AMSE JOURNALS-AMSE IIETA publication-2017-Series: Advances B; Vol. 60; N°1; pp 210-223
Submitted Jan. 2017; Revised March 15, 2017, Accepted April 15, 2017

Classification Method for Uncertain Data based on Sparse De-Noising Auto-Encoder Neural Network

*Jijiang Yu, **Yuwen Huang, ***Chunying Liu

*Department of Computer and Information Engineering, Heze University, China, Heze 274015, Shandong, (hzxy_yjj@163.com)

** Key Laboratory of computer Information Processing, Heze University, China, Heze 274015, Shandong, (hzxy_hyw@163.com)

*** Department of Computer and Information Engineering, Heze University, China, Heze 274015, Shandong, (Corresponding Author E-mail: lcy810204@163.com)

Abstract

Due to its importance in machine learning, pattern recognition, and many other applications, uncertain data mining has attracted much attention. This paper proposes a classification method for uncertain data based on a sparse de-noising auto-encoder neural network. Firstly, a hyper-ellipsoid convex model is used to describe the uncertain interval vector, and give an approach for uncertain data classification based on an interval uncertainty support vector machine. Secondly, this paper introduces a sparse de-noising auto encoder neural network, which can convert high-dimension data into low-dimensional characteristic space. Finally, this paper establishes a three layered auto-encoder neural network, and whose deep structure is tuned with stochastic gradient descent parameters fine-tuned by layer greedy pre-training and back propagation. Experimental results show that this proposed method creats better classification accuracy, and has stronger robustness for noise parameter, so it is effective for uncertain data classification.

Key words

Uncertain data, auto-encoder neural network, hyper-ellipsoid support vector machine, sparse coding.

1. Introduction

As big data approaches, there are more and more data problems related to acquisition, storage, and processing. Due to the various influences of data noise, data loss, transmission delay, and inaccurate measurement, a large number of the data attributes are uncertain. Traditional data mining methods are designed for deterministic data, but uncertain data mining considers more factors in comparison with the traditional data mining method, and their classification models are more complex, so the traditional technology cannot be applied. In recent years, it has been a popular research subject as to how to design mining methods and models for uncertain data. Paper [1] proposed a methodology to classify uncertain data by a new probabilistic distance measure both for uncertain data and a group of uncertain objects. Paper [2] proposed an Extreme Learning Machine algorithm for the classification of uncertain data which is uniformly distributed. Paper [3] proposed a Robust Bayes classifier with feature selection and classification tasks for uncertain data. Paper [4] proposed a new uncertain item set mining algorithm to consider importance of items. Paper [5] presented a classification method for uncertain data by the belief functions and K-nearest neighbors. Paper [6] gave a new approach for mining incomplete data with singleton, subset and concept probabilistic approximations. The auto-encoder neural network based on deep learning is a new hotspot of dimensional reduction techniques, and it decreases irrelevant and redundant data by unsupervised learning, which leads to a reduction in its dimension and low-level features, and combines low-level features to form high abstract representation. The auto-encoder neural network is widely discussed since Hinton presented it in 2006, and it is able to obtain the initialized weights of each layer by preliminary training of unlabeled data, so the data features can efficiently be extracted. Paper [7] proposed an effective patch extraction method for halftone image classification based on a sparse auto-encoder. Paper [8] proposed a sparse auto-encoder-based deep neural network approach for induction motor fault diagnosis. Paper [9] proposed a new deep neural network architecture for numeral recognition. Paper [10] proposed a novel unsupervised approach based on a de-noising auto-encoder for automatic acoustic novelty detection. A sparse auto-encoder neural network is added to a sparse punitive term, and the activeness of its hidden layer nodes is limited. The de-noising auto-encoder neural constructs the original signal by the noise data, so it is more robust. De-noising auto-encoder neural networks were enhanced based on the idea of sparsity, which enabled abstract features of sparse representation to become more effective for data classification. The traditional surface learning algorithm is able to extract the surface character, and the auto-encoder network has more powerful modeling capability as a typical deep networks model, so this paper proposes a classification method for uncertain data based on a sparse de-noising auto-encoder neural network.

2. Interval uncertain support vector machine

2.1 Uncertain data

There is broad application area on uncertain data in scientific research and engineering practice, and the interval number is used to describe uncertain data. The interval number is usually defined as a pair of ordered real numbers.

$U^I = \left[U^L, U^H\right] = \left\{U \mid U \in R, U^L \le U \le U^R\right\}$, U^R is upper bound, and U^L is lower bound. If $U^R = U^L$, U is the certain data. Interval numbers are alos represented as the midpoint and radius . $U^I = \left\langle U^C, U^R \right\rangle = \left\{x \mid U^C - U^W \le x \le U^C + U^W\right\}$, $U^C = \left(\dfrac{U^L + U^R}{2}\right)$, $U^R = \left(\dfrac{U^R - U^L}{2}\right)$.

On the basic of the convex model theory, use the hyper-ellipsoid convex model to describe the uncertain interval vector.

$U = \left[u_1, u_2, ..., u_m\right] \in I\left(R^m\right)$. $E\left(U\right) = \left\{u \in R^m \mid (u - U^C)^T G_U^{-1}(u - U^C) \le 1\right\}$.

G_U **is the characteristic matrix of the hyper-ellipsoid convex model, and it is the positive definite matrix constituted by the interval vector** $U^R = \left[u_1^R, u_2^R, ..., u_m^R,\right]$.

2.2 Interval uncertainty support vector machine

In order to obtain a better classification for interval uncertainty data, Paper [11] puts forward an interval uncertainty hyper-ellipsoid support vector machine, which is used to get the smallest hyper-sphere that contains most samples of a class, so it can divide the class samples from others. The constraint rules of the optimal hyperplane in interval uncertainty support vector machine are as follows:

$$\min_{R,a} R^2 + C\sum_{i=1}^{l}\xi_i$$
$$s.t \left\|\varphi\left(x_i\right) - a\right\|^2 \le R^2 + \xi_i$$
$$x_i \in E_{x_i}, \xi_i \ge 0, i = 1, 2, ..., n$$

(1)

a is the spherical point, R is the spherical radius, ξ is relaxation factor, and C is the penalty factor. The uncertainty vector x_i is a sample in hyper-ellipsoid convex set. E_{X_i} is a sample of uncertain range.

The optimization of the hyper-sphere in the interval uncertainty support vector machine problem can be transformed into the deterministic constraints programming as follow :

$$\min_{R,a} \max_{x_i \in E_{X_i}} R^2 + C\sum_{i=1}^{l}\xi_i$$
$$s.t \min_{x_i \in E_{X_i}} R^2 + \xi - \varphi(x_i) - a^2 \geq 0$$
$$x_i \in E_{x_i}, \xi_i \geq 0, i = 1,2,\ldots,n$$

(2)

The above optimal mathematical model nests convex optimization with two layers, with the outer designing the vector optimization on the one hand, While on the other hand, the inner optimization looks for the most unfavourable response of the uncertain objective function and constraints. In order to reduce the computational complexity of the convex models, two layers with nested optimization are turned into an alternating iterative search for the upper and lower two layers, and the converted optimization is as follows :

$$\min_{R(t),a(t),\xi_i(t)} F(t) = R(t)^2 + C\sum_{i=1}^{l}\xi_i(t)$$
$$s.t\ R(t)^2 + \xi_i - \left\|\varphi(x_i(t)) - a(t)\right\|^2 \geq 0$$
$$\xi_i \geq 0, i = 1,2,\ldots,n$$

(3)

The solution of the dual problem is as follows.

$$\max_{\lambda_i(t)} n F(t) = \sum_{i=1}^{n}\lambda_i(t) K(x_i(t), x_i(t)) -$$
$$\sum_{i,j=1}^{n}\lambda_i(t)\lambda_j(t) K(x_i(t), x_j(t))$$
$$s.t \sum_{i=1}^{n}\lambda_i(t) = 1$$
$$0 \leq \lambda_i(t) \leq C, i = 1,2,\ldots,n$$

(4)

The optimization of the layer below is as follows.

$$\min_{x_i(t+1)} R(t)^2 + \xi_i(t) - \left\|\varphi(x_i(t+1)) - a(t)\right\|^2$$
$$s.t\ x_i(t+1) \in E_{X_i}, i = 1,2,\ldots,n$$

(5)

The solutions of the optimal hyperplane are transformed into the optimization solution of the upper and lower layers, until convergence of the optimal solution is obtained. If the non-linear convex set is directly solved, the computational complexity is too large, so Taylor expansion is used to obtain the approximate optimal solution.

3. Sparse de-noising auto-encoder neural network

3.1Network structure of auto-encoder neural network

The auto-encoder neural network is an unsupervised learning neural network so far as possible as it restructures the input signal, and initializes the network weight through layer by layer training. At the same time, the auto-encoder neural network can fine-tune the network parameters by a hidden back propagation algorithm, and optimize the overall performance. The auto-encoder neural network is as follows in figure 1:

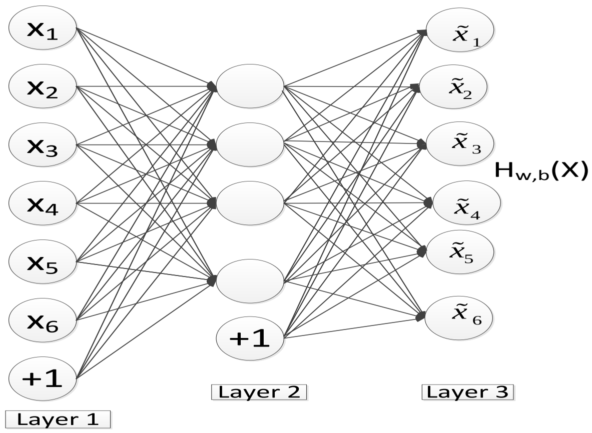

Fig.1. Structure auto encoder neural network

In order to get the network parameters (w,b) in the stack auto-encoder neural network, the network hidden layer is trained by the unlabelled training samples, and obtains its parameters (w_1,b_1). The output of the hidden layer is the input of the output layer when the network parameters (w_2,b_2) are altered. When one parameter (w,b) is altered, the others remain constant. After the pre-training, the parameters are adjusted by the back propagation algorithm, and this process is called fine-tuning.

w_{ji}^{l} is the weight coefficient between the jth neuron in the ith layer and the ith neuron in L + 1 layers in the neural network, and b_i^{l+1} is the offset item of the ith neurons in the L+1 layer. z_i^{l+1} is the weighted sum of the ith neurons in the in the L+1 layer, and h_i^{l+1} are the activation values.

$$z_i^{l+1} = \sum_{j=1}^{s_l} w_{ji}^{l} x + b_i^{l+1}$$

$$h_i^{l+1} = f\left(z_i^{l+1}\right)$$

(6)

In the above formula, s_l is the total number of neurons in the L layer.

Amend the adjustment parameter, and minimize the error of between the initial input and the output, and the loss function is as follows :

$$J_{AE}\left(w,b\right)=\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|h_{w,b}\left(x\right)^{(i)}-y^{(i)}\right\|^{2}\right)+\frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i}^{s_l}\sum_{j}^{s_{l+1}}\left(w_{ji}^{(l)}\right)^{2}$$

(7)

Where m is the number of input samples X . The first item is the average reconstruction error, and the second is the weight decay for reducing the over-fitting phenomenon.

3.2 Realization process of sparse de-noising auto-encoder neural network

The sparseness refers to only a few non-zero elements, or a few elements that are greater than zero. If a neuron is active, the output is close to 1. Otherwise, if a neuron is inhibited, the output is close to zero. If most neurons are limited, the state is seen as sparse restrictions. The auto encoder neural network with a sparse constraint condition is named the sparse auto encoder, and the cost function of the sparse auto- encoder neural network is as follows:

$$J_{SAE}\left(w,b\right)=\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|h_{w,b}\left(x\right)^{(i)}-y^{(i)}\right\|^{2}\right)+\frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i}^{s_l}\sum_{j}^{s_{l+1}}\left(w_{ji}^{(l)}\right)^{2}+\beta\sum_{j=1}^{s}KL\left(\rho\,\|\,\tilde{p}_j\right)$$

$$\sum_{j=1}^{s}KL\left(\rho\,\|\,\tilde{p}_j\right)=\left(\rho\log\frac{\rho}{\tilde{p}_j}+\left(1-\rho\right)\log\frac{1-\rho}{1-\tilde{\rho}_j}\right)$$

(8)

β is the sparse punitive factor, and λ is the weight decay. ρ is the sparse parameter, and s is the neurons number in hidden layer. $\tilde{\rho}_j$ is the average activation value of the hidden layer nodes, $\tilde{\rho}_j=\frac{1}{m}\sum_{i=1}^{m}a_j^l\left(x^{(i)}\right)$. $a_j^l\left(x\right)$ is the activation value, and m is the sample number.

In order to minimize the refactoring losses, get the optimal parameter, and use the stochastic gradient descent algorithm to update the weight matrix by the following formula.

$$w_{ij}^{l}=w_{ij}^{l}-\alpha\frac{\partial}{\partial w_{ij}^{(l)}}J_{AE}\left(W,b\right)$$

$$b_{i}^{l}=b_{i}^{l}-\alpha\frac{\partial}{\partial b_{i}^{(l)}}J_{AE}\left(W,b\right)$$

(9)

α is the learning rate, $\frac{\partial}{\partial w_{ij}^{(l)}} J_{AE}(W,b)$ and $\frac{\partial}{\partial b_i^{(l)}} J_{AE}(W,b)$ are the partial derivative of the weight and bias. The residual of error each neuron in the hidden layer is as follow.

$$\delta_i^l = -(y_i - a_i) f'(z_i) \qquad \textit{input layer}$$

$$\delta_i^l = \left[\left(\sum_{j=1}^{s_2} w_{ji}\delta_i^{l+1}\right) + \beta\left(-\frac{\rho}{\tilde{\rho}_j} + \frac{1-\rho}{1-\tilde{\rho}_j}\right)\right] f'(z_i) \qquad \textit{hidden layer}$$

(10)

The following formula is the partial derivative of the weight and bias.

$$\nabla_{W^l} J(W,b) = \frac{\partial}{\partial W_i^l} J_{SDAE}(w,b) = a_j^l \delta_i^{l+1}$$

$$\nabla_{b^l} J(W,b) = \frac{\partial}{\partial b_i^l} J_{SDAE}(w,b) = \delta_i^{l+1}$$

(11)

The training process of the sparse auto encoder neural network is as follows.

1. Initialization.

$\Delta W^{(l)} = 0, \Delta b^{(l)} = 0$. Set all weights and biases as zero.

2. Use the forward formula, and calculate for feedforward conduction. $Z^{(2)} = w^{(1)}D + b^{(1)}, a^{(2)} = f\left(z^{(2)}\right)$

for $l = 3 : L$

$$Z^l = w^{(l)}a^{(l)} + b^{(l)}, a^{(l)} = f\left(z^l\right) .$$

3. Calculate the average output of all hidden layer nodes.

for j=1: s_l

$$\tilde{p}_j = \frac{1}{m}\sum_{i=1}^{m} a_j\left(x^{(i)}\right)$$

4. Calculate the total cost function.

$$J_{SAE}(w,b) = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|h_{w,b}(x)^{(i)} - y^{(i)}\right\|^2\right) + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i}^{s_l}\sum_{j}^{s_{l+1}}\left(w_{ji}^{(l)}\right)^2 + \beta\sum_{j=1}^{s} KL\left(\rho \,\|\, \tilde{p}_j\right)$$

5. Calculate the residual error of all output layers.

$$\delta_i^l = -(y_i - a_i) f'(z_i) , \delta_i^l = -(y_i - a_i) f'(z_i) .$$

6. Calculate the residual error of the others.

for l=$n_l - 1 : -1 : 2$

{

$$\delta_i^l = \left[\left(\sum_{j=1}^{s_2} w_{ji} \delta_i^{l+1} \right) + \beta \left(-\frac{\rho}{\tilde{\rho}_j} + \frac{1-\rho}{1-\tilde{\rho}_j} \right) \right] f^{'}(z_i)$$

$$\nabla_{W^l} J(W,b) = \frac{\partial}{\partial W_i^l} J_{SAE}(w,b) = a_j^l \delta_i^{l+1}$$

$$\nabla_{b^l} J(W,b) = \frac{\partial}{\partial b_i^l} J_{SAE}(w,b) = \delta_i^{l+1}$$

$$\Delta W^{(l)} = \Delta W^{(l)} + \nabla_{w^l} J_{SAE}(W,b)$$

$$\Delta b^{(l)} = \Delta b^{(l)} + \nabla_{b^l} J_{SAE}(W,b) \; w^{(l)} = w^{(l)} - \alpha \left[\left(\frac{1}{m} \Delta w^{(l)} + \lambda w^{(l)} \right) \right], b^{(l)} = b^{(l)} - \alpha \left[\frac{1}{m} \Delta b^{(l)} \right]$$

}

The input data are joined by a certain probability distribution of noise to form the sparse de-noising auto-encoder neural network, which can learn to remove the noise and reconstruct the undisturbed input as much as possible, so that the characteristics obtained from the noise data are more robust. At the same time, the generalization ability of the neural network is promoted. Raw data X is disturbed into noise data $\tilde{X}$ by probability distribution, and the cost function of the sparse de-noising auto-encoder neural network is defined as follows:

$$J_{SDAE}(w,b) = \frac{1}{m} \sum_{i=1}^{m} \left(\frac{1}{2} \left\| h_{w,b}(\tilde{x})^{(i)} - y^{(i)} \right\|^2 \right) + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i}^{s_l} \sum_{j}^{s_{l+1}} \left(w_{ji}^{(l)} \right)^2 + \beta \sum_{j=1}^{s} KL\left(\rho \,\|\, \tilde{p}_j \right) \quad (12)$$

4. Model of sparse de-noising auto-encoder neural network classifier

There was considerable success in that the layer by layer greedy algorithm was widely used to train the deep neural networking, and only a hidden layer of network was trained; when the training ends, two layer hidden networks start in the layer by layer greedy unsupervised training, and on the process continues. In the process of training, when the k-1 trained layers are fixed, the kth layer is trained. The input of the kth layer is the output of the k-1. The unsupervised learning method is used for the deep network training. All trained weights initialize the final deep network, and the entire network is optimized to use the training error of the training set with label. There is a priori information that unlabeled input data provides input data, and the initial parameters are located in a good position in the parameter space, so it is easy to obtain a local optimal solution.

This paper establishes three layers an auto-encoder neural network, and the whole training include two processes, which are the layer by layer greedy pre-training and the back propagation with stochastic gradient descent parameters fine-tuning. Use the sparse auto encoder to learn the essential characteristics, and thus the first encoder can train the first order characteristic, which is the input of the second encoder, and so on. The bottom hidden layer can extract the low-level features, and high-level learning can describe better the essential, until the last layer classifier can predict the class. The structure of sparse de-noising auto-encoder neural network classifier includes the auto-encoder and the classifier, and its model is as follows: (figure 2)

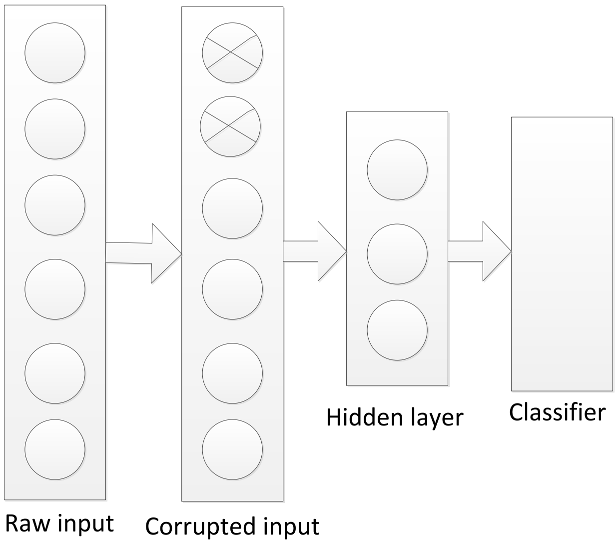

Fig. 2. Model of sparse de-noising auto encoder neural network classifier

The auto-encoder neural networks are whole, and all weights are optimized after iterations. After the optimization ends, the global optimal parameters are obtained.

5. Simulation experiment

5.1. Experimental data and parameters

At present, there are no uncertain data sets, and uncertain information is introduced as the common method. Choose the UCI database to demonstrate the effectiveness of the algorithm and the experimental data set is as follows:

Table 1. Experimental data set in UCI

Name	Attributes	Instances
Arcene	10000	900
Dexter	20000	2600
Gisette	5000	13500
Madelon	500	4400
Semeion	256	1593

At first, the normalization processing for the original data is carried out, and is effective in reducing redundancy and linked character. The certain data $x_i^c = \left(x_{i1}^c, ..., x_{id}^c\right)$ in UCI is added to the uncertain information, and is turned into interval-valued uncertain data.

$$x_{ij} = \left[x_{ij}^c - w_j, x_{ij}^c + w_j\right], j = 1, 2, ..., d$$

$$w_j = \alpha \times \frac{\max_i\left(x_{ij}^c\right) - \min_i\left(x_{ij}^c\right)}{10}, j = 1, 2, ..., d$$

$\alpha \in [0.5, 2]$ is the parameter of the noise amplitude.

Choose three layers of the auto-encoder neural network, and the number of input layers is 200. The neuron number of the first hidden layer is 200, and the second is 100. The third hidden layer neuron number is 20, and the output neuron number is 6. The noise parameter is 0.12, and the weighting attenuation factor is 0.002. The sparse parameter is 0.1, and the sparse penalty coefficient is 0.28. Choose Limited-Memory BFGS as the optimization algorithm, and the activation function is Sigmoid. The experimental platform is Windows 7 with Intel E5800 (3.2GHz), Matlab 2015a. The experiments use the 10-fold cross validation to calculate the classification accuracy, and choose a radial basis kernel function.

5.2. Result of experiments

The classification algorithm of the sparse de-noising auto-encoder neural network for uncertain data is abbreviated to SDAEUD. In order to verify the effectiveness of SDAEUD for classifying uncertain data, choose Arcene, Dexter, Gisette, Madelon and Semeion in UCI dataset as test data, and the noise amplitude parameter α is set to be 1. RBC [11], GBSD [12] are other algorithms for uncertain data. Average accuracies of the different are as follows after 100 repeated experiments in each data set.

Table 2. Average accuracies of the different algorithms

Algorithm	Arcene	Dexter	Gisette	Madelon	Semeion
RBC	89.65%	90.56%	89.43%	78.89%	90.41%
GBSD	90.48%	94.64%	90.68%	82.36%	96.53%
SDAEUD	94.43%	96.58%	96.32%	90.54%	98.46%

From table 2, SDAEUD has higher classification accuracy than RBC and GBSD for uncertain data, so SDAEUD is effective and feasible for the classification of uncertain data.

In order to verify the sparsity and de-noising process for classifying uncertain data, the sparsity in SDAEUD is removed, and obtains the de-noising auto-encoder neural network for uncertain data (DAEUD). The de-noising process in SDAEUD is removed, and obtains a sparse auto-encoder neural network for uncertain data (SAEUD). The sparsity and de-noising process in SDAEUD are all removed, and result in the auto-encoder neural network for uncertain data (AEUD). The noise amplitude parameter α is set to be 1, and the experimental results are as follows in table 3 after 100 times in Arcene, Dexter, Gisette, Madelon and Semeion.

Table 3. Average accuracies of the different algorithms

Algorithm	Arcene	Dexter	Gisette	Madelon	Semeion
AEUD	92.48%	94.71%	94.68%	89.12%	96.54%
DAEUD	94.12%	95.62%	95.42%	90.43%	97.58%
SAEUD	93.67%	95.25%	95.61%	90.15%	97.49%
SDAEUD	94.43%	96.58%	96.32%	90.54%	98.46%

As can be seen from table 3, SDAEUD for classifying uncertain data is better than AEUD, DAEUD, and SAEUD, so the sparsity and de-noising process are beneficial for classification. At the same time, the uncertainty has a great influence on classification data, and DAEUD has higher classification accuracy than SAEUD, so the de-noising process is effective for classification of uncertain data.

In order to verify the influence of the neurons of the hidden layer number, contrast experiments were carried out. The experimental results of the network configuration were as follows. The noise amplitude parameter was set to be the same, and the experimental results are as follows in table 4 after 100 times in Gisette and Semeion.

Table 4. Average accuracies of the different neuron number in hidden layer

hidden layer number	neuron number	accuracy(Gisette)	accuracy(Semeion)
1	200	94.23 %	96.15 %
1	400	94.65%	96.78%
2	200, 100	95.32%	97.71%

2	400, 200	95.79%	97.86%
3	200, 100, 20	96.32%	98.46%
3	400, 200, 40	96.62%	98.54%

From table 4, it can be seen that the classification accuracy increases when the neurons are greater in the same hidden layers. The greater the amount of neurons, the more information loss will decrease, so the learning ability is enhanced.

In order to verify the influence of the noise amplitude, the noise parameter is set from 0.5 to 2. The classification accuracies with the noise amplitude in Dexter and Gisette are as follows in figure 3 and figure 4:

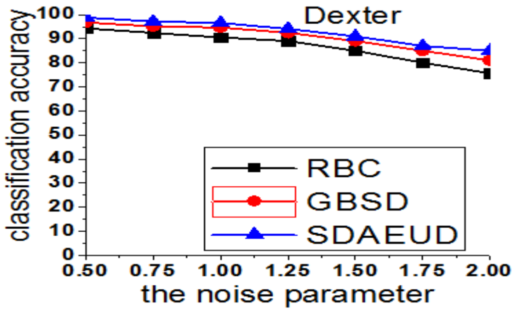

Fig.3. Classification accuracy with various noise parameters in Dexter

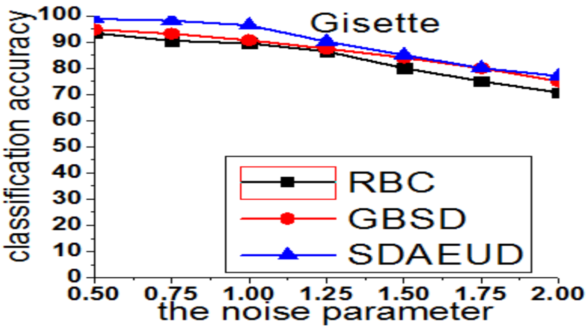

Fig.4. Classification accuracy with various noise parameters in Gisette

From figure 3 and Figure 4, it can be seen that the classification accuracy with an increasing noise parameter in Dexter and Gisette decreases gradually. SDAEUD is higher than RBC and GBSD, so SDAEUD has a better performance when it classifies uncertain data with different noise parameters.

6. Conclusion

In recent years, there has been more and more research on auto-encoder neural networks based on deep learning, and which can eliminate irrelevant and redundant information effectively and improve the efficiency of the inherent characteristics of the learning data. This paper proposes a classification method for uncertain data based on a sparse de-noising auto encoder neural network, which overcomes the limitations of the surface learning. We dispose of uncertain data by the hyper-ellipsoid convex model, and provide an approach for uncertain data classification based on an interval uncertainty support vector machine. This paper introduces a three layer auto-encoder neural network, with its deep structure tuned with the stochastic gradient descent parameters fine-tuned by layer greedy pre-training and the back propagation. The results of the experiments demonstrate that the proposed algorithm is effective for uncertain data classification, which can dispose of incomplete data with different noise parameters.

Acknowledgement

This work was supported by the science and technology projects of the Shandong province universities (No. J15LN80). This work has been partially supported by grants from the research fund of heze university (No. XY14KJ08, XY1509).

References

1. B. Tavakkol, M.K. Jeong, S.L. Albin. Object-to-group probabilistic distance measure for uncertain data classification, 2016, Neurocomputing, Vol. 230, pp. 143-151.
2. K.Y. Cao, G.R. Wang, D.H. Han, M. Bai, S.R. Li. An algorithm for classification over uncertain data based on extreme learning machine, 2016, Neurocomputing, vol. 174, no. 1, pp. 194-202.
3. H.C. Lin and C.T. Su, A selective Bayes classifier with meta-heuristics for incomplete data, 2013, Neurocomputing, Elsevier, Vol. 106, 95-102.
4. G. Lee, U. Yun, H. Ryang. An uncertainty-based approach: Frequent item set mining from uncertain data with different item importance, 2015, Knowledge-Based Systems, vol. 90, pp. 239-256.
5. Z. Liu, Y. Liu, J. Dezert, Q. Pan. Classification of incomplete data based on belief functions and K-nearest neighbors, 2015, Knowledge-Based Systems, vol. 89, pp. 113-125
6. G.C. Patrick, W.G.B. Jerzy, W. Rzasa , Mining incomplete data with singleton, subset and concept probabilistic approximations, 2014, Information Sciences, Springer International Publishing, vol. 280, pp. 68-384.

7. Y. Zhang, E. Zhang, W. Chen. Deep neural network for halftone image classification based on sparse auto-encoder. 2016, Engineering Applications of Artificial Intelligence, vol.50, pp. 245-255
8. W. Sun, S. Shao, R. Zhao, R. Yan, X. Zhang, X. Chen. A sparse auto-encoder-based deep neural network approach for induction motor faults classification, 2016, Measurement, vol. 89, pp. 171-178
9. D. Mellouli, T. Hamdani, A. Alimi. Deep neural network with RBF and sparse auto-encoders for numeral recognition. 2015, International Conference on Intelligent Systems Design & Applications, Marrakesh, Morocco, pp. 468-472
10. E. Marchi，F. Vesperini, F. Eyben, S Squartini. A novel approach for automatic acoustic novelty detection using a denoising auto encoder with bidirectional LSTM neural networks, 2015, IEEE International Conference on Acoustics, South Brisbane, Queensland, Australia, pp. 1996-2000.
11. W.J. Li, X.F. Xiong, Y.M. Mao. Classification method for uncertainty data based on hyper-sphere support vector machine, 2015, Computer engineering and design, vol.36, no. 7, pp, 1778-1783.
12. M. Ramoni, P. Sebastiani, Robust Bayes classifiers, 2001, Artificial Intelligence, Elsevier, vol. 125, no. 1-2, pp. 209-226.
13. J.N. Chen, H.K. Huang, X. Li, C.H. Yin, Constructing Hybrid Selective Classifiers for Incomplete Data with Gain-Ratio, 2009, Journal of Beijing Jiao tong University, vol. 33, no. 5, pp. 117-121.