17. V. Morkun, O. Savytskyi, S. Ruban (2015) The use of heat pumps technology in automated distributed system for utilization of low-temperature energy of mine water and ventilation air, *Metallurgical and Mining Industry*, 7(6), p.p.118-121.

Multidimensional Data Classification Based on BP Neural Network Algorithm

Rong Deng

Department of Information Engineering, Chongqing Vocational Institute of Engineering, Chongqing, 402260, China

Abstract
This paper selects the BP neural network algorithm for data classification, and carries out a detailed derivation of its work principle. For the existing slow network convergence and prone to falling into local minimum and other defects of this algorithm, the paper adopts the method of the combination of using the variable learning rate and added with the momentum factor to improve the traditional BP algorithm. In the specific network training experiment, it is found that the improved algorithm has improved the speed of network convergence to certain extent. And the final data classification results show that, BP neural network has a relatively high success rate when applied for the multi-dimensional data classification in the Internet of Things.
Key words: INTERNET OF THINGS, DATA CLASSIFICATION, BP NEURAL NETWORK, CLOUD COMPUTING, MULTI-DIMENSIONAL DATA

1. Introduction

In the process of data mining, data classification is a very important method of data analysis [1]. In the world that we live in, all human activities are derived from the accumulated wisdom and knowledge of mankind himself. Human makes the right judgments and decisions based on the combination of the knowledge which has been gained through experience and the perception to the external things. In fact, if we only study the data itself, it will not make any sense, because it's just the raw materials that human has obtained through a variety of available tools and instruments. While the internal logic of things hidden behind these data is what we want to explore, and these existing experiences will be the foundation to provide direct assistance for us to make decisions [2]. Therefore, we need to make analysis on the raw data so as to find the internal relations between them; hence the things that just make sense mathematically will be turned into useful information. However, they cannot be directly identified by human, and need to summarize and classify a lot of multi-dimensional data before it can be converted to the knowledge that human can make use of. Thus, when faced with huge amount of original data, classification and extraction of the data can provide us the knowledge and wisdom

engineering that can generally be understood by us, with very high research value. In fact, in everyday life, the human brain does the classification on all things that appear around us at all times, and makes judgment according to the classification result. In the field of data mining, the data classification is defined as the process to construct the classification rules according to certain classification method based on the existing data in line with the characteristics of the original data set [3]. The classification rules obtained by this method can map the data record accumulated in the raw data to one of the target classification results, so as to achieve the classification of the data set. In short, the process to perform data classification is a kind of potential classification rules constructed through the analysis of the existing data, and then use it to classify the data with similar properties but unknown data type.

Currently, many scholars carried out extensive research on data classification, the combination of the cost-sensitive learning method and the classical algorithm is of positive effect to improve the recognition rate of the subclass, as well as the weighted random forests algorithm proposed by C. Chen et al. [4], giving relatively greater weight to the sample of subclass, and relatively smaller weight to the division; Y Sim, et al. leveraged the idea of cost-sensitive learning method to improve the Adaboost algorithm [5], etc.; however, all these algorithms integrated with the cost-sensitive learning method have the same weakness: In many cases, the cost of misclassification of divisions and subclasses is difficult to be estimated correctly. The combination of the integrated learning method and the classical algorithm has higher accuracy in the reference classifier, and when the various sub-classifiers are different in classification, it can also improve the performance of classification of the classic classification algorithm on the unbalanced datasets. Intelligent optimization method has also been used in the data classification area: Combine the genetic algorithm with the sampling technology to handle the issue of data classification in the software quality evaluation model [6]; the particle swarm algorithm is used to optimize the neural network decision boundary to solve the influence of the classified dataset to the neural network classifier [7]; genetic algorithm, task decomposition techniques and other methods are utilized for hybrid modular neural network, which can also solve the issue of classification learning [8]; and the ant colony algorithm method combined with sampling method can solve the DNA microarray data classification issue [9-10].

In practical application, it is discovered that the traditional BP neural network has the problems of slow convergence and the susceptibility to fall into local minimum, etc. Obviously these problems will constrain the application of the BP neural network in the Internet of Things with massive data [11-12]. In this regard through the theory analysis on the neural network, this paper proposes the use of the method of the combination of the variable learning rate and added with momentum factor to improve the traditional BP learning algorithm. In order to verify the effectiveness of the improved algorithm, in the MATLAB environment, this paper makes full use of the neural network toolkit and GUI interface it provides, to realize a data classification experimental system. And use the system to make classification on the data collected by the Internet of Things in a multidimensional manner, and provide the comparison of the network convergence performance before and after the improved algorithm.

2. Fundamental principle of BP neural network

In various feedforward neural networks, the widely used training algorithm is the error back propagation algorithm, that is, the BP algorithm. Usually we refer to the feedforward neural network that adopts the BP algorithm for training BP neural network, and the BP neural networks mentioned herein refer to the feedforward neural network that adopts the BP algorithm. In respect of many tough problems of multilayer feedforward neural network learning that have been bothering us, the BP algorithm has very good performance. According to statistics, in the practical application in the artificial neural networks, 80% to 90% of the artificial neural network models apply the BP network or its variations, which is also the core part of the feedforward network, reflecting the most essential part of the artificial neural network [10]. Generally the learning process of BP algorithm can be divided into two phases, the training input data forward propagation phase and the error signal back propagation phase.

Under the three-layer feedforward neural network model, the energy function adopted by the BP algorithm is the mean square error calculation formula as shown in the following:

$$E(\bar{x}) = \frac{1}{2}\sum_{k=1}^{k}[y_k(\bar{x}) - t_k(\bar{x})]^2 \quad (1)$$

In the forward propagation phase of the training data, in the standard BP learning algorithm, the input information of neurons in each layer of the network is derived from its immediate predecessor neurons, and passes the calculated output only to the immediate successor neurons. After all layers of neurons process the input data, it is transferred from the input layer

to the output layer from the front to the back layer by layer, and finally output the final calculation result from the output layer. If the network output error cannot meet the condition, then enter the error signal back-propagation phase, to adjust the threshold value of each neuron node that regulates the neural networks and each weight. Among them, the partial derivative of the energy function for each connection weight in the neurotic network and the neuron threshold can be calculated by the adoption of the following equation.

$$\frac{\partial E(\bar{x})}{\partial \theta_j} = \sum_{k=1}^{k} \frac{\partial E(\bar{x})}{\partial y_k(\bar{x})} \frac{\partial y_k(\bar{x})}{\partial net_k(\bar{x})} \cdots \frac{\partial y_j(\bar{x})}{\partial net_j(\bar{x})} \frac{\partial net_j(\bar{x})}{\partial \theta_j} \quad (2)$$

Although the BP network has a solid theoretical foundation, as well as rigorous derivation process, it still has some flaws, which are summarized as follows:

1) Slow Convergence

Standard BP algorithm is actually a gradient descent method, and the objective function it requires to optimize is very complicated, hence there will be relatively flat area in the error curve. When the weight value adjustment enters these areas, the change in the error gradient will be so small that the convergence rate will be stagnated, making the whole training process almost a standstill.

2) The Presence of Over-fitting Problem

Typically, if the training effect is not ideal, its generalization capacity will certainly not be ideal. As far as BP network is concerned, each adjustment on the weight is the learning for knowledge, if the ability to learn is not strong enough, it is certain that the final application will not be ideal. However, this trend has a limit, that is, when the training exceeds this limit, the generalization capability of the network will no longer be strengthened with the increase of the number of training. Academically speaking, this phenomenon is called "over-fitting." At this time the neural network cannot be generalized for the internal law contained in the sample due to the learning of too many sample details.

3) Difficult to Determine the Network Structure

All along, for the BP network topology selection, in particular the selection of the number of hidden layers and number of neurons of the hidden layer is lack of theoretical guidance, which is normally selected according to experience. However, different topology structure has huge influence on the performance of different BP neural networks.

4) Easy to fall into Local Minimum Value

During the training and learning process of BP neural network, the weight of each layer always gradually changes from certain starting point to reach the minimum value of errors. However, for a very complicated network, taking into account the weight, threshold and the number of training and other relationship at the same time, we can draw the error function as a multi-dimensional curved surface, and there may be a lot of local minimum points at the bottom of the curved surface. Typically the standard BP algorithm always approaches the local minimum point from one direction, without the ability to jump out, therefore, even training continues, the error value will no longer have huge change.

3. Improved BP neural network classification algorithm

In response to these shortcomings of traditional BP algorithm, this paper presents an improved BP algorithm. The improvement method is proposed as below. First of all, adopt the variable learning rate method to solve the training problem of BP network, and then try to include the momentum factor in the algorithm so that during the entire training process, the network will not fall into the local minimum.

3.1. Variable Learning Rate

In practical application, the value taking for the learning rate has a great impact on the convergence and effectiveness of the BP algorithm. Its optimal value is often associated with specific issues; therefore, there is no definite learning rate which is suitable for any issue. In fact, even in a particular issue, awe cannot find an appropriate learning rate that is suitable from the beginning to the end. As BP algorithm is very sensitive to the changes in the performance of learning rate, if the learning rate is set too small, the convergence speed will be very slow; if the learning rate is set too large, although the convergence speed can be accelerated, it may lead to the fluctuation in the adjustment of the weights. Therefore, we should modify the network learning rate according to the specific training situation. The modification on the network learning rate is as follows:

First, from the initial output of the BP network and the calculation of the error, equation (3) is obtained:

$$w_{ji}(n+1) = w_{ji}(n) + \eta(n) D(n) \quad (3)$$

Where $D(n) = -\frac{\partial E}{\partial w_{ji}} = \delta_j x_i$

namely the negative gradient at the time n

$$\Delta \omega_{ij} \quad (4)$$

$$\Delta \omega_{ij}(t) \quad (5)$$

When obtaining a new weight value by the current rate and the error function, do not rush to throw away the old error. Wait until the modified new error is

generated, then compare the new error with the 1 day error, if the ratio is greater than the set value, it shows that the error change is too large, then we need to reduce the learning rate, for the specific adjustment we can obtain by equation (4) and (5) through calculation. When the old error ratio is too small, we will increase the learning rate by this equation. Thus, the application of this method can maintain a stable value at the learning rate area, so as to meet the normal convergence rate in the training process.

When using this algorithm, if two consecutive iterative gradients have the same direction, it indicates that energy decline is too slow, then we can consider double increasing the learning rate, and improve the training convergence speed; when the energy drops too quickly, there will be two consecutive iteration gradients in the opposite direction, then, we should halve the learning rate μ, so as to prevent fluctuation.

3.2. Adding Momentum to Overcome the Local Minimum Value

In fact, a good gradient descent should be taken at infinitely small intervals. Learning rate μ directly indicates the learning rate of the network, which can be seen from the equation as a constant scaling factor. The change of μ will directly affect the rate of change in the weight of the network, when I select it, I need to take into account that in the case as large as possible, to ensure that there will be no fluctuation occurred in the network. For this purpose, this paper propose to add a momentum term into the weight adjustment formula, in this mode the network needs to record the amount of adjustment to the weight each time, and the weight adjustment formula is:

$$\Delta\omega(t+1) = \eta\frac{\partial E}{\partial\omega} + \alpha\Delta\omega(t) \tag{6}$$

Where α is the momentum factor, usually it takes about 0.9.

After the introduction of the momentum item, the error function's curved surface change will become very small, so that $\Delta\omega(t+l)$ is approximately equal to $\Delta\omega$, then the average $\Delta\omega$ can be approximated represented as equation (7):

$$\Delta\omega \approx \frac{-\eta}{1-\alpha}(\frac{\partial E}{\partial\omega}) \tag{7}$$

Where $\frac{-\eta}{1-\alpha}$ is the scale factor, after adding the momentum item, the scale factor of this ratio can free the weight modification process from local saturation region.

If ω_{ij} is only the value of a variable, equation (7) can be revised as follows:

$$\Delta\omega \approx \frac{-\eta}{1-\alpha}(\frac{\partial E}{\partial\omega})\Delta\omega_{ij}(t+1) = \Delta\omega_{ij}(t+1) + \alpha\Delta\omega_{ij}(t) \tag{8}$$

As $\Delta\omega_{ij}(t+1) = -\eta\frac{\partial E}{\omega_{ij}}$, its value represents the current value modification direction, if two consecutive $\Delta\omega_{ij}$ have the consistent signs, which shows that on axis $\Delta\omega_{ij}$, the error has a decreasing trend. At this point we should increase the modification amplitude for the weight next time, that is, increase the value of $\Delta\omega_{ij}(t+2)$. If they have inconsistent signs, it is clear that there is a local minimum error in the two corrections, then we should reduce the modification amplitude to decrease the weight next time, that is, reduce the value of $\omega_{ij}(t+2)$, after such continued training, the error will slowly converge at the next minimum. Typically at this time we will add up $\Delta\omega_{ij}(t+1)$ and $\Delta\omega_{ij}(t)$ as the value for $\omega_{ij}(t+2)$, so as to prevent the network error from fluctuation. Through the above theoretical analysis, we found that the introduction of momentum item can inhibit the neural networks from falling into certain local minimum so as to improve the convergence performance of the network.

By exploring the aforementioned two improvement plans, this paper proposes the combination of the variable learning rate and adding the momentum item, so that the BP network will not fall into the local minimum, and the training convergence speed can also be improved, without the occurrence of error fluctuation phenomenon in the same area.

4. Experimental analysis

In this paper, the experimental data samples adopted are collected by KJ and KJF series of coal mine security monitoring system, which is the security monitoring system commonly used in most of the coal mines in our country. The monitoring system is a small scale sensor network under the Internet of things, and different sensor nodes are installed in different positions under the coal mine, to perform real-time monitoring of a variety of data, which is stored in the database and saved once every three minute for record. The data model applied in the test this time includes time, wind speed, temperature, carbon monoxide, methane concentration, negative pressure and other 10 property values that the monitor collected, as the security level in the coal mine in certain period has an internal relationship with the current wind speed, temperature and other properties. It is still very necessary to perform classification on the multi-dimensional data collected at the current moment, as the security monitoring needs to filter out the data with low level of security for focused analysis. The name of the data set is KJ2010D110K, saved in the KJ2010D110K.txt under the path of test folder, for details please refer to Figure 1. Every action of the file has a sample (sample), with a total of 10 property

values and a type value. As can be seen that, every 10-dimensional vector is corresponding to a data type, and there is a total of 6 all data types, that is to say, the finally realized classification should be a 10-dimensional property to one-dimensional type of mapping.

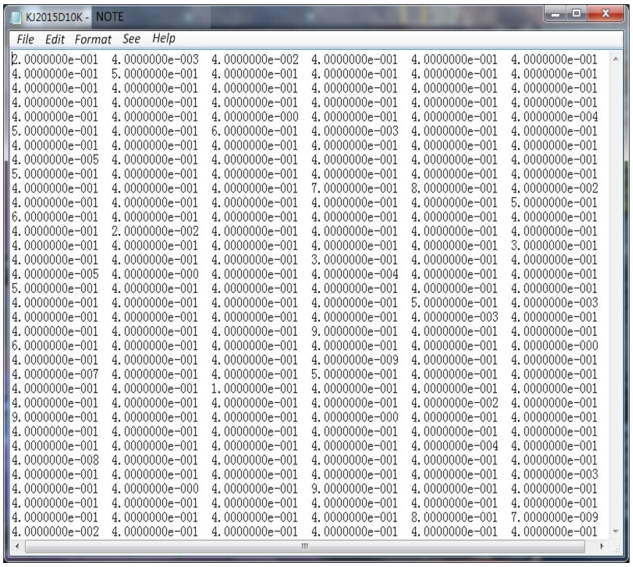

Figure 1. The content of data set sample

Through the analysis of the sample data in Figu- re 1, it can be found that the original data values used in this experiment was too broad, in order to facilitate the realization of BP network training in MATLAB, it is necessary to perform normalization process on the data before using. Data normalization is actually a way to simplify the calculation, theoretically speaking it is certain specific conversion of the dimensionless expression results, thus to avoid different physical meaning and the inequality generated during use of the dimensional input variable. For example, wind speed is generally at lOm/s or so, while the carbon dioxide concentration measured is about 0.2%, so if only comparing and process such two properties in the numerical sense, it is clearly inappropriate. In fact, the specific role of normalization in statistics is to summarize and unify the statistical distribution of the sample. Normalization between [0,1] is the statistical probability distribution, normalization between [-1,1] is the statistical distribution of the coordinates. Since the unit of data collected is not the same, in this experiment the normalization process [-1,1] to data is necessary, and there are many methods for normalization. To simplify the operation, this paper adopted the linear function conversion, with the expression as follows:

$$y=(x-MinValue)/(MaxValue-MinValue) \quad (9)$$

Where x is the data value before the conversion, y is the data value after the converted, while MaxValue and MinValue are the maximum and minimum of the property respectively taken from the sample. In this experimental system, when the data upload is completed, click on the data normalization, the system will follow the linear function a conversion method to normalize the experimental raw data. And print in the MATLAB Window Command window the data results after the normalization, as shown in Figure 2:

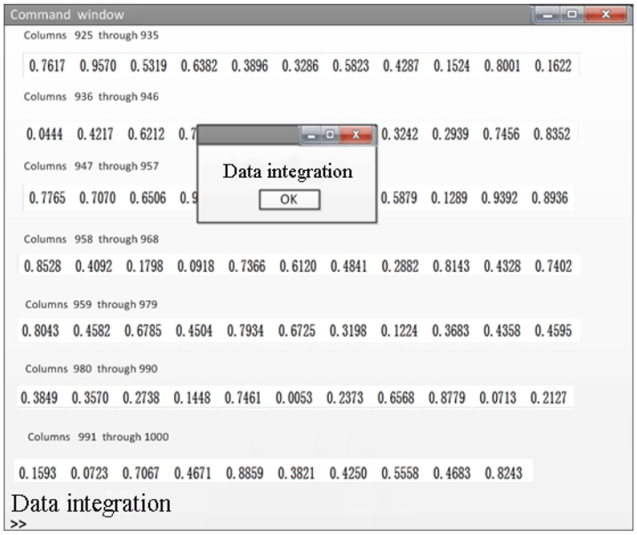

Figure 2. Data normalization

According to the comparison of different classification error of the BP network consisted by different number of hidden layer nodes and the convergence speed, it can be found that when the number of the hidden layer nodes is 24, the network MSE basically reaches the minimum, then the number of training necessary to achieve the target network accuracy is 11,710 times which can still be accepted by the experiment. And by comparison, it can be found that, when the hidden layer nodes continues to grow after more than 24, it will not bring significant improvement to network performance, instead, it will result in significantly slower convergence. Therefore, it is finally confirmed that the BP network topology in the classification experiment for the data set in the Internet of Things is: 10 input layer nodes, 24 hidden layer nodes, and 6 output layer nodes. The final classification experiment is carried out by the adoption of the selected network structure, and the results are shown in Figure 3:

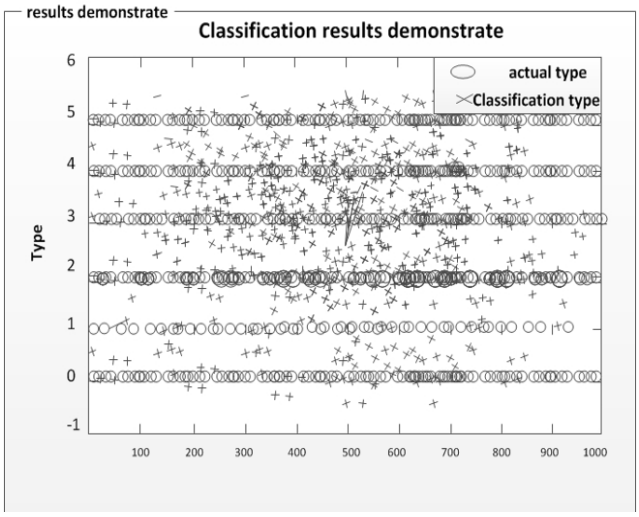

Figure 3. The results of data classification

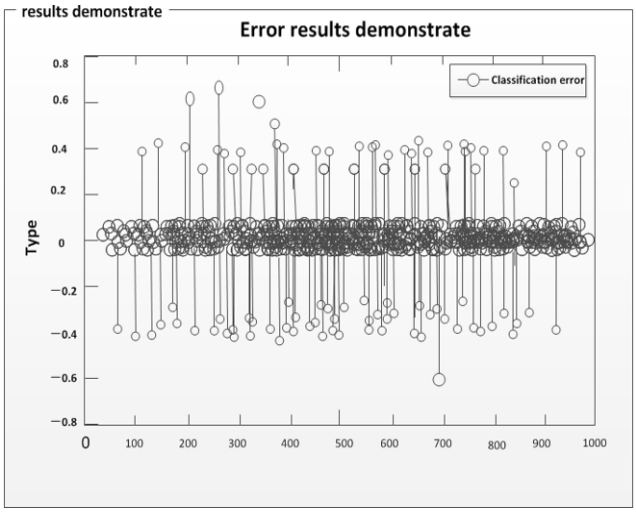

Figure 4. The graph of classfication result error

Figure 4 is the classification result error, which is a direct error, that is, the classification result root directly subtracts the error obtained from the actual results. It can be seen that in experiment determined optimal BP neural network, our results are still very good, when the mean square error MSE is at 0.015 or so. By using the errsurf function provided by the neural network toolbox of MATLAB, it is easy to obtain the curved surface for error of certain neuronal node in the training process, as shown in Figure 5. Obviously this curved surface has only one extreme point, therefore, in the process of the adoption of the improved BP algorithm to train the network to achieve the convergence; the correction to the neuron weight does not fall into the local minimum. Finally, this paper sets the classification rule that the classification error is less than 0.5 for the correct classification result, and the classification result after the statistics is shown in Table 1, and the result shows that the final classification accuracy rate is as high as 98%.

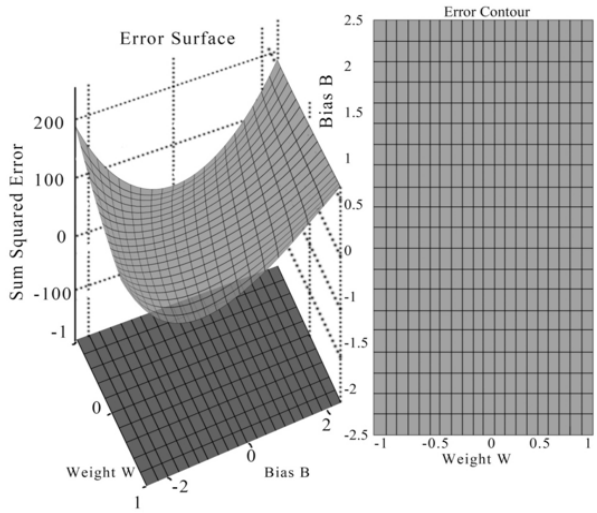

Figure 5. The error surface of data classification

Table 1. Comparison Table between Network Learning Rate and the Number of Training

Experimental Data Set	Classification Type	Training Set	Testing Set	Number of Correct Classification	Classification Accuracy Rate
KJ2010D110K	First classification results	1000	1000	981	98.1%
	Result after the cross validation	1000	1000	989	98.9%

In the overall process of the classification experiment, it is recorded that the time consumed for 11,710 times of stand-alone execution of BP network training is about 257s, while the time consumed for the classification of 1000 groups of testing sets is less than 1s. It can be seen that the improvement to the BP neural

network algorithm can enhance the speed of network convergence to a greater extent.

5. Conclusion

In order to verify the effectiveness of the improved BP algorithm proposed in this paper, this paper realized a data classification system in the Internet of Things under the GUI environment in MATLAB, and through the system carried out a huge number of classification experiments. The results showed that in certain training interval, the improved BP algorithm significantly accelerated the convergence speed of the network. In addition, this paper also performed testing on the BP network performance for different topology structures under this system, through the integrated analysis of the network classification error and convergence speed, it ultimately determined the BP neural network topology structure which was suitable for the data type, and finally, carried out data classification experiment by the adoption of this topology structure on the forecast data sets.

The classification results show that, when BP neural network is used for data classification on the data set in the Internet of Things, the success rate of classification is as high as more than 98%, which can meet the general requirements for data classification. Through the analysis of individual neuron error curved surface, it is found that the network does not fall into the local minimum. Therefore, it can be seen that the improved BP algorithm proposed in this paper has enhanced the performance of BP neural network to a certain extent.

Acknowledgments

This paper is a scientific and technological research project of Chongqing Municipal Education Commission, Project Name: *AIS-based Controlled Section Vessel Traffic Monitoring and Command System*. Project Number: 152072; which is also a scientific research project of Chongqing Vocational Institute of Engineering, Project Name: *Research on the Teaching and Training System of Mobile Communication Base Station Survey and Design*. Project Number: JG142024.

References

1. Ou G., Murphey Y. (2007) Multi-class pattern classification using neural networks. Pattern Recognition, 47(1), p.p. 4-18.
2. Rong H.J., Huang G.B. (2009) Online sequential fizzy extreme learning machine for function approximation and classification problems. IEEE Transactions on Systems, Man and Cybernetics-Part B, 42(4), p.p. 1067-1072.
3. Thai-Nghe N., Gantner Z., Schmidt-Thieme L. (2011) A new evaluation measure for learning from imbalanced data. Proc. of the 2011 International Joint Conference on Neural Networks, p.p. 537-542.
4. Hand D.J. (2009) Measuring classifier performance: A coherent alternative to the area under the ROC curve. Machine Learning, 80(1), p.p. 103-123.
5. Fawcett T. (2005) An introduction to ROC analysis. Pattern Recognition Letters, 33(8), p.p.861-874.
6. Batista G.E., Prati R.C., Monard M.C. (2004) A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter, 12(1), p.p.20-29.
7. Murphey Y.L., Guo H., Feldkamp L. (2004) Neural learning from unbalanced data. Applied Intelligence, 31(2), p.p. 117-128.
8. Kubat M., Holte R., Matwin S. (2005) Learning when negative examples abound. Proc. of the 16th European Conference on Machine Learning, p.p.146-153.
9. Spackman K.A. (1989) Signal detection theory: valuable tools for evaluating inductive learning. Proceedings of the Sixth International Workshop on Machine Learning, p.p.160-163.
10. Gao D.Q., Yang G.X. (2003) Influences of variable scales and activation functions on the performances of multilayer feedforward neural networks. Pattern Recognition, 43(4), p.p. 869-878.
11. Polato M., Sperduti A., Burattin A., et al (2014) Data-aware remaining time prediction of business process instances. Proc. of 2014 International Joint Conference on Neural Networks, p.p. 816-823.
12. Duch W., Adamczak R., Grabczewski K. (1999) Optimization of logical rules derived by neural procedures. Proc. of International Joint Conference on Neural Networks, p.p.669-674.