Negative Selections in Ensemble Learning

Yong Liu
School of Computer Science and Engineering
The University of Aizu
Aizu-Wakamatsu, Fukushima 965-8580, Japan

***Abstract*—In the ensemble learning methods for training individual learners in a committee machine, two learning items should be optimized, including minimization of both the squared difference between the target and the learner's output and the estimated correlation between the learner and the rest of learners in the ensemble. The first term is to force each learner to learn the given data. The second term palys the role in having direct interactions between each individual and the rest of individuals in the ensemble. However, these two learning terms might not always help in the performance of the learned ensemble. Two different modifications are developed in adjusting minimization of these two terms in negative correlation learning. One is to increase the weight of the estimated correlation, while the other is to increase the squared difference on some selected data. The first modification was implemented in negative correlation learning by fixing the weight of the minimization of the squared difference and increasing the weight of the minimization of the estimated correlation . The second modification was to shift the target output so that the learned output could be in a wider range. It would lead the individual learners to be more different. Experimental results were carried out to show how the individual learners would change their learning behaviors under such two modifications in negative correlation learning.**

***Index Terms*—Neural network ensembles, negative correlation learning, correlation penalty.**

I. Introduction

Big data needs powerful learning systems to analyze nowadays. With the increased data, a learning system should have the flexible structures to adapt to the data in the learning process. However, in a single monolithic system, the learned knowledge is normally distributed among the learning nodes that are tightly related each other. If the structure of a monolithic system would be changed by adding some new learning nodes or deleting some existing nodes, the learned knowledge should be re-distributed among the changed learning nodes through relearning. If the relearning in a monolithic system would take too much time, the system might not be practical in the applications with data in changing.

Different to the monolithic system, a committee machine contains of a set of individual learners that learn a complex task together by subdividing the task [1], [2], [3], [4], [5] Because each learner only deal with a subtask of the original task, two learners could be independent or even negatively correlated. Therefore, if the committee machine would change its structure by adding a new learner or deleting an existing learners, other learners might need no changes or less changes as long as they do not depend on those changed learners much. Such loose relations among the individual learners let a committee machine be more flexible in the structure change and adaptive learning the new coming data.

In an ensemble learning system, all learners should be cooperative in covering the different learning tasks among them. It is obvious that it is not helpful to combine a set of the same or very similar learners in a committee machine. There are many different ways to create those different learners in a committee machine. Ensemble learning methods could be divided into following three different groups based on how the interactions are implemented in the learning: independent learning [6], [7]; sequential learning [5]; and concurrent learning [8], [9], [10], [11], [12], [13], [14], [15], [16], [17]. There are no interactions in independent learning that often builds individual learners separately. One approach in independent learning is through sampling the data by cross-validation or resampling the data with repetition. Although it is likely that the correlations among the individual learners could be drastically reduced in learning the subsets rather than the full data, there are no interactions when they are trained independently on the different subsets. It could happen that some data might not be learned well in such independent learning.

Partial interactions appear in sequential learning that normally filters the data based on the previously trained learners [5]. Boosting algorithm is a well-known sequential learning. In theory, a boosting algorithm is able to build a strong committee machine with arbitrary accuracy even if the individual learners among it are weak and just slightly better than random guessing. In practice, the learners are often not so weak after learning the filtering the data. Besides, such interactions are only applied sequentially from the previously trained learners to the later trained learners. The early trained learners would not be affected by the later trained learners.

In the ensemble learning methods for training individual learners in a committee machine, two learning items should be optimized, including minimization of both the squared difference between the target and the learner's output and the estimated correlation between the learner and the rest of learners in the ensemble. The first term is to force each learner to learn the given data. The second term palys the role in having direct interactions between each individual and the rest of individuals in the ensemble. However, these two learning terms might not always help in the performance of the learned ensemble. If the minimization of the squared difference would dominate the learning directions, all the individuals could be too similar to be helpful in combination. If the minimization of the estimated correlation would be done too much, even the combination of all the individual learners might not learn the given data well. Two different modifications are developed in adjusting minimization of these two terms in negative correlation learning [8], [9], [10], [11], [12]. One is to increase the weight of the estimated correlation, while the other is to increase the squared difference on some selected data. The first

modification was implemented in negative correlation learning by fixing the weight of the minimization of the squared difference and increasing the weight of the minimization of the estimated correlation . The second modification was to shift the target output so that the learned output could be in a wider range. It would lead the individual learners to be more different. Experimental results were carried out to show how the individual learners would change their learning behaviors under such two modifications in negative correlation learning.

The rest of this paper is organized as follows. Section II describes two different negative correlation learning methods by adaptive learning forces and directions. Section III show how the adaptive learning forces and directions could change the learning behaviors at both the ensemble and individual levels. Section IV summarizes the comparison results, and point out some possible improvements.

II. Negative Correlation Learning with Adaptive Learning Forces and Directions

Negative correlation learning (NCL) [18] is a simultaneous ensemble learning method in which all individual learners are trained at parallel by exchanging the the learned knowledge in learning each given data. Particularly, each individual neural network could learn to be closer or away from a given data based on how this data point has been learned by the other individuals in the ensemble. For the i-th neural network, its learning error function E_i is formed from two terms in negative correlation learning, including the sum of the squared error between the target and output by the i-th neural network, and a sum of the squared differences between the i-th neural network and the rest of neural networks on the given data

$$\begin{aligned} E_i &= \frac{1}{N}\Sigma_{n=1}^{N} E_i(n) \\ &= \frac{1}{N}\Sigma_{n=1}^{N}\frac{1}{2}\left[(F_i(n)-y(n))^2 - \lambda(F_i(n)-F(n))^2\right] \quad (1) \end{aligned}$$

In Eq.(1), $F_i(n)$ and $F(n)$ denote the outputs of the i-th neural network and the ensemble on the nth training pattern $\mathbf{x}(n)$ in the data D, respectively. λ is a parameter for adapting the amounts of the estimated correlation between the i-th neural network and the ensemble. The combination output $F(n)$ of the ensemble is given by simple average with the same weight:

$$F(n) = \frac{1}{M}\Sigma_{i=1}^{M} F_i(n) \quad (2)$$

where M is the number of individual neural networks combined in the ensemble.

Both the squared error and the estimated correlation are minimized simultaneously in the right side of Eq.(1). The minimization directions are decided by the partial derivative of the error E_i to the output F_i on $\mathbf{x}(n)$, which is given by

$$\frac{\partial E_i(n)}{\partial F_i(n)} = (1-\lambda)(F_i(n)-y(n)) + \lambda(F(n)-y(n)) \quad (3)$$

The weight λ was limited in the range from 0 to 1 in the implementation of NCL for the convergence of learning. It has been shown that this setting could let NCL to create the negatively correlated neural networks when both $F_i(n)$ and $F(n)$ could be either larger or smaller than $y(n)$. In another word, the two terms of $(F_i(n)-y(n))$ and $(F(n)-y(n)$ should have the different signs. Otherwise, the learning direction decided by the positive weighted sum of these two terms would be the same as the learning direction decided by either $(F_i(n)-y(n))$ or $(F(n)-y(n)$ alone. In learning a two-class classification problem, $y(n)$ in D is normally set to 0 or 1. When the sigmoid function is used for defining the output of individual neural networks, both $F_i(n)$ and $F(n)$ given by Eq.(2) have the values in the range of from 0 to 1. Therefore, $(F_i(n)-y(n))$ and $(F(n)-y(n))$ would always have the same sign no matter how different they are. With such setting, NCL would fail in generating negatively correlated neural networks.

When $(F_i(n)-y(n))$ and $(F(n)-y(n))$ would always have the same sign, $(1-\lambda)(F_i(n)-y(n))$ and $\lambda(F(n)-y(n))$ could have the different signs if λ is larger than 1. This increased λ would let NCL to be able to generate much different individual neura networks. However, it should be awared that $-(F_i(n)-y(n))$ would actually let the i-th neural network not to learn the given target. If λ would be too big, none of individual neural networks might learn the given data. Anyhow, λ could be adjusted so that the learning convergence could still be reached, or the certain performance on the given learning data should be satisfied. In order to satisfy the learning convergence, the larger λ than 1 is only conditionally applied in NCL. If $F_i(n)$ is closer to the target $y(n)$ than $F(n)$ is, $\lambda = 1$ will be set in Eq.(3) so that the learning directions would be decided by

$$\frac{\partial E_i(n)}{\partial F_i(n)} = F(n) - y(n) \quad (4)$$

If $F_i(n)$ is further away from the target $y(n)$ than $F(n)$ is, $\lambda > 1$ will be chosen in Eq.(3).

Besides enlarging λ, the values of $(F_i(n)-y(n))$ and $(F(n)-y(n))$ might have the different signs if $y(n)$ would be some values between 0 and 1. One simple way is to replace $y(n)$ with $y(n) = |y(n)-\alpha|$ with $0 < \alpha < 0.5$. After shifting the target values, $F_i(n)$ and $F(n)$ could be either smaller or bigger than the new shifted targets. When the error signals of $(F(n)-Y(n))$ and $(F_i(n)-Y(n))$ become the same sign, weight learning in each learner by $(F(n)-Y(n))$ or $(F_i(n)-Y(n))$ are in the same direction. However, when their sign values are different, weight learning defined by these two terms would go the opposite directions. Another modification on NCL could be developed based on the the sign values of $(F(n)-Y(n))$ and $(F_i(n)-Y(n))$. If $(F(n)-Y(n))$ and $(F_i(n)-Y(n))$ have the same sign, the learning signal would be simplized by

$$\frac{\partial E_i(n)}{\partial F_i(n)} = \beta(F(n) - |y(n)-\alpha|) \quad (5)$$

where β is between 0 and 1. If $(F(n)-Y(n))$ and $(F_i(n)-Y(n))$ have the different signs, the learning signal would be

$$\frac{\partial E_i(n)}{\partial F_i(n)} = F(n) - |y(n)-\alpha| \quad (6)$$

After two modifications, NCL would have three learning parameters of λ, α, and β, which could be represented as NCL(λ,α,β). The original NCL is NCL(λ,0,1) with $\alpha = 0$ and $\beta = 1$. In the next section, experimental results would show how the performance of NCL(λ,0,1) and NCL(1,0.25,β) would change with the different values of λ and β.

III. Simulation Results of the Modified NCL

The error rates on both the training sets and the testing sets have been measured on the ensemble levels and the individual levels through the whole learning period. In a classification problem, an error rate on a data set indicates the percentage of wrongly classified data points among all the data points in the data set. Therefore, the lower the error rates are, the better performance the learned models are. Normally, only the testing error rates were given in the results to show the generalizations. Besides the testing error rates, the training error rates were provided in the section in order to show how well the models had learned the given data points.

It is important to keep the diversity among the individuals in a committee machine. One simple way to represent such diversity among the individuals is through the overlapping rate between the two individuals, which counts the percentage of the two individuals' having the same output on the measured data set. When the two individuals output the exactly the same answer on every data point in the measured data set, their overlapping rate becomes one. In contrast, when the two individuals output the opposite answers on every data point in the measured data set, their overlapping rate is zero.

The modified NCL were tested in the following two data sets from the UCI machine learning bench-mark. The first data set is the breast cancer data set whose purpose is to classify a tumour as either benign or malignant based on cell descriptions gathered by microscopic examination. The data set contains 9 attributes and 699 examples with 458 benign examples and 241 malignant examples. The second data set is the heart disease data set on predicting the presence or absence of heart disease given the results of various medical tests carried out on a patient. This database includes 13 attributes and 270 examples.

The average results were obtained from the averages of 5 runs by 10-fold cross-validation. Therefore, each given result value was calculated from total 50 runs. Ten neural networks are used in the neural network ensemble in which each neural network has one hidden layer and ten hidden nodes. Six values of λ at 1, 1.1, 1.2, 1.3, 1.4, and 1.5 were tested in NCL(λ,0,1) while ten values of β were checked in NCL(1,0.25,β) from 0.1 to 1 with the step size of 0.1. The training time was set at 1000 epochs.

A. Results of NCL(λ,0,1)

Tables I, II, and III present the average results of the error rates of both the ensembles and individuals, and overlapping rates among them in NCL(λ,0,1) with λ from 1 to 1.5 at epoch 50 and 1000, respectively. With the the increased λ in NCL(λ,0,1), the ensemble training error rates showed the greatly different trends on the two data sets. On the cancer data, it was rather easy to learn so that the ensemble training error rates quickly dropped below 0.03 just after the first 50-epoch training. After another 1500-epoch training, the training error rates were able to be further reduced to less than 0.02 at $\lambda \leq 1.1$. However, the ensemble testing error rates were above 0.035 at $\lambda \leq 1.1$. When λ is between 1.2 and 1.5, the ensemble testing error rates stayed around 0.03 from the training epoch 50 to 1000 with little changes of the ensemble training error rates.

In comparisons on the heart data, the ensemble training error rates fell to 0.079 or larger after the first 50-epoch training. At $\lambda = 1$, the ensemble training error rates decreased to 0.019 after another 1500-epoch training. At $\lambda \geq 1.1$, the ensemble training error rates became 0.051 or larger although they were still able to be reduced in the later 1500-epoch training. Overfitting did not appear on the heart data so that NCL(1,0,1) achieved the best performance with the ensemble testing error rates at 0.061.

For the individual training error rates, they generally became much bigger than those values of the ensembles' except for those on the cancer data $\lambda \leq 1.1$. It suggests that NCL(λ,0,1) is able to create rather weak learners with the error rates as high as 0.343. When the learners were weak, thy had rather similar performances on both the training and testing sets.

For the average overlapping rates, NCL(1,0,1) with the default setting could hardly generate different individual learners with the overlapping rates 0.0974 on the cancer training sets and 0.922 on the heart training sets. With the increased λ, NCL(λ,0,1) was able to create more different individuals that just had the overlapping rates around 0.6 on the heart data sets.

TABLE I. Average error rates of the ensembles by NCL(λ,0,1) with the increased λ on the cancer and heart data sets at 50 and 1000 respectively. Tr represents the training error rates while Te indicates the testing error rates.

	Cancer		Heart	
	50	1000	50	1000
λ	(Train, Test)	(Train, Test)	(Train, Test)	(Train, Test)
1.0	0.025, 0.035	0.016, 0.039	0.079, 0.124	0.019, 0.061
1.1	0.024, 0.035	0.019, 0.035	0.082, 0.134	0.051, 0.090
1.2	0.023, 0.031	0.021, 0.032	0.092, 0.134	0.066, 0.109
1.3	0.022, 0.030	0.021, 0.030	0.100, 0.149	0.078, 0.118
1.4	0.022, 0.029	0.021, 0.031	0.101, 0.147	0.084, 0.127
1.5	0.022, 0.029	0.021, 0.030	0.113, 0.158	0.095, 0.131

TABLE II. Average error rates of the individuals by NCL(λ,0,1) with the increased λ on the cancer and heart data sets at 50 and 1000 respectively.

	Cancer		Heart	
	50	1000	50	1000
λ	(Train, Test)	(Train, Test)	(Train, Test)	(Train, Test)
1.0	0.032, 0.042	0.023, 0.047	0.127, 0.173	0.051, 0.093
1.1	0.040, 0.050	0.034, 0.052	0.157, 0.195	0.125, 0.160
1.2	0.127, 0.133	0.133, 0.141	0.221, 0.248	0.203, 0.225
1.3	0.150, 0.156	0.146, 0.153	0.279, 0.296	0.260, 0.287
1.4	0.191, 0.197	0.184, 0.189	0.304, 0.317	0.292, 0.307
1.5	0.217, 0.221	0.217, 0.222	0.343, 0.362	0.331, 0.345

B. Results of NCL(1,0.25,β)

Tables IV, V, and VI display the average results of the error rates of both the ensembles and individuals, and overlapping rates among them in NCL(1,0.25,β) with β from 0.1 to 1 at epoch 50 and 1000, respectively. With the the increased β in NCL(1,0.25,β), the ensemble training error rates became smaller, and even reached 0.002 at $\beta = 1$ on the heart set. When the overfitting was not observed on the heart set, the

TABLE III. AVERAGE OVERLAPPING RATES AMONG THE INDIVIDUALS BY NCL(λ,0,1) WITH THE INCREASED λ ON THE CANCER AND HEART DATA SETS AT 50 AND 1000 RESPECTIVELY.

	Cancer		Heart	
	50	1000	50	1000
λ	(Train, Test)	(Train, Test)	(Train, Test)	(Train, Test)
1.0	0.973, 0.973	0.974, 0.962	0.851, 0.825	0.922, 0.885
1.1	0.955, 0.954	0.956, 0.948	0.793, 0.777	0.812, 0.791
1.2	0.786, 0.787	0.771, 0.771	0.699, 0.693	0.711, 0.706
1.3	0.751, 0.750	0.755, 0.754	0.657, 0.655	0.676, 0.663
1.4	0.698, 0.696	0.703, 0.708	0.654, 0.655	0.664, 0.662
1.5	0.674, 0.673	0.674, 0.672	0.613, 0.609	0.622, 0.621

ensemble testing error rates achieved 0.033. However, light overfitting happened in NCL(1,0.25,β) on the cancer set, which NCL(1,0.25,β) obtained the ensemble testing error rates 0.038 at $\beta = 1$, and 0.029 at at $\beta = 0.3$.

As shown in Table V, the smaller β is, the weaker the individuals by NCL(1,0.25,β) are. Actually the individuals by NCL(1,0.25,0.1) were close to be random guessing on the heart set. Rather small differences existed between the training error rates and the testing error rates by NCL(1,0.25,β).

The individuals by NCL(1,0.25,β) were quite different. It was surprised to see that the overlapping rates could be even less than 0.5 at $\beta = 0.1$. However, the results suggest that NCL(1,0.25,0.1) might not be able to learn the data well. In another word, β should not be too small so that NCL(1,0.25,β) is still able to learn, and generate different individual learners at the same time.

TABLE IV. AVERAGE ERROR RATES OF THE ENSEMBLES BY NCL(1,0.25,β) WITH THE INCREASED β ON THE CANCER AND HEART DATA SETS AT 50 AND 1000 RESPECTIVELY.

	Cancer		Heart	
	50	1000	50	1000
β	(Train, Test)	(Train, Test)	(Train, Test)	(Train, Test)
0.1	0.194, 0.195	0.040, 0.045	0.437, 0.448	0.437, 0.448
0.2	0.026, 0.030	0.022, 0.030	0.388, 0.396	0.371, 0.382
0.3	0.024, 0.029	0.021, 0.029	0.235, 0.250	0.140, 0.167
0.4	0.023, 0.031	0.021, 0.031	0.129, 0.153	0.100, 0.130
0.5	0.023, 0.031	0.021, 0.032	0.119, 0.152	0.096, 0.131
0.6	0.023, 0.033	0.021, 0.032	0.112, 0.153	0.083, 0.120
0.7	0.024, 0.034	0.020, 0.035	0.103, 0.141	0.067, 0.106
0.8	0.024, 0.035	0.020, 0.034	0.097, 0.133	0.052, 0.083
0.9	0.025, 0.034	0.018, 0.038	0.098, 0.144	0.023, 0.072
1.0	0.026, 0.034	0.009, 0.038	0.107, 0.150	0.002, 0.033

IV. CONCLUSIONS

This paper discussed two ways of changing the error signals so that more different individual learners could be generated in an ensemble for the classification problems. When each individual learner is able to decide its learning direction and learning distance based on its relationship with the rest individuals, it naturally builds its self-awareness. When the output values of the individual learners are within the range defined the target values specified in the classification problems, individual learners might choose to learn to be away from the target values instead of getting closer to the targets. When the output values of the individual learners could be larger or less than the target values, individuals were encouraged to go further when their learning error signals bring them away from the rest individuals.

TABLE V. AVERAGE ERROR RATES OF THE INDIVIDUALS BY NCL(1,0.25,β) WITH THE INCREASED β ON THE CANCER AND HEART DATA SETS AT 50 AND 1000 RESPECTIVELY.

	Cancer		Heart	
	50	1000	50	1000
β	(Train, Test)	(Train, Test)	(Train, Test)	(Train, Test)
0.1	0.420, 0.422	0.327, 0.320	0.493, 0.498	0.494, 0.498
0.2	0.292, 0.294	0.274, 0.276	0.484, 0.485	0.478, 0.479
0.3	0.278, 0.281	0.270, 0.275	0.436, 0.440	0.408, 0.413
0.4	0.272, 0.276	0.265, 0.271	0.361, 0.373	0.341, 0.353
0.5	0.266, 0.270	0.265, 0.271	0.348, 0.364	0.331, 0.346
0.6	0.261, 0.264	0.264, 0.273	0.334, 0.354	0.307, 0.326
0.7	0.253, 0.257	0.264, 0.274	0.319, 0.340	0.293, 0.315
0.8	0.243, 0.248	0.261, 0.273	0.308, 0.330	0.282, 0.305
0.9	0.230, 0.235	0.263, 0.276	0.302, 0.325	0.262, 0.289
1.0	0.228, 0.232	0.262, 0.279	0.293, 0.318	0.230, 0.253

TABLE VI. AVERAGE OVERLAPPING RATES AMONG THE INDIVIDUALS BY NCL(1,0.25,β) WITH THE INCREASED β ON THE CANCER AND HEART DATA SETS AT 50 AND 1000 RESPECTIVELY.

	Cancer		Heart	
	50	1000	50	1000
β	(Train, Test)	(Train, Test)	(Train, Test)	(Train, Test)
0.1	0.474, 0.474	0.529, 0.530	0.455, 0.455	0.455, 0.455
0.2	0.554, 0.553	0.573, 0.573	0.450, 0.450	0.452, 0.452
0.3	0.568, 0.567	0.575, 0.574	0.464, 0.465	0.478, 0.477
0.4	0.574, 0.574	0.581, 0.580	0.509, 0.507	0.519, 0.518
0.5	0.580, 0.580	0.580, 0.579	0.518, 0.515	0.529, 0.526
0.6	0.586, 0.587	0.581, 0.579	0.532, 0.528	0.551, 0.544
0.7	0.595, 0.595	0.580, 0.578	0.548, 0.544	0.560, 0.551
0.8	0.607, 0.606	0.583, 0.580	0.559, 0.555	0.569, 0.556
0.9	0.623, 0.622	0.581, 0.577	0.569, 0.565	0.584, 0.568
1.0	0.626, 0.627	0.579, 0.575	0.588, 0.583	0.613, 0.598

In this paper, α was fixed at 0 or 0.25 in NCL(λ,α,β) where λ and β were tested separately. It would be interesting to see how the individual learners would behave when these three parameters would be adjusted at the same time. λ,α and β should be set at different values in the different learning periods while each individual learner might have their own different setting. In such ways, a truly adaptive NCL could be developed.

ACKNOWLEDGMENT

This work is partially supported by Kahenhi Grant #15K00343 to Y. Liu from the Japan Society for the Promotion of Science (JSPS).

REFERENCES

[1] O. G. Selfridge. Pandemonium: a paradigm for learning. In *Mechanisation of Thought Processes: Proc. of a Symp. Held at the National Physical Lab.*, pages 513–526. HMSO, London, 1958.

[2] N. J. Nilsson. *Learning Machines: Foundations of Trainable Pattern-Classifying Systems*. NY: McGraw Hill, New York, 1965.

[3] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990.

[4] H. Drucker, C. Cortes, L. D. Jackel, Y. LeCun, and V. Vapnik. Boosting and other ensemble methods. *Neural Computation*, 6:1289–1301, 1994.

[5] R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.

[6] D. Sarkar. Randomness in generalization ability: a source to improve it. *IEEE Trans. on Neural Networks*, 7(3):676–685, 1996.

[7] J. Kim, J. Ahn, and S. Cho. Ensemble competitive learning neural networks with reduced input dimensions. *International Journal of Neural Systems*, 6(2):133–142, 1995.

[8] Y. Liu, Q. Zhao, and Y. Pei. Error awareness by lower and upper bounds in ensemble learning. In *Proceedings of 2015 11th International Conference on Natural Computation*.

[9] Y. Liu, Q. Zhao, and Y. Pei. Bounded learning for neural network ensembles. In *Proceedings of IEEE International Conference on Information and Automation*.

[10] Y. Liu, Q. Zhao, and Y. Pei. Balanced ensemble learning with adaptive bounds. In *Proceedings of the 2015 IEEE ICSPCC2015*.

[11] Y. Liu. Negative correlation learning with difference learning. In *Communications in Computer and Information Science*, volume 575, pages 264–274. Springer, 2016.

[12] Y. Liu, Q. Zhao, and Y. Pei. From low negative correlation learning to high negative correlation learning. In *Proceedings of 2014 International Joint Conference on Neural Networks (IJCNN 2014)*.

[13] Y. Liu, Q. Zhao, and Y. Pei. Control of correlation in negative correlation learning. In *Proceedings of 2014 10th International Conference on Natural Computation*.

[14] Y. Liu, Q. Zhao, and Y. Pei. Ensemble learning with correlation-based penalty. In *Proceedings of 2014 World Ubiquitous Science Congress*.

[15] Y. Liu, Q. Zhao, and Y. Pei. Transition learning between balanced ensemble learning and negative correlation learning. In *Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics*.

[16] Y. Liu, Q. Zhao, and N. Yen. Make decision boundary smoother by transition learning. In *Proceedings of the 5th International Conference on Awareness Science and Technology*.

[17] Y. Liu. Transition learning for creating diverse neural networks. In *Proceedings of the 6th International Conference on BioMedical Engineering and Informatics*.

[18] Y. Liu and X. Yao. Simultaneous training of negatively correlated neural networks in an ensemble. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 29(6):716–725, 1999.