

Negative Correlation Incremental Integration Classification Method for Underwater Target Recognition

Ming He^{a,b}, Nianbin Wang^a, Hongbin Wang^{a,*}, Ci Chu^a, and Songyan Zhong^c

^aCollege of Computer Science and Technology, Harbin Engineering University, Harbin, 150001, China

^bCollege of Computer and Information Engineering, Heilongjiang University of Science and Technology, Harbin, 150022, China

^cBeijing Institute of Computer Technology and Applications, Beijing, 100854, China

Abstract

In this paper, an incremental learning algorithm based on negative correlation learning (NCL) is used as an identification classifier for underwater targets. Based on Selective negative incremental learning SNCL (Selective NCL) algorithm in the process of training, there are numbers of hidden layer nodes that are difficult to determine training time. Problems such as over fitting analysis arise. The algorithm combined with Bagging makes the difference between individual network further increase, and ensures the generalization performance of the whole. On the basis of this method, the use of the selective integration method based on clustering and a new proposed algorithm called SANCLBag, combined with the convolution of underwater target recognition neural network shows that the proposed integration approach can make the difference between individual network in the classification process further increase, and ensure the whole generalization performance. The model has higher identification accuracy, and can effectively solve the problem of incremental learning.

Keywords: underwater target recognition; convolution neural network; incremental learning; negative correlation learning

(Submitted on January 29, 2018; Revised on March 12, 2018; Accepted on April 24, 2018)

© 2018 Totem Publisher, Inc. All rights reserved.

1. Introduction

Underwater passive target recognition plays an important role in national economy and national defense. With the development of information technology, national defense and submarine resources, accurate classification and detection of sonar target is of great importance for activities such as survivability and combating ability of naval vessels, exploitation of submarine resources, detection of fish stocks and exploration of oil and so on. Therefore, it is necessary to enable the sonar to detect and recognize the target, on which scholars at home and abroad have studied from many aspects. Many resolution methods have been proposed. At present, the underwater target recognition mainly analyzes recognition target based on underwater acoustic signal detection and processing. However, the complexity of underwater environment makes the effective information extracted from Low SNR underwater acoustic signal fail to meet requirement of practical operation. Therefore, this paper proposed a method based on incremental learning to study the underwater target recognition. This paper, based on the negative correlation learning (NCL) algorithm, introduces steepness factors to effectively reduce the number of iterations in model training. Also, by combining Bagging algorithm with NCL algorithm and making use of selection method based on clustering and ordering, it proposes a new kind of algorithm called SANCLBag; in addition, the experimental results show that the generalization capability of the algorithm is better than that of selective negative correlation learning algorithm, and its time complexity is also superior to the latter.

The remainder of the paper is structured as follows. In section 2, methods and results of previous studies on incremental learning are summarized. In section 3, a detailed introduction is presented to explain the framework of the proposed method, and the methods of feature selection and classification. In section 4, the experimental setup and results, and the discussion are presented, and the conclusion is given in section 5.

* Corresponding author.

E-mail address: wanghongbin@hrbeu.edu.cn

2. Related Work

Ensemble learning is a method to study and make decisions on the same problem by constructing several models [5,14,18,21], which is the same as different people vote on the same problem. In the existing algorithms related to integration learning, Bagging [2,15] and Boosting [1,7,12,27] are the most classical ones. Bagging is a method of resampling to construct multiple different data sets to generate multiple different base learners. In the Bagging algorithm, training can be parallel. Boosting is to set weights for each sample. After each round of training, the weight of the round error-prone samples is increased and the next round of training is performed. With this method, several weak classes can be promoted to strong classifiers.

Incremental learning is an algorithm in which new knowledge is learned from new data; meanwhile, it is able to maintain the learned knowledge as much as possible when learn new knowledge. The earliest approach on incremental learning study is the Adaptive Resonance Theory modules map (ARTMAP) [28,25], which was proposed in 1991 by Carpenter etc. Neural network is able to solve problems of incremental learning in certain degrees, and many algorithms related to incremental learning are related to the artificial neural network (ANN). In 2001, Kasabov proposed a method of Evolving Fuzzy Neural Networks (EFuNN) [11,10], which is similar with the above two methods. And most of the earlier incremental learning algorithms take the method of single learner. All these methods learn new knowledge and reduce old knowledge loss by changing internal structure of models; however, these modified methods are easily affected by the initial algorithm, leading to limitations in incremental learning. After that, some researchers proposed methods to solve incremental learning with integration learning, and one of the representative algorithms is the Learn++ algorithm. It is proposed by Polikar [13,22,23], etc. in 2001 and the core of it adopts AdaBoost algorithm. Negative Correlation Incremental Learning [17,19,26] is a method proposed by Minku in 2009, in which Negative Correlation Learning is the algorithm for ANN integration training. In his study, the NCL is used directly to implement incremental learning, and in their papers, two methods that can solve incremental problems are proposed; one is the Fixed Size NCL (FSNCL), and the other is the Growing NCL (GNCL). In the former, the number of learner is fixed, and when new data training sets are added, the neural network integration trained before would be used for training. Different from the former, in GNCL, the number of learners in integration would increase as the new data training sets are added. When there are new training sets, a new ANN is constructed and the NCL algorithm is used to train in new training sets, which would be added into the previous integration after training.

3. Selective negative incremental learning based on clustering

This paper has referred to the selective negative learning framework of Minlong Lin and modified the model training and model selection of it. The selective negative incremental learning framework is shown in Figure 1.

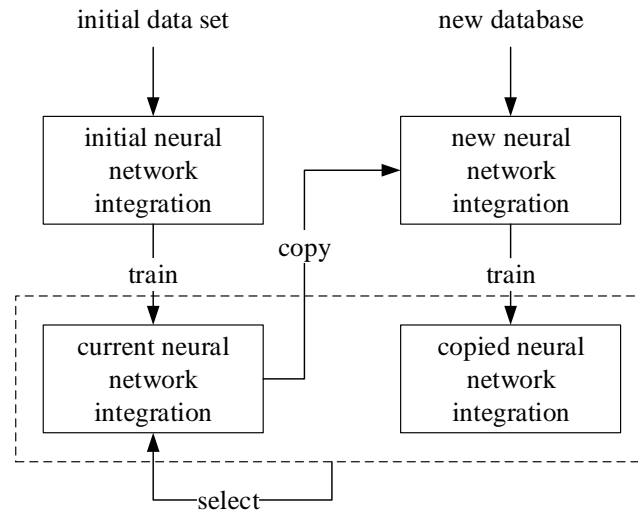


Figure 1. Framework of selective negative incremental learning algorithm

3.1. Data Description Improved negative correlation learning algorithm ANCL

NCL is based on the BP algorithm, and its essence is to modify the error function. However, it has not been optimized properly in areas like network training [6,16,26]. In order to solve the problem, this paper combines advantages and

disadvantages of existed algorithm based on pruning and increment, and proposes a new adaptive negative correlation learning algorithm, abbreviated as ANCL.

In the existed structure-adjustable algorithms based on growth, there is only one node in the hidden layer at the initial time generally; however, in the algorithms based on pruning, the number of hidden nodes is set to a maximum value in a reasonable range as possible at first [4,24]. In the methods based on both growing and pruning, the random number is taken as the initial number of hidden nodes. The obvious disadvantage of the former two methods is that the initial condition is set up extremely. In the training node, numbers may need to be added or reduced a lot, which may increase the training time. Also, it may make the network structure fall into local optimized. The third method is superior than the former two; however, there may be uncertainty for the random number, which may also lead to extreme conditions. The common points of the three methods are that they all abandoned the method to determine the number of hidden nodes based on experiment values proposed before. Though there may be defects in these methods, a large number of studies have showed its high validity in certain range. The ANCL algorithm proposed in this paper takes the experiment formula as a reference value when set up the initial value, which may effectively reduce the training time, and make the network structure achieve local optimized.

Compared with the existed algorithms, ANCL differentiates from others since it does not add or reduce hidden nodes mechanically according to fixed strategy. Plus, the time to add or reduce nodes totally depends on the increasing condition of hidden nodes learning ability in the training. In addition, the node numbers do not to be changed each time. Sometimes, the hidden nodes number may be the optimized, and in this condition what we need to do is just to adjust weights in each node connecting the network reversely. The overall flow of ANCL algorithm is seen in Figure 2, and the details are as following.

1. Construct M BP network for integration, and for each initial network, the node number in input and output layers in the network is consistent with the feature numbers and type numbers of given samples. The number of nodes in hidden layers is set with the Equation (1):

$$m = \sqrt{n+l} + \alpha \quad (1)$$

Where m is the number of nodes in hidden layers, l is the number of nodes in input layer, n is that in output layer. And α is a constant ranging from 1 to 10. In order to make structures of each network different, the α here is set as a random number.

2. For each network in the integration, a training time counter μ_i is set in each hidden node h_j , where $i = 1, 2, \dots, m$, and the initial value is 0. The counter is used to calculate the current training time in certain hidden nodes. When there are new nodes, the counters also need to be set.

3. Train each network for τ times, and the τ here is stipulated artificially. Steepness factors are added in training and when the adjustment of weights enters into flat regions (ΔE close to 0), increase the factor, while when it exits the flat regions (ΔE is greater than the minimum value), restore the factor. After training, judge whether the output error is less than the preset object error. If so, end the training; otherwise, implement the operation in each network in the integration.

1) For each node h_j in hidden layer, the Equation (2) is used to update μ_i .

$$\mu_i = \mu_i + \tau, \quad i = 1, 2, \dots, N \quad (2)$$

Where N is the number of nodes in hidden layers, and its initial value equals to m .

2) Delete all the nodes in tag set S and calculate the importance η_i of each node h_j in hidden layer. The η_i is calculated as Equation (3).

$$\eta_i = \frac{\sigma_i}{\sqrt[3]{\mu_i}} \quad (3)$$

Where σ_i is the standard deviation of hidden nodes i , and its value is calculated by the output data of all samples of the hidden nodes. A smaller standard deviation indicates that the hidden node has almost the same value for all the samples; in other words, the node has a weak ability to identify different samples. In the three layer forward feedback network, the hidden layer and output layer connect directly, and the dealing results of hidden layer for different data will have a direct impact on the output of the network [8]. The other parameter in the formula is the training time of the node, and the reason to adopt training time here is that the training time of new nodes are different from the previous ones for new nodes are generated in the adjustment. The small standard deviation in certain conditions is not simply caused by its weaker ability to identify different samples; it may be also caused by the shorter training time. Therefore, synthesizing these two aspects, the importance of the node is determined by its standard deviation and its training time. A larger standard deviation and shorter training time indicate the importance of the node, vice versa.

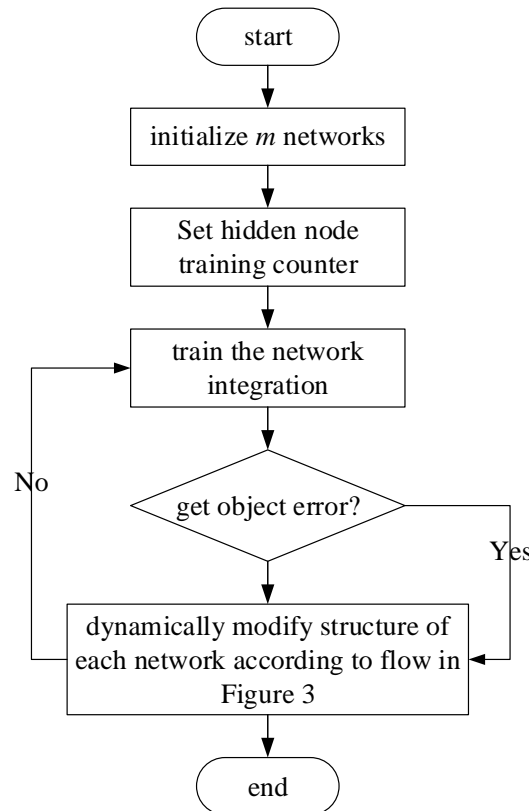


Figure 2. Flow chart of ANCL algorithm

- 3) If the importance of one or more nodes is less than the preset threshold θ , add it into tag set S . If the number of nodes added into set is greater than 0, continue the process; otherwise jump to step 7. Pay attention to selecting proper value when set θ ; in the algorithm, the number of nodes added into tag sets S should be more than half of all the nodes.
- 4) Calculate the correlation between the nodes in the set and the nodes haven't been added in the set. For convenience, the former are called unimportant nodes, while the later important nodes. Correlation is a common and useful statistical value to describe the relevant degree between two variables [3,9]. In statistics, there are many methods to measure correlation. Here, the famous Pearson product-moment correlation coefficient is used to calculate the correlation between different hidden nodes in the network [20]. The correlation coefficient calculation formula between important nodes h_i and unimportant nodes h_j is shown as Equation (4).

$$C_{ij} = \frac{\sum_{p=1}^p (h_i(p) - \bar{h}_i)(h_j(p) - \bar{h}_j)}{\delta_i \delta_j} \quad (4)$$

Where $h_i(p)$ and $h_j(p)$ are output of the i th and j th hidden nodes in sample p of training sets respectively. The variable \bar{h}_i and \bar{h}_j are the average value of h_i and h_j . Standard deviation δ_i and δ_j are that of hidden nodes h_i and h_j in all the training samples, and the calculation methods of which are same as that in Equation (3).

- 5) Merge the nodes in set S together with those of high correlation that haven't been added into S ; merge the unimportant nodes with the important nodes of high correlation. Calculate the correlation coefficient between any two important nodes and the unimportant nodes. The nodes pairs are sorted in a reverse order in accordance with the correlation coefficient, by which the node pair of highest correlation is obtained. The way to merge two nodes is show as Equation (5) and (6):

$$w_{mi} = \frac{w_{ai} + w_{bi}}{2}, \quad i = 1, 2, \dots, p \quad (5)$$

$$w_{jm} = w_{ja} + w_{jb}, \quad j = 1, 2, \dots, q \quad (6)$$

Where p and q are node numbers in input layer and output layer in the network respectively. Weights are the link weight of the i th input node and hidden node, correspondingly, and are link weights of and with the j th output node. The reason to merge nodes this way is to ensure the impact of the merged node is same as the impact of the two nodes, which also maintains the current training results of the network.

- 6) Set the training step counter μ_{new} of new added nodes, the μ_{new} is calculated as Equation (7).

$$\mu_{\text{new}} = \frac{\mu_a + \mu_b}{2} \quad (7)$$

Where μ_a and μ_b are counters of the merged node a and b respectively.

- 7) Judge whether the condition for adding node is met. If so, continue to the next step; otherwise, go back to step 2. The condition to judge whether nodes should be added is: after τ times of training, if the error deduction of network doesn't reach ε , which is a preset value. The equation of the judging condition is shown as Equation (8).

$$E(t) - E(t + \tau) \leq \varepsilon, \quad t = \tau, 2\tau, 3\tau, \dots \quad (8)$$

Where $E(t)$ and $E(t + \tau)$ are the training error of t iterative training and $t + \tau$ iterative training respectively.

- 8) Add hidden nodes into the current network, and go back to step 2. The node splitting method is used to split the current node to add nodes, equaling to add two nodes and delete one. Due to the stronger ability of the information processing of the new generated network, it is able to learn the object function faster than the previous one. Equation (9) and (10) used to split node is:

$$w^1 = (1 + \alpha)w \quad (9)$$

$$w^2 = -\alpha w \quad (10)$$

Where w is the weight vector of split node, w^1 is the weight vector of the first new node generated by the splitting, while w^2 is the weight vector of the second new node. The parameter α is distributed with fixed value, random value or certain probability. In this paper, α is set up with standardized normal distribution. The two new nodes generated by the splitting

and the deleted original node adds the number of hidden nodes, and improves the information processing ability of the network.

- 9) Set the training step counter for the split nodes; the new counters need to refer to the split node. The calculation equation is shown as Equation (11) and (12).

$$\mu^1 = (1-\alpha)\mu \quad (11)$$

$$\mu^2 = \alpha\mu \quad (12)$$

Where the value of α is consistent with the Equation (9) and (10). μ is the counter of the split node, while μ^1 and μ^2 are the counters of the new node. The decimal number obtained in the calculation should be rounded up to an integer.

Increasing differences between individual classifiers is an important research direction in integration learning. It is able to increase the difference between classifiers if the two algorithms are combined together, which can promote the generalization capability of the whole integration. Based on the above assumption, there have been scholars that combined the two algorithms. There are two steps to apply the Bagging algorithm in the negative correlation learning. The first step is to generate the training set of specified number with repeated sampling method in Bagging algorithm; the second step is to, in the training process, use the negative correlation learning algorithm and the proposed ANCL algorithm as well as the combined algorithm, which is abbreviated as ANCLBag algorithm.

The main step of the algorithm is:

1. Divide the original sample set into initial training set and validation set. With the method of repeated sampling, M different sample sets are generated in the initial training set for the following training. The size of each sample set equals to that of the initial training set.
2. Adopt ANCL algorithm to train M network for integrating.
3. Calculate the error rate of the validation sets in each network and network integration. If the error rate is less than the preset value, end the training, and the network is the final one; otherwise, continue the training. The calculation of error rate is shown as Equation (13).

$$E = 100 \frac{O_{\max} - O_{\min}}{KV} \sum_{v=1}^V \sum_{i=1}^K (Y_i(v) - Z_i(v))^2 \quad (13)$$

Where O_{\max} and O_{\min} are the maximum value and minimum value of the output in corresponding data set, and V is the sample size of the validation set, K is the number of output node. $Y_i(v)$ and $Z_i(v)$ are the actual output and object output of the ith output node corresponding to a verified sample v. V and K are used to normalize the above error formulas, which is to make the error rate free of impacts of validation set size and output nodes numbers.

4. After training, set weights for each network in accordance with the current error rate of each individual network in the validation set for the final integrating.

3.2. Selective negative correlation learning SNCL based on clustering

The selective negative correlation learning algorithm based on clustering includes two parts: the network generation and network selection. In the first part, the network generation means to generate fixed number of neural network with the method of ANN integrating learning training; in the second part, the network selection means to select part of the network from the current networks for next step with the selection algorithm.

To sum up, the whole process of the algorithm can be described as Algorithm 1 (each of the subset are S1, S2, ..., SN) .

Algorithm 1 selective algorithm based on clustering

Input: 2N networks

Output: N optimized networks

Set A is the final network set after selection, and set B is the network set without processing

Initial condition, A is empty and B is the current all the 2N networks

```

1. while (Set  $B$  is not empty)
2.     Select network Net  $i$  from set  $B$  randomly;
3.     Calculate Net  $i$  and the correlation value of other networks in  $B$ ;
4.     if (Net $j$  has the maximum correlation with Net  $i$ )
5.         Label Net  $j$  and Net  $i$  as a pair. Pair  $n$ ,  $n \leq N$ ;
6.         Delete Net  $j$  and Net  $i$  from  $B$ ;
7.     endif;
8. endwhile;
9. foreach (Pair  $n$ )
10.    if (accuracy of Net  $j$  is greater than that of Net  $i$ )
11.        Add Net  $j$  into set  $A$ ;
12.    else
13.        Add Net  $i$  into set  $A$ ;
14.    endif;
15. endforeach;
16. return set  $A$ ;
17. end

```

4. Experiments

4.1. Data set selection

At present, the feature extraction and recognition of underwater target are implemented by means of sequential signal structure, power spectrum characteristics and time-spectrum analysis. Under the current conditions in the laboratory, several underwater noises collecting services are set in real waters in this experiment to collect data. In terms of the actual observing value, the width of the river is a dynamic value, which is influenced by the external environment a lot. The target vessel, with the sounding equipment, moves in the river along different directions and at different speeds according to the experiment requirements.

The vessel moves with different sounding bodies and at different speeds. The experiment defines it as several different objects, and the specific data size is show in Table 1.

Table 1. Experiment data size and distribution

| Data content | Audio parameter | | | | total |
|--------------|-----------------|--------------|--------------------|-----------------|-------|
| | Channels | Quantization | Sampling frequency | Sampling number | |
| 3K | 1 | 2 | 48000 | 4800 | 4879 |
| 5K | 1 | 2 | 48000 | 4800 | 5045 |
| 9K | 1 | 2 | 48000 | 4800 | 4821 |
| 13K | 1 | 2 | 48000 | 4800 | 4836 |
| 15K | 1 | 2 | 48000 | 4800 | 5122 |

It can be seen in Table 1 that the experiment has carefully classified the three kinds of objects. The total number of the samples is 24703, 18000 of which are used as training set, taking 72.8% of all the samples. 6800 are used as test set, taking 27.5% of all the samples.

4.2. Underwater target recognition step

In this paper, MFCC is used as the characteristic parameter of extraction and recognition. In the experiment, the number of Triangular Bandpass Filters is set as 24, and the parameters from the second dimension to 13th dimension are selected (number of feature dimension is counted from 0), in which 12 dimensions are included. The MFCC feature parameter from 0 dimension to 13 dimensions extracted from the underwater target noise signal is shown in Figure 3. In addition, first difference processing is done for selected parameters, which may add dynamic feature of underwater target noise into feature parameters; however, the disadvantage is that it may double the dimension of feature parameter.

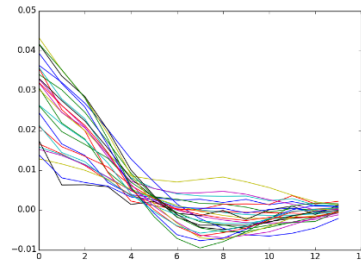


Figure 3. Extraction of 24 dimensional MFCC feature parameters from underwater Target noise

4.3. Results and analysis

The experiment includes two parts. The first is to select the optimized parameter in accordance with the experiment and the number of compressed feature vector. Then, verify the effectiveness of the SANCLbag in incremental learning and compare it with the common incremental learning algorithm Learn++ based on increment and fixed FSNCL algorithm.

In the experiment, the left frame number of each frame of sound is 25 to 30 after separating frame and eliminating mute segment. In the experiment of parameter comparison, the number of feature vector after compressing is set as 25, 20, 15 and 10. The compressed condition of parameter matrix shown in Figure 4.

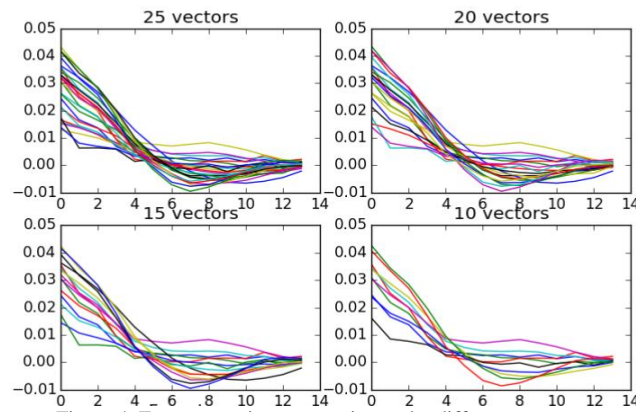


Figure 4. Feature matrix compression under different parameters

It can be seen from Figure 5 that the recognition accuracy rate rises with the increase of number of feature vector at first. When the number of vector reaches 15, the accuracy rate is the highest. Then, although the number of vectors increases, the accuracy rate begins to decrease and fluctuate. It can be seen that when the number of vectors is too small, the compression rate of feature data is rather high, and the loss of effective data is too high, making the extracted feature parameter cannot be learned by the model, which decreases the recognition accuracy rate of the model; however, when the number of feature vector reaches to certain value, the accuracy rate would not increase any more. Instead, it would decrease a little. Therefore, in the following experiments, the number of compressed feature vector is selected as 15.

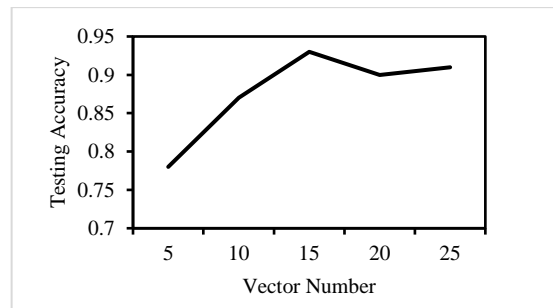


Figure 5. Recognition accuracy rate under different parameters

In incremental experiments, 10 groups of data set are trained in accordance with order to simulate the condition where the data sets may not come together in incremental learning, and the test data set in each test is the same. In order to observe

the relation between training time and recognition accuracy rate more clearly, the recognition results of the three algorithms are shown as the line chart in Figure 6. The abscissa is the incremental training time, and the ordinate is the accuracy rate. The three algorithms are presented by different types of lines.

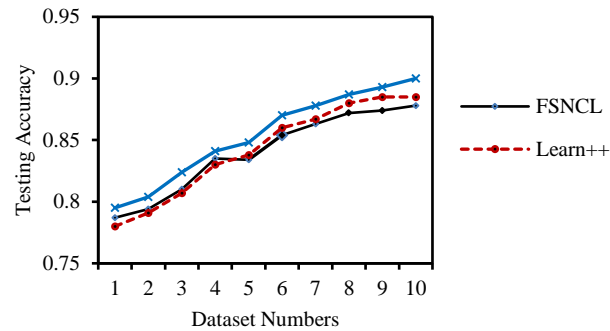


Figure 6. Incremental training recognition accuracy rate of three algorithms

Generally, it can be seen that the recognition accuracy rates of the three algorithms rises as the times of incremental training increases. But it is not a consistent rising; there are fluctuations. Also, the rising tendency increases slowly. The reason that why the recognition accuracy rates rise is that the model has learned more and more features of samples after the previous data training, which is also an important feature of incremental learning. However, when the training samples accumulate to certain degree, the model has learned and mastered most of the feature, and the new added sample would not add new features for the training samples. Therefore, the recognition accuracy rate would not rise consistently any more.

5. Conclusions

This paper has studied and improved the negative correlation incremental learning to enable it to solve the multi-classifier integration problems in underwater target recognition. Based on the NCL algorithm, this paper studies its advantages and disadvantages in training. Also, this paper proposes an adaptive NCL algorithm, by taking experiment formula as reference value when set up the initial value of hidden layer nodes, it is able to adaptively add or delete number of hidden nodes in training, and it also introduces the steepness factors to reduce the iteration times in the training. In addition, a new SANCLBag algorithm is proposed by combining it with Bagging algorithm with the selective integration method based on clustering and ordering. The algorithm has taken into consideration the accuracy and diversity of the model, and the experiment results show that the model taking SANCLBag as recognition module is of higher recognition accuracy rate. It is able to solve the feature integration and classification problems of underwater target effectively.

Acknowledgements

This work was funded by the National Natural Science Foundation of China under Grant (No. 61772152 and No. 61502037), and the Basic Research Project (No. JCKY2016206B001, JCKY2014206C002 and JCKY2017604C010).

References

1. Bauer, Eric, Kohavi, Ron, "An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants," *Machine Learning*, vol. 36, no. 1-2, pp. 105-139, 1999
2. L. Breiman, "Bagging Predictors," *Machine Learning*, vol., no., pp., 1996
3. J. X. Che, Y. L. Yang, "Stochastic Correlation Coefficient Ensembles for Variable Selection," *Journal of Applied Statistics*, vol. 44, no., pp. 1-22, 2017
4. M. S. Cheon, S. Ahmed, F. Alkhayyal, "A Branch-Reduce-Cut Algorithm for the Global Optimization of Probabilistically Constrained Linear Programs," in *Proceedings of Mathematical Programming, Ser B*, pp. 617-634, 2006
5. T. G. Dietterich, "Ensemble Methods in Machine Learning," *Proc 1st International Workshop on Multiple Classifier Systems*, 2000, vol. 1857, no. 1, pp. 1-15, 2000
6. Y. Ding, "Negative Correlation Learning Based on Neural network," *Computer Simulation*, vol. 24, no. 6, pp. 142-145, 2007
7. J. Friedman, T. Hastie, R. Tibshirani, "Special Invited Paper. Additive Logistic Regression: A Statistical View of Boosting," *Annals of Statistics*, vol. 28, no. 2, pp. 337-374, 2000
8. Y. Fuji "Artificial Neural Network." Springer Publishing Company, Incorporated, 2016
9. S. Hilbert "Correlation Coefficient." Springer International Publishing, 2017
10. N. Kasabov, "Evolving Fuzzy Neural Networks," *Algorithms, Applications and Biological Motivation*, vol., no., pp., 1998
11. N. Kasabov, "Evolving Fuzzy Neural Networks for Supervised/Unsupervised Online Knowledge-based Learning," *IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society*, vol. 31, no. 6, pp. 902-918, 2001

12. M. Korytkowski, L. Rutkowski, R. Scherer, "Fast Image Classification by Boosting Fuzzy Classifiers," *Information Sciences*, vol. 327, no. C, pp. 175-182, 2016
13. G. Krempel, V. Lemaire, R. Polikar, B. Sick, D. Kottke, A. Calma, "Interactive Adaptive Learning," vol., no., pp., 2017
14. L. I. Kuncheva "Ensemble Methods." John Wiley & Sons, Inc., 2014
15. R. Q. Leo Bbeiman, "Bagging Predictors," vol., no., pp., 2010
16. M. Lin, K. Tang, X. Yao, "Selective Negative Correlation Learning Algorithm for Incremental Learning," in *Proceedings of IEEE International Joint Conference on Neural Networks*, pp. 2525-2530, 2008
17. Y. Liu, "Enforcing Negativity in Negative Correlation Learning," in *Proceedings of IEEE International Conference on Information and Automation*, pp. 1122-1125, 2017
18. Y. Liu, X. Yao, "Ensemble Learning via Negative Correlation," *Neural Networks*, vol. 12, no. 10, pp. 1399-1404, 1999
19. F. L. Minku, H. Inoue, X. Yao, "Negative Correlation in Incremental Learning," *Natural Computing*, vol. 8, no. 2, pp. 289-320, 2009
20. A. Onwuegbuzie, L. Daniel, N. Leech, A. Onwuegbuzie, L. Daniel, N. Leech, A. Onwuegbuzie, L. Daniel, N. Leech, "Pearson Product-moment Correlation Coefficient," *Covariance*, vol., no., pp., 2012
21. N. C. Oza, "Online Ensemble Learning," in *Proceedings of Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, July 30 - August 3, 2000, Austin, Texas, USA*, pp. 1109, 2000
22. Polikar, Robi, "Ensemble Based Systems in Decision Making," *IEEE Circuits & Systems Magazine*, vol. 6, no. 3, pp. 21-45, 2006
23. R. Polikar, L. Upda, S. S. Upda, V. Honavar, "Learn++: An Incremental Learning Algorithm for Supervised Neural Networks," *Systems Man & Cybernetics Part C Applications & Reviews IEEE Transactions on*, vol. 31, no. 4, pp. 497-508, 2001
24. J. Sommars, J. Verschelde, "Pruning Algorithms for Pretropisms of Newton Polytopes," in *Proceedings of International Workshop on Computer Algebra in Scientific Computing*, pp. 489-503, 2016
25. A. H. Tan, "Adaptive Resonance Associative Map," *Neural Networks*, vol. 8, no. 3, pp. 437-446, 1995
26. K. Tang, M. Lin, F. L. Minku, X. Yao, "Selective Negative Correlation Learning Approach to Incremental Learning," *Neurocomputing*, vol. 72, no. 13-15, pp. 2796-2805, 2009
27. I. Tolstikhin, S. Gelly, O. Bousquet, C. J. Simongabriel, B. Schölkopf, "AdaGAN: Boosting Generative Models," vol., no., pp., 2017
28. N. C. Yeo, K. H. Lee, Y. V. Venkatesh, S. H. Ong, "Colour Image Segmentation Using the Self-organizing Map and Adaptive Resonance Theory," *Image & Vision Computing*, vol. 23, no. 12, pp. 1060-1079, 2005

Ming He graduated from the School of Computer Science and Technology, Harbin Institute of Technology, for the degree of Bachelor, and received his M.S. degrees in the Collage of Computer Science and Technology, Harbin Engineering University. He is a student member of CCF. His current research interests include computer underwater target detection and identification, machine learning, and deep learning.

Nianbin Wang received the PhD degree in computer science and technology from Harbin Institute of Technology, Harbin, China, in 2001. He is a student member of CCF, he has been a Professor with the Department of Computer Science and Technology, Harbin Engineering University. His interests include: Dataspace, Deep Learning, Data Integration.

Hongbin Wang received his PhD degree in computer application technology from Harbin Engineering University in 2010. He is currently an associate professor in the college of Computer Science and Technology of Harbin Engineering University. His research interests include information management, data integration, data space, semantic web, ontology and information system design.

Ci Chu received his Bachelor's degree in computer application technology from Harbin Engineering University in 2018. He is now studying for a Master's degree at the Harbin Engineering University. His research interests include deep learning, web architecture, data integration, data space, and data classification.

Songyan Zhong received his master's degree in computer science and technology from Tianjin University in 2012. He is currently an engineer in the Beijing Institute of computer technology and applications. His research interests include network architecture, networking protocol, data integration, data space.