.

IV. FUZZY CLUSTERING MODEL BASED ON DAN

Gene expression data has the characteristics of high dimensionality, high complexity and difficult to identify. In this paper, a fuzzy clustering model based on DBN is introduced to cluster the existing brain gene expression data [18], which provides a basis for brain gene identification and classification. As shown in Figure 3, the flow chart of DBN-based fuzzy clustering algorithm is presented.

C-means clustering algorithm (FCM algorithm) is a clustering algorithm based on objective function [19], [20]. In the late 1960s, Ruspini first defined the fuzzy partition of sets. Based on the concept of fuzzy partition of sets defined by Ruspini, Dunn analyzed the hard c-means clustering algorithm. The hard c-means clustering algorithm is written into the shape of objective function, and the most intuitive and

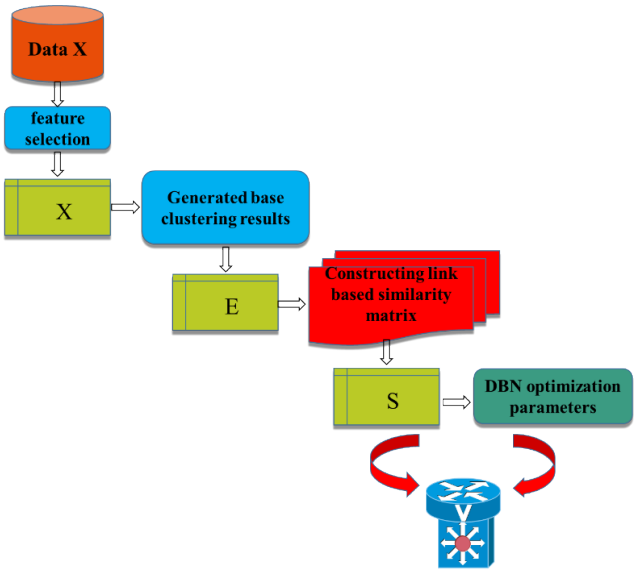


FIGURE 3. Fuzzy clustering algorithm flow based on DB.

simple fuzzy c-means clustering algorithm is obtained by analogy to the fuzzy case. Then Bezdek did more general work and proved the convergence of the algorithm [21]. Nowadays, clustering algorithms based on objective function have developed vigorously, and have produced fuzzy C-family algorithm, fuzzy c-shell algorithm, relational and fuzzy number c-means clustering algorithm. They realize the clustering of linear, planar, hyper planar, ellipsoid, shell, relational and fuzzy data respectively.

A. DATA SET PARTITION

Given that data set $X = \{x_1, x_2, \cdots, x_n\} \subset R^s$ is a set of finite observation samples of N patterns in the pattern space, $x_k = \{x_{k1}, x_{k2}, \cdots, x_{kn}\}^T \in R^s$ is the feature vector or pattern vector of the observation sample x_k, corresponding to a point in the feature space, x_{kj} is the assignment of the j-dimensional feature of the feature vector x_k. Clustering analysis of a given sample set X is to form a c-Partition of X.

If the membership function $\mu_{ik} = \mu_{xi}(x_k)$ is used to represent the membership relationship between sample x_k and subset $x_i (1 \le i \le c)$, then μ_{ik} is the characteristic function of subset X_i in hard c partition, obviously $\mu_{ik} \in \{0, 1\}$. The hard c partition of X can also be represented by a membership function, i.e. a matrix $U = [\mu_{ik}]_{c\times n}$ composed of the Eigen function values of c subsets. Ruspini extends the membership function μ_{ik} from {0, 1} binary to [0, 1] interval by using the theory of fuzzy sets, thus extending the concept of hard c partition to fuzzy c partition. Therefore, the fuzzy c partition space of X is:

$$M_{fc}\left\{U \in R^{cn} \middle| \mu_{ik} \in [0,1], \Lambda k; 0 < \sum_{k=1}^{n} \mu_{ik} < n, \Lambda i\right\} \quad (1)$$

Because the uncertainty degree of samples belonging to each column can be obtained by fuzzy partition, the uncertainty description of categories can be established, so it can reflect the real world more objectively. In the result of partition, the fuzzy partition can also indicate the circumference of partition, the connection and discreteness of different partition blocks, so more detailed information can be mined.

B. CLUSTERING OBJECTIVE FUNCTION

In order to find reasonable classification results among many possible classifications, it is necessary to establish reasonable clustering criteria. In hard classification, the clustering criterion commonly used is the sum of least square deviation [22].

Assuming that $U = [\mu_{ik}]_{c\times n}$ is a hard partition matrix, $p_i (i = 1, 2, \cdots, c)$ is a representative vector of class i or a clustering prototype vector, $p_i(p_{i1}, p_{i2}, \cdots, p_{is}) \in R^s$. The objective function of cluster analysis is defined as:.

$$\begin{cases} J_1(U,P) = \sum_{i=1}^{c} (\sum_{x_i e X_i} (d_{ik})^2) \\ s.t\ U \in M_{hc} \end{cases} \quad (2)$$

In the formula, d_{ik} denotes the distortion between sample x_k in class i and typical sample p_i in class i. It is often measured by the distance between two vectors. $J_1(U, P)$ represents the sum of squares of errors between various samples and their typical samples. Using μ_{ik}, $J_1(U, P)$ can also be expressed as:

$$\begin{cases} J_1(U,P) = \sum_{k=1}^{n} \sum_{i=1}^{c} \mu_{ik} (d_{ik})^2 \\ s.t\ U \in M_{hc} \end{cases} \quad (3)$$

Clustering criterion is to find the best group pair (U, P) so that $J_1(U, P)$ is the smallest under the condition of satisfying constraint $\mu_{ik} \in M_{hc}$.

C. PARAMETER ANALYSIS OF FUZZY C-MWANS CLUSTERING ALGORITHM

According to the mathematical model of fuzzy clustering, for a given set of samples, it is easy to get a fuzzy c partition $U = \{\mu_{ik} | 1 \le i \le c, 1 \le k \le n\}$ by fuzzy clustering analysis [23]–[25]. However, to ensure meaningful partitioning, it is necessary to define appropriate partitioning criteria according to the needs of the problem, such as the commonly used similarity (difference) criterion $D(\cdot)$. Assuming that each fuzzy set $\tilde{X}_i (1 \le i \le c)$ forms a pattern p_i (often referred to as clustering prototype), the similarity between sample x_k and fuzzy subset $\tilde{X}_i$ can be measured by the distortion $d_{ik} = D(x_k, p_i)$ between sample x_k and clustering prototype p_i to determine the fuzzy partition matrix U. However, the prototype of clustering can not be known beforehand, so it needs to be formed gradually in the process of clustering.

In order to ensure that the clustering results can make "objects clustered by clusters", the objective function of fuzzy clustering is constructed by minimizing the distortion between each class of samples and this kind of pattern [26], and then the optimal fuzzy c partition $U = [\mu_{ik}]$ and pattern

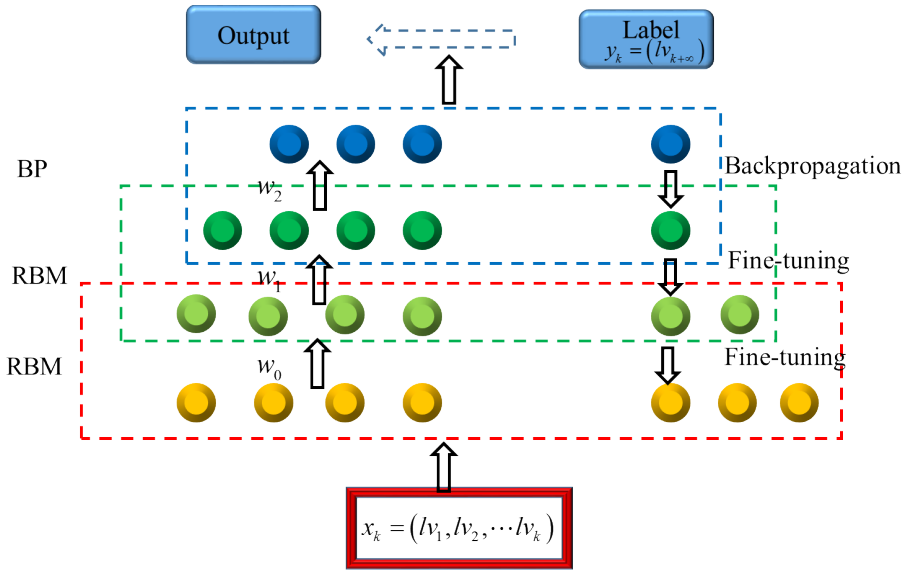


FIGURE 4. DBN structure mode.

$P = \{p_i, 1 \le i \le c\}$ of each class of samples are obtained by optimizing the objective function. The common objective function of fuzzy clustering is as follows: it is a constrained non-linear function.

$$\min_{(U,P)} \left\{ J_m = \sum_{i=1}^{c} \sum_{x_i e X_i} (\mu_{ik})^m D^2(x_k, p_i), s.t. f(\mu_{ik}) \in C \right\} \quad (4)$$

In the formula, C is the constraint condition and m is the weighted index. The objective function of fuzzy clustering is determined by parameter set $\{U, D(\cdot), P, m, X\}$.

D. IMPROVEMENT OF FUZZY CLUSTERING MODEL PARAMETERS BY DEEP BELIEF NETWORK

The parameter selection measure of the fuzzy c-means clustering algorithm is always in a fuzzy state. In this paper, the deep belief network is introduced to optimize the parameters of the clustering algorithm (the upper and lower thresholds a and b of the fuzzy partition matrix U, the weighted index m, the number of the fuzzy clustering X). The following is the basic idea and the establishment process of the deep belief network model [19], [27].

Deep belief network is a neural network with multiple hidden layers published by Hinton *et al.* [28]. It is stacked by a probability model called restricted Boltzmann machine (RBM). RBM is a classical neural network. The visible layer and the hidden layer units of the network are connected with each other, but there is no connection between the same layer. The hidden unit can get the high-order correlation characteristics of the input visible unit. Compared with sigmoid network, the training of RBM parameters is easier, so the learning of RBM is very important for the application of DBN.

The DBN structure model is shown in Figure 4, which consists of several bottom-up RBM stacks. The output of RBM training in the former layer becomes the input of the next layer. In order to enable each layer to learn more about the features of the previous layer, it is necessary to train DBN layer by layer repeatedly, so Hinton et al. proposed a fast and effective DBN learning algorithm. The learning process of DBN is carried out according to the training sequence of RBM one by one, that is, training layer by layer. The basic idea is as follows:

The training sample set is randomly selected and then directly put into the network to train the first RBM, which enables the hidden layer's meridians to extract the important features of the input sample data. That is, the first DBN hidden layer of DBN, and then use the data features acquired before as input data for the next layer, followed by the training of the second RBM, according to the above steps continue to repeat the training of each level of RBM in DBN until all the RBM layers have been trained.

DBN component RBM is a training model without tutors, so it is not necessary to select labeled sample data manually when pre-training RBM [29]. Because the CD algorithm in RBM model can reduce the pre-training time and accelerate the convergence of the model. In order to better enable each layer of RBM to better learn the characteristics of the previous layer, DBN uses a layer-by-layer unsupervised training algorithm called greedy learning. The running process of greedy learning algorithm is described as follows:

Input: training sample data.

Output: DBN network model parameters.

Step 1: Training the bottom RBM model, if the energy of the model reaches balance or the number of training times reaches the pre-set iteration number, then stop training; otherwise continue training the RBM model.

Step 2: Use the trained parameters of the RBM model of the upper layer to calculate the state value of the hidden layer, and use the state value of the hidden layer as the input data of

the next RBM, continue training the RBM of the layer until the stopping condition of step 1 is satisfied.

Step 3: Repeat step 2 until all levels have completed training.

Step 4: Use the mean square error of the actual value and the predicted value as the objective function, and then use the back propagation algorithm (algorithm) to fine-tune the model parameters of the whole network to obtain better network parameters.

In the layer-by-layer greedy learning method, an optimization algorithm called Wake-Sleep is used. Wake-Sleep algorithm idea is as follows: Wake stage: using the learned weights to generate the training data needed for the next layer can be seen as bottom-up recognition direction; Sleep stage: using weights to reconstruct data can be seen as top-down generation direction. Therefore, DBN can be regarded not only as a recognition model, but also as a generation model.

E. RESTRICTED BOLTZMANN MACHINES

RBM is the core component of DBN. It consists of visual layer (V) and hidden layer (H), and the nodes in each layer are not connected with each other. All nodes are random values of 0 or 1, and the total probability distribution $P(V, H)$ obeys Boltzmann distribution. Compared with sigmoid reliability network, the learning of RBM weights is more simple and easy. The weights of the generated model are obtained in advance by using the greedy layer-by-layer learning method without tutors [30]–[32]. The learning process is to map the visible vector value to the hidden layer unit, then reconstruct the visual layer unit by using the hidden layer unit, and then map the visual layer unit to the hidden layer unit again by using the visual layer unit. The process of repeated execution of the above steps is called Gibbs sampling.Constrained Boltzmann

Machine Structure is shown in Figure 5.

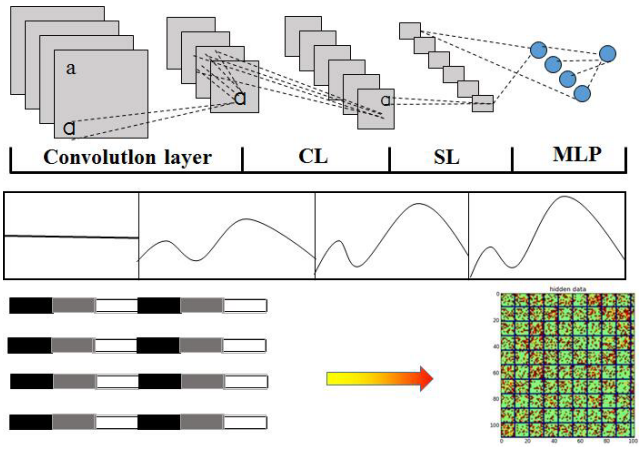


FIGURE 5. Constrained Boltzmann machine structure.

Assuming that RBM has n visual units and m hidden units, vectors v and h are used as the states of visual units and hidden units respectively.v_i is the state of the i visual unit and h_j is the state of the j hidden unit.W_{ij} is the connection weight between the visual unit i and the hidden unit j, a_i is the offset of the visual unit i and b_j is the offset of the hidden unit.

Because the nodes in the same layer are independent and independent of each other, all hidden layer nodes are independent of each other under the premise of knowing V. So the probability distribution of the j node in the hidden layer can be expressed as:

$$\begin{cases} P(H\,|V\,) = \prod p_i(v_i\,|h\,) \\ P(h_j = 1\,|v\,) = f(b_j + \sum w_{ij} v_i) \\ P(h_j = 0\,|v\,) = 1 - p(h_j = 1\,|v\,) \end{cases} \tag{5}$$

Similarly, under the premise that the hidden layer H is known, all the nodes in the visible layer are conditionally independent, so the probability distribution of the i node in the visible layer can be expressed as follows:

$$\begin{cases} P(H\,|V\,) = \prod p_i(v_i\,|h\,) \\ P(v_j = 1\,|h\,) = f(a_j + \sum w_{ij} h_j) \\ P(v_i = 0\,|h\,) = 1 - p(h_j = 1\,|h\,) \end{cases} \tag{6}$$

$f(z)$ is a Sigmoid activation function and can also be a normalization factor. Where $\prod p_i(v_i\,|h\,)$ A is the measure of global distance.

$$f(z) = \frac{1}{1 + e^{-z}} \tag{7}$$

Therefore, for a pair of given states (v, h), assuming that RBM is considered as a system, its energy function can be expressed as:

$$E(v, h\,|\theta\,) = -\sum_{i=1}^{n} a_i v_i - \sum_{j=1}^{n} b_j h_j - \sum_{i=1}^{n} \sum_{j=1}^{n} v_i w_{ij} h_j \tag{8}$$

In the above formula, $\theta = \{w_{ij}, a_i, b_j\}$ is the parameter of RBM, and they are all real numbers. So as long as all the parameters are known, the joint probability distribution of (v, h) can be obtained according to the energy function of Formula (9), as shown in Formula (10):

$$p(v, h\,|\theta\,) = \frac{e^{-E(v,h|\theta\,)}}{Z_\theta} \tag{9}$$

$$Z_\theta = \sum_{v,h} e^{-E(v,h|\theta\,)} \tag{10}$$

In the formula, Z_θ is a normalization factor, and it can also be called partition function.

Although Gibbs sampling can obey the random sample defined by RBM, it still needs more sampling steps, which will reduce the learning efficiency of RBM [33]. Hinton *et al.* published an algorithm called Contrast Divergence CD, which can speed up RBM learning. The difference between the contrast divergence algorithm and Gibbs sampling is that at the beginning, the state of the visual unit is changed into a training sample. At the same time, the state of all the hidden layer units is calculated. After all the hidden layer units are calculated, the probability of the

TABLE 1. Gene expression database.

ID_REF	GSM 51763	GSM 51764	GSM 51765	GSM 51766	GSM 51767	GSM 51768
A28102_at	26.4	36.5	61.5	43.2	61.5	28.4
AB000324_at	12.3	35.6	18.5	17.3	24.6	31.5
AB000407_at	8.2	7.6	8.3	8.4	9.5	6.2
A28114_at	25.4	29.6	31.5	68.4	59.1	48.6
AB000408_at	63.2	35.8	16.7	16.2	35.1	65.8
AB000415_at	31.8	29.4	61.7	54.3	32.5	43.6
AB000320_at	12.6	23.6	15.6	74.3	62.8	43.5
AB000243_at	30.4	13.4	27.6	32.5	66.4	20.3
AB000210_at	25.1	16.8	27.4	63.5	61.3	25.6
AB000307_at	21.6	45.6	15.6	64.8	65.2	42.3
AB000105_at	21.2	12.5	63.2	68.2	47.6	56.1
AB000220_at	41.6	10.5	29.3	66.2	45.6	31.4
AB000190_at	14.5	26.4	24.5	45.5	51.3	15.9
AB000420_at	14.6	35.1	35.9	83.1	63.5	41.9
AB000447_at	42.3	68.3	48.6	52.6	49.5	42.6
AB000340_at	23.5	33.6	25.2	73.5	68.3	41.5
AB000460_at	204.1	216.8	280.4	216.5	246.3	353.1

i visual unit being 1 is calculated to complete the visual layer. A reconstruction. Therefore, the updating function of each parameter can be expressed as an expression (14) when the algorithm of random gradient rise is used to maximize the value of the logarithmic likelihood estimation function on the learning data.

$$\begin{aligned} \Delta w_{ij} &= (\varepsilon \langle v_i, h_j \rangle_{data} - \langle v_i, h_j \rangle_{recon}) \\ \Delta a_{ij} &= (\varepsilon \langle v_i \rangle_{data} - \langle v_i \rangle_{recon}) \\ \Delta b_j &= (\varepsilon \langle h_j \rangle_{data} - \langle v_j \rangle_{recon}) \end{aligned} \tag{11}$$

In the formula, ε is the learning rate, $\langle\rangle_{data}$ is the distribution of training set of observation data, and $\langle\rangle_{recon}$ is the distribution of model expression after further reconstruction.

V. SIMULATION AND SIMULATION

Some data from China National Gene Bank, an online resource warehouse for brain gene expression data, high density oligonucleotide array (HAD), hybrid membrane (filter) and brain gene expression sequence analysis (SAGE) were selected and many types of brain gene expression data were accepted, registered and archived. Firstly, the data are filtered and processed by using methods such as de-negative, data conversion, outlier processing and data filling. Finally, 3200 brain gene data are selected as data sample set to analyze the problem. The results are shown in Table 1.

The first column in the table is the probe used in the acquisition of brain gene expression data. A sample describes the processing conditions, operation of the sample and the abundance measurements of each element. The stochastic forest model is used to repair the lost data, transform the data and extract the features of the expressed data sets. Then the parameters of the fuzzy c-means clustering algorithm are optimized as shown in figure 6 to figure 9.

As shown in the figure, the parameters of the fuzzy c-means clustering algorithm are optimized by deep belief network. The upper and lower thresholds a and b of the fuzzy partition matrix U are 5.1 and 6.8 respectively, the weighted index m is 0.182, and the number of the fuzzy clustering X is 595 groups.

Random forest method is used to pre-process data such as feature extraction, and then fuzzy c-means clustering is applied to the dimension-reduced data. As shown in Figure 10-11, the clustering result without deep belief network optimization is obviously different from the clustering

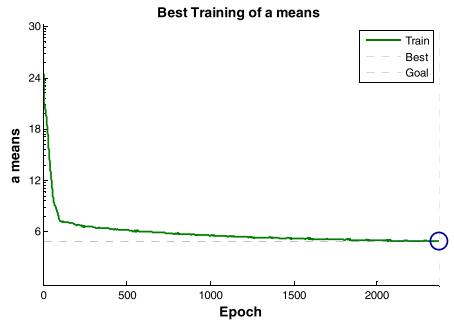


FIGURE 6. Parameter a optimization process.

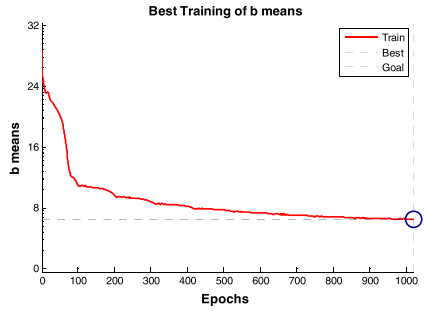


FIGURE 7. Parameter B optimization process.

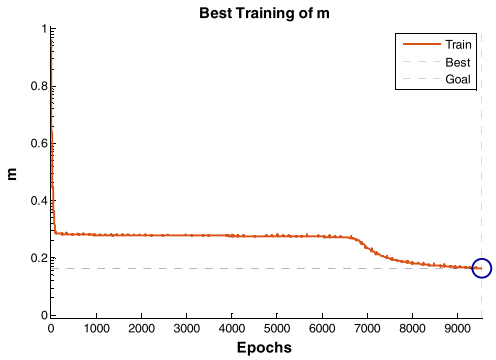


FIGURE 8. Parameter m optimization process.

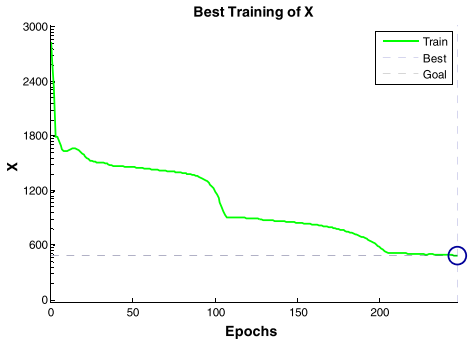


FIGURE 9. Parameter X optimization process.

result after optimization. Specifically, under the condition of the same brain gene expression data, there are no clear boundaries of brain gene classification in the clustering

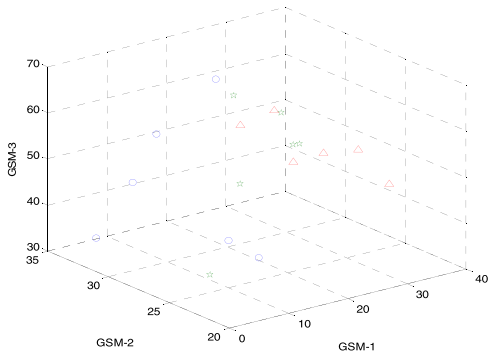


FIGURE 10. Fuzzy c- means clustering results.

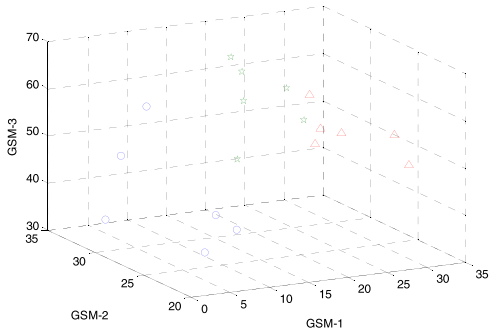


FIGURE 11. Fuzzy c- mean clustering results after DBN optimization.

results of non-optimized fuzzy c-means clustering, and no good classification lines can be obtained, while the optimized clustering boundaries are clearer. Under the condition of the same brain gene expression data, other kinds of brain genes were not mixed into the optimized fuzzy c-means clustering homologous genes, which effectively improved the accuracy of clustering.

VI. CONCLUSION

Starting from the establishment and citation of brain gene database, a stochastic forest model for extracting eigenvalues from gene expression data was established. After preprocessing the selected brain gene database data, a fuzzy c-means clustering model based on deep neural network is established. Based on some data from the National Gene Bank of China, the following conclusions are drawn.

(1) Because of the basic characteristics of brain gene data, feature extraction is needed before processing brain gene data. The stochastic forest model established in this paper can efficiently extract the eigenvalues of complex data in brain gene database.

(2) By optimizing the model parameters of the fuzzy c-means clustering model through deep belief network, the accuracy of the optimized model is obviously improved, and the classification of brain genes is clearer, which provides a theoretical basis for the classification of brain gene expression data.

(3) Classification of complex brain gene data has always been a difficult problem for people to study brain gene expression. In this paper, 3200 sets of data are selected to optimize

clustering algorithm with deep belief network, and satisfactory clustering results are obtained. But in the practical application, processing a larger amount of data puts forward higher requirements for the performance of computers and the efficiency of algorithms. In the future research, we will focus on the optimization of the algorithm, simplify the operation process and improve the operation speed.

REFERENCES

[1] K. Blin, M. H. Medema, R. Kottmann, S. Y. Lee, and T. Weber, "The antiSMASH database, a comprehensive database of microbial secondary metabolite biosynthetic gene clusters," *Nucleic Acids Res.*, vol. 45, pp. D555–D559, Jan. 2017.

[2] X. Sheng *et al.*, "MTD: A mammalian transcriptomic database to explore gene expression and regulation," *Briefings Bioinf.*, vol. 18, pp. 28–36, Jan. 2016.

[3] L. Qing and Y. Xiaotao, "Data analysis method of microarray gene expression based on support vector machine," *Minicomput. Syst.*, vol. 26, no. 3, pp. 363–366, 2005.

[4] Y. Wei, L. Shutao, and T. Mingkui, "Gene selection method based on SVM-RFE-SFS," *Chin. J. Biomed. Eng.*, vol. 29, no. 1, pp. 93–99, 2010.

[5] W. Wei and L. Hong, "Gene microarray analysis based on genetic algorithm and support vector machine," *China Tissue Eng. Res.*, vol. 14, no. 17, pp. 3099–3103, 2010.

[6] A.-M. Yang, X.-L. Yang, J.-C. Chang, B. Bai, F.-B. Kong, and Q.-B. Ran, "Research on a fusion scheme of cellular network and wireless sensor for cyber physical social systems," *IEEE Access*, vol. 6, pp. 18786–18794, 2018.

[7] S. Jiang, M. Lian, C. Lu, Q. Gu, S. Ruan, and X. Xie, "Ensemble prediction algorithm of anomaly monitoring based on big data analysis platform of open-pit mine slope," *Complexity*, vol. 2018, Aug. 2018, Art. no. 1048756.

[8] E. Clough and T. Barrett, "The gene expression omnibus database," *Methods Mol. Biol.*, pp. 93–110, 2016.

[9] K. Jaakson *et al.*, "Genotyping microarray (gene chip) for the *ABCR (ABCA4)* gene," *Hum. Mutation*, vol. 22, no. 5, pp. 395–403, 2003.

[10] S. J. Evans *et al.*, "Evaluation of Affymetrix Gene Chip sensitivity in rat hippocampal tissue using SAGE analysis," *Eur. J. Neurosci.*, vol. 16, no. 3, pp. 409–413, 2002.

[11] X. He *et al.*, "Newborn screening of genetic mutations in common deafness genes with bloodspot-based gene chip array," *Amer. J. Audiol.*, vol. 27, pp. 57–66, Mar. 2018.

[12] W. Wu, P. Cheng, J. Lyu, and Z. Zhang, "Tag Array gene chip rapid diagnosis anti-tuberculosis drug resistance in pulmonary tuberculosis—A feasibility study," *Tuberculosis*, vol. 110, pp. 96–103, May 2018.

[13] Y. Han, J. Li, X.-L. Yang, W.-X. Liu, and Y.-Z. Zhang, "Dynamic prediction research of silicon content in hot metal driven by big data in blast furnace smelting process under Hadoop cloud platform," *Complexity*, vol. 2018, Oct. 2018, Art. no. 8079697.

[14] O. S. Arabeyyat, "Information security model using decision tree for Jordanian public sector," *Int. J. Electron. Secur. Digit. Forensics*, vol. 10, no. 3, pp. 228–241, 2018.

[15] J. Jia, Z. Liu, X. Xiao, B. Liu, and K.-C. Chou, "pSuc-Lys: Predict lysine succinylation sites in proteins with PseAAC and ensemble random forest approach," *J. Theor. Biol.*, vol. 394, pp. 223–230, Apr. 2016.

[16] E. Lombaert, M. Ciosi, N. J. Miller, T. W. Sappington, A. Blin, and T. Guillemaud, "Colonization history of the western corn rootworm *(Diabrotica virgifera virgifera)* in North America: Insights from random forest ABC using microsatellite data," *Biol. Invasions*, vol. 20, no. 3, pp. 665–677, 2018.

[17] L. Huang *et al.*, "Longitudinal clinical score prediction in Alzheimer's disease with soft-split sparse regression based random forest," *Neurobiol. Aging*, vol. 46, pp. 180–191, Oct. 2016.

[18] S. D. Mai and T. N. Long, "Multiple kernel approach to semi-supervised fuzzy clustering algorithm for land-cover classification," *Eng. Appl. Artif. Intell.*, vol. 68, pp. 205–213, Feb. 2018.

[19] V. Bhatia and R. Rani, "DFuzzy: A deep learning-based fuzzy clustering model for large graphs," *Knowl. Inf. Syst.*, vol. 57, no. 1, pp. 159–181, Oct. 2018.

[20] W. Zhang and W. Zang, "A fuzzy density peaks clustering algorithm based on improved dna genetic algorithm and K-nearest neighbors," in *Proc. Int. Conf. Intell. Sci. Big Data Eng.* Cham, Switzerland: Springer, 2018, pp. 467–487.

[21] H. Fu, Z. Li, Z. Liu, and Z. Wang, "Research on big data digging of hot topics about recycled water use on micro-blog based on particle swarm optimization," *Sustainability*, vol. 10, no. 7, p. 2488, 2018.

[22] L. Kang, H. L. Du, H. Zhang, and W. L. Ma, "Systematic research on the application of steel slag resources under the background of big data," *Complexity*, vol. 2018, Oct. 2018, Art. no. 6703908.

[23] X. Liu, H.-K. Chen, B.-Q. Huang, and Y.-B. Tao, "Optimal sizing for wind/PV/battery system using fuzzy *C*-means clustering with self-adapted cluster number," *Int. J. Rotating Machinery*, vol. 2017, Sep. 2017, Art. no. 5142825.

[24] A. Feizollah, N. B. Anuar, and R. Salleh, "Evaluation of network traffic analysis using fuzzy *C*-means clustering algorithm in mobile malware detection," *Adv. Sci. Lett.*, vol. 24, no. 2, pp. 929–932, 2018.

[25] R. J. Kuo, T. C. Lin, Ferani E. Zulvia, and C. Y. Tsai, "A hybrid metaheuristic and kernel intuitionistic fuzzy *C*-means algorithm for cluster analysis," *Appl. Soft Comput.*, vol. 67, pp. 299–308, Jun. 2018.

[26] E. Maag and Y. Li, "Discrete-valued gravity inversion using the guided fuzzy *C*-means clustering technique," *Geophysics*, vol. 83, no. 4, pp. G59–G77, 2018.

[27] R. Davoodi and M. H. Moradi, "Mortality prediction in intensive care units (ICUs) using a deep rule-based fuzzy classifier," *J. Biomed. Inform.*, vol. 79, pp. 48–59, Mar. 2018.

[28] J. Wu, L. Zhang, S. Yin, H. Wang, G. Wang, and J. Yuan, "Differential diagnosis model of hypocellular myelodysplastic syndrome and aplastic anemia based on the medical big data platform," *Complexity*, vol. 2018, Nov. 2018, Art. no. 4824350.

[29] X. Shi *et al.*, "Power transformer fault classifying model based on deep belief network," *Power Syst. Protection Control*, 2016.

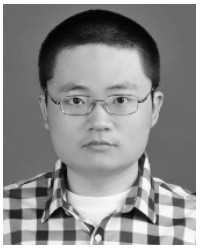
[30] S. Ting and G. Guohua, "Application of greedy learning method based on optimal path forest classification in CBIR system," *J. Sichuan Univ. (Eng. Sci. Ed.)*, vol. 48, no. 5, pp. 135–142, 2016.

[31] J. Chen, S. Cheng, H. Xie, L. Wang, and T. Xiang, "Equivalence of restricted Boltzmann machines and tensor network states," *Phys. Rev. B, Condens. Matter*, vol. 97, no. 8, p. 085104, 2018.

[32] G. Bresler, F. Koehler, A. Moitra, and E. Mossel. (2018). "Learning restricted Boltzmann machines via influence maximization." [Online]. Available: https://arxiv.org/abs/1805.10262

[33] F. Liu, Y. Liu, D. Jin, X. Jia, and T. Wang, "Research on workshop-based positioning technology based on Internet of Things in big data background," *Complexity*, vol. 2018, Oct. 2018, Art. no. 7875460.

YINA SUO was born in 1980. She received the master's degree from the School of North China University of Science and Technology, in 2012. She is currently a Lecturer with the North China University of Science and Technology. Her interest is in the research of multimedia data compression and the development of mobile games in Android system.



TINGWEI LIU was born in Tangshan, Hebei, China, in 1998. He is currently pursuing the master's degree with the School of Mathematics, Shandong University, majoring in information and computing science.

XUEYONG JIA was born in Xingtai, Hebei, China, in 1998. He is currently pursuing the master's degree with the North China University of Science and Technology, majoring in electrical engineering and automation. His main research interests include automatic control, data mining, writing, and typesetting.

FUXING YU was born in 1979. He received the master's degree from the University of Science and Technology, Beijing, in 2010. He is currently a Lecturer with the North China University of Science and Technology. His interest is the research of big data processing and the development of mobile games in Android system.