

## Accepted Manuscript

A deep stochastic weight assignment network and its application to chess playing

Zhi-Wang, Xi-Zhao Wang

PII: S0743-7315(17)30243-5  
DOI: <http://dx.doi.org/10.1016/j.jpdc.2017.08.013>  
Reference: YJPDC 3734

To appear in: *J. Parallel Distrib. Comput.*

Received date: 22 February 2017  
Revised date: 3 July 2017  
Accepted date: 21 August 2017

Please cite this article as: Zhi-Wang, X. Wang, A deep stochastic weight assignment network and its application to chess playing, *J. Parallel Distrib. Comput.* (2017), <http://dx.doi.org/10.1016/j.jpdc.2017.08.013>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.



# A Deep Stochastic Weight Assignment Network and its Application to Chess Playing

ZHI-WANG<sup>a</sup>, XI-ZHAO WANG<sup>b(\*)</sup>

<sup>a</sup> College of Computer Science and Technology, Hebei University, Baoding, 071002, China

<sup>b</sup> College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, 518060

(\*) Corresponding author E-MAIL: xizhaowang@ieee.org

## Abstract

Chinese chess is an ancient game in which the chess situation is the information ensemble of

chess pieces' spatial locations and interrelations. The situation evaluation plays an extremely important role in the policy decisions of Chinese chess game. However, the situation evaluation is too complex for human to cover every detail with naked eyes. The deep stochastic weight assignment network (DSWAN) proposed in this paper to classify the situations in advantages and disadvantages can solve above problem. DSWAN is a type of multi-layer perception (MLP) which is divided into two main components: unsupervised feature extractor formed by multi-layer auto encoder and supervised classifier trained by stochastic weight assignment network (SWAN). We summarize a series of chess situation features by gathering specialized knowledges of Chinese chess, and these features are proved valid to estimate the situation. Another highlight of paper is that the auto encoder is constrained by  $L_{1/2}$  regularization. By doing so, it can bring more sparsity to data set, make situation features more representative and ease the overfitting trouble of SWAN.

*Keywords:* Chinese chess game, chess situation analysis, deep stochastic weight assignment network (DSWAN), auto encoder,  $L_{1/2}$  regularization, stochastic weight assignment network (SWAN)

Chinese chess is an old game which can be traced back to the Warring States Period. Red side chess player and black side chess player control their own chess pieces, follow common game rules, take flexible strategies, and finally checkmate the opponent's king by capturing chess piece step by step. In the field of artificial intelligence (AI), machine game is a typical research direction. In the imagination of AI researchers, machine should accumulate its own chess experience gradually, master similar strategies, and have capabilities to defeat human players. AI researches in other board games have attached to some gorgeous achievements. Deep Blue, a super computer chess player, which is designed by Hsu [1] defeated temporal world chess champion Kasparov by 3.5 to 2.5 in 1997. A computer program named AlphaGo developed by Silver and Huang [2] beat Go master Lee who is the 9-dan go professional player of South Korea by 4 to 1 in March 2016. The research of Chinese chess machine game began early in the 1980s [3]. But AI research referred to Chinese chess has not obtained such breakthrough like chess [4] and go. It remains on some traditional methods, e.g., alpha-beta pruning [5-8] to optimize the search algorithm, Chinese chess opening base [9] to store the beginning strategies of the game and transposition table [10-11] to record the node which has been searched to reduce the searching time. So, we import some advanced machine learning methods to Chinese chess game.

In the play of Chinese chess, the evaluation of chess situation is very critical. The chess situation is the information ensemble of chess pieces' spatial locations and interrelation on chess board in a step of the Chinese chess game. At the beginning of a game, all the chess pieces are placed on the Chinese chess board by rule, and the initial chess situation arises. In the process of the game, some of pieces are moved in some position that decided by players. With the piece's translocation, a new chess situation generates. Both of machine and human players decide to adopt whether offensive strategies or defensive strategies by analyzing chess situation comprehensively. Because of the complexity and fuzziness of chess situation, a machine can handle this task better than an emotional human player with the help of machine learning methods. Research results already existed are always focused on founding a function [12] to evaluate chess situation. The independent variables of the function are the situation's properties such as relative location of chess pieces, different values of pieces, chess pieces' mobility, and their offensive and defensive relationships. Although these properties are representative, the dependent variable of situation evaluation function is a single value that cannot reflect the details of chess situation. This paper develops a group of Chinese chess situation features which are multi-dimensional to cover every corner of a chess situation. Then the deep stochastic weight assignment network (DSWAN) is proposed to judge the situation.

DSWAN is a multi-layer perception (MLP) whose former two hidden layers as a whole are an auto encoder and the last one hidden layer is a neural network classifier trained by stochastic weight assignment network (SWAN). SWAN is a new rising algorithm [13-16] to train the single-hidden layer feedforward neural networks (SLFNs). The core concept of SWAN is that the weights and biases between the input layer and the hidden one are randomly generated and the weights of output layer are analytically computed through solving a generalized inverse of matrix. Unlike some traditional learning algorithms, e.g., back-propagation (BP) [17] that the weights and biases should be tuned by gradient descent and errors back propagation in iterations, the iteratively tuning is not required in SWAN, so the learning speed is accelerated. And the smallest norm of weights [14] is included in the cost function that the generalization ability is improved. SWAN's fast learning speed and high testing accuracy have attracted many developers [18-22] to adopting

it in machine learning projects. Auto encoder [23-24] in the former two hidden layers is an unsupervised feature extractor, in which the SWAN core concept is also adopted. Added  $L_{1/2}$  regularization [25-26] in the auto encoder makes the Chinese chess situation data set sparser than the original data set. From the experiments, DSWAN's classification performance is illustrated on the Chinese chess situation data set which is founded from Chinese chess middlegame situations in 50000 chess manuals.

This paper is organized as follows. Section II sets forth the Chinese chess situation feature extracting method and data set. In Section III, basic SWAN algorithm and auto encoder structure is introduced, respectively. We present the proposed DSWAN framework and  $L_{1/2}$  regularization sparsity-constrained auto encoder in Section IV. Experiment results are given in Section V and Section VI concludes the paper.

## II CHINESS CHESS SITUATION FEATURE EXTRACTING METHOD

Feature extraction is the vital component of machine learning projects [27-32]. Highly representative features can accelerate the learning process. We propose a method to extract the features of Chinese chess situation. Chinese chess game [5] tools contain a rectangle chess board which is printed with 10 horizontal lines and 9 vertical lines, thus there are 90 cross points on the board. On some of these cross points, there are 16 red pieces and 16 black pieces which can be categorized to 7 arms: king, advisor, elephant, horse, chariot, cannon, and soldier. A chess solution is composed of the chess board with pieces which is similar to a 10 by 9 matrix. The chess piece is regarded as the object to extract features. And all the pieces' features constitute the situation's features. Features of one Chinese chess piece are as follows.

- 1) Existence, in a chess situation, not all the pieces exist on the board. If piece A has been captured, A's existence is assigned to 0, otherwise it is assigned to 1.
- 2) Location, the chess board approximates a 10 by 9 matrix. Every piece on board has an independent location information. It is represented by an integer coordinate  $(X, Y)$ , where X and Y are the row index and column index of Chinese chess board respectively.
- 3) Location-based value, Chinese chess piece's value varies with the change of piece's location. For example, there is Chu River Han Border (CRHB) which divides the board from the middle and 3 by 3 areas called Palace in both sides on the chess board. Soldier A which has crossed CRHB has the higher value than soldier B which has not crossed CRHB. Soldier A would get a higher value if it enters the opposite Palace.
- 4) Move number, in the current situation, every piece would have countable legal move methods.
- 5) Mobility, the more move methods, the more mobility. Different pieces have different mobility weights. The mobility of one piece is the product of mobility-move weight and the number of legal moves.
- 6) Threats, in the current situation, one piece would be captured by an opposite piece in the next step. Every piece is assigned a unique number to be identified. And the threat of one piece equals to the product of the offensive piece's unique number and location-based value.
- 7) Protection, in the current situation, one piece's location is in its own side piece's legal move paths. Then the protection of one piece equals to the product of its protective piece's unique number and location-based value.

- 8) Assault, in the current situation, one piece can capture an opposite piece. Then the assault of one piece equals to the product of embattled piece's unique number and location-based value.

Every piece can be extracted these 8 features. In Chinese chess, the total piece number is 32. So, the total feature number of Chinese chess situation is 256.

For training the performance of chess situation classifier, the game result of each Chinese chess manual is set as the class attribute of each chess situation, i.e., when red player wins the game, the class attribute is 1, otherwise it turns to be 2.

A Chinese chess manual records a complete game process which is from the opening to the end. Each line of the manual notes down one move of one piece. So, a manual consists of a string of chess situations. The middlegame of manual is the most complicated period of game process so that it is very difficult for human to judge which player is in a dominant position. Therefore, making the middlegame situation as the research object is the most challenging and instructive issue. The Chinese chess situation feature data set of middlegame is designed.

TABLE I Chinese chess situation feature data set

X&F	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$	$F_8$	...	$F_{256}$	Class
$X_1$	1	215	2	3	0	0	3880	0	...	0	1
$X_2$	1	206	11	2	0	0	2231	0	...	0	2
$\vdots$	$\ddots$	$\vdots$	$\vdots$								
$X_N$	1	204	11	2	0	0	12571	0	...	0	1

Table I denotes the data set which is extracted from the middlegame situations of Chinese chess manual. In this data set, from  $F_1$  to  $F_8$  are eight features of Chinese chess piece A, e.g.,  $F_1$  represents whether piece A exists on the chess board;  $F_2$  represents piece A's location which is transformed from the integer coordinate  $(X, Y)$ ;  $F_3$  represents piece A's location-based value.

### III REVIEWS OF SWAN AND AUTO ENCODER

In this section, the basic SWAN conception and algorithm is briefly reviewed. Multi-layer auto encoder is also described in details.

#### A. SWAN

Stochastic weight assignment network (SWAN) is a fast learning algorithm [13-14] which has excellent regression and classification capability to train the single-hidden layer neural networks (SLFNs).

For N samples  $(x_i, t_i)$ , where  $x_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in R^n$  is the data of conditional attributes and  $t_i = [t_{i1}, t_{i2}, \dots, t_{im}]^T \in R^m$  is the data of class attributes, standard SLFNs with  $\tilde{N}$  hidden neurons and activation function  $g(x)$  are modeled as

$$\sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j + b_i) = o_j, j = 1, \dots, N \quad (1)$$

where  $w_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$  is the weight vector connecting the  $i$ th hidden neuron and the input neurons,  $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$  is the weight vector connecting the  $i$ th hidden neuron and the output neurons, and  $b_i$  is the threshold of the  $i$ th hidden neuron.  $w_i \cdot x_j$  denotes the inner product of  $w_i$  and  $x_j$ .

This model can approximate these  $N$  samples with zero error means that  $\sum_{j=1}^N \|o_j - t_j\| = 0$ , i.e.,

$$\sum_{i=1}^{\bar{N}} \beta_i g(w_i \cdot x_j + b_i) = t_j, j = 1, \dots, N \quad (2)$$

The above  $N$  equations can be written compactly as:

$$H\beta = T \quad (3)$$

where

$$H(w_1, \dots, w_{\bar{N}}, b_1, \dots, b_{\bar{N}}, x_1, \dots, x_N) = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_{\bar{N}} \cdot x_1 + b_{\bar{N}}) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_N + b_1) & \dots & g(w_{\bar{N}} \cdot x_N + b_{\bar{N}}) \end{bmatrix}_{N \times \bar{N}} \quad (4)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\bar{N}}^T \end{bmatrix}_{\bar{N} \times m} \quad \text{and} \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m} \quad (5)$$

where  $H$  expresses the output matrix of hidden layer in neural network; the  $i$ th column of  $H$  is the  $i$ th hidden neuron's output vector.

The essence of SWAN is that the weight  $w_i$  and threshold  $b_i$  are all randomly generated which is inspired by biological learning. Some parts of brain should have random neurons with parameters independent of the environment. So, the brain can work universally for the feature extraction, clustering, regression and classification with in almost zero time.

If the number of hidden neurons is equal to the number of distinct training samples, the matrix  $H$  will be square and invertible, and SLFNs can approximate these training samples with zero error. The solution of output weights  $\beta$  is equal to  $H^{-1}T$ . But generally, the number of hidden neurons is much less than the number of training samples. Thus,  $H$  is a non-square matrix and we cannot expect an exact solution of the system. In this case, the equation (2) could be derived:

$$T = H\beta + E \quad (6)$$

where  $E = [e_1 \dots e_N]_{N \times m}^T$  is the training error, so a cost function is defined as:

$$J = \sum_{j=1}^N \left( \sum_{i=1}^{\bar{N}} \beta_i g(w_i, b_i, x_j) - t_j \right)^2 \quad (7)$$

which can be simplified:

$$J = (H\beta - T)^T (H\beta - T) \quad (8)$$

Then the problem turns to be cost function minimization. The solution  $\hat{\beta}$  of minimum mean square error (MMSE) is requested:

$$\|H\hat{\beta} - T\| = \min_{\beta} \|H\beta - T\| \quad (9)$$

with minimum norm of the output weight  $\beta$  for the best generalization performance where  $\|\cdot\|$  denotes 2-norm.

The solution of MMSE of above equation is:

$$\hat{\beta} = H^+ T \quad (10)$$

where  $H^+$  is the Moore-Penrose generalized inverse [14] of matrix  $H$ .

The SWAN training algorithm can be summarized as follows:

Given a training data set  $\{(x_i, t_i) | x_i \in R^n, t_j \in R^m, i = 1, \dots, N\}$  where  $n$  is the dimension of input layer and  $m$  denotes the dimension of output layer.

- 1) Randomly assign the hidden neuron parameters, e.g., the input weights  $w_i$  and biases  $b_i$  for hidden neurons,  $i = 1, \dots, \tilde{N}$ ;
- 2) Calculate the hidden layer output matrix  $H$ ;
- 3) Obtain the output weight vector using (10).

The orthogonal projection method can be efficiently used for the calculation MP inverse to accelerate the learning speed:

$$\hat{\beta} = (H^T H)^{-1} H^T T \quad (11)$$

where  $H^\dagger = (H^T H)^{-1} H^T$ .

Given the testing data set  $x$ , the output function of SWAN is:

$$f(x) = h(x)\hat{\beta} \quad (12)$$

where  $h(\cdot)$  is the output matrix of the hidden layer with the same input weights and biases as in the training procedure.

## B. Multi-layer Auto Encoder

Given a simple SLFN, the learning object of the training data set is the training data itself, i.e., the output matrix of the SLFN is equal to the input matrix, then the auto encoder (AE) [23-24] is founded. The main function of the AE is dimension transformation and feature extraction. From Section II, the dimension of the Chinese chess situation feature data set is 256. Their effects to predict the advantages and disadvantages of chess situation is different. Some deep-seated relationships of these features are hiding. AE can help change the dimension of data set and the more representative features will be extracted.

For  $N$  samples  $x_i$ , where  $x_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in R^n$  is the input matrix and output matrix meanwhile. The neurons' number of input layer and output layer of AE are both set as  $n$ . The number of the hidden neurons is  $\tilde{L}$  which is also the objective dimension of data set to transfer. There are three different types of AE according to the hidden neurons' number:

- 1) If  $\tilde{L} < n$ , the dimension of the coded data set is compressed.
- 2) If  $\tilde{L} > n$ , the dimension of the coded data set is sparse.
- 3) If  $\tilde{L} = n$ , the dimension of the coded data set is equal to the original data set.

The training of AE is an unsupervised learning process, where the SWAN algorithm [33] can be also adopted. The input weights and hidden neurons' biases of the AE is randomly assigned. According to the equation (10), the output weights  $\hat{\beta}_1$  is computed by:

$$\hat{\beta}_1 = H^\dagger X \quad (13)$$

Then the transposition of output weights  $\hat{\beta}_1^T$  is the encoding factor. The coded matrix  $X_1$  of the original data set  $X$  is:

$$X_1 = g(X\hat{\beta}_1^T) \quad (14)$$

where  $g(\cdot)$  denotes the activation function.

Add a hidden layer to AE, and the first hidden layer's output matrix  $X_1$  is performed the input matrix of the second layer, then the multi-layer AE is founded.

Applied the same method as (14), the coded matrix  $X_2$  of the matrix  $X_1$  is:

$$X_2 = g(X_1\hat{\beta}_2^T) \quad (15)$$

The main learning process of the multi-layer AE is the confirmation of output weights 2-tuple

$\langle \hat{\beta}_1, \hat{\beta}_2 \rangle$ . Once the AE is trained to extract proper features, the output weights 2-tuple is fixed and not need to be fine-tuned.

#### IV DEEP STOCHASTIC WEIGHT ASSIGNMENT NETWORK

In this section, deep stochastic weight assignment network (DSWAN) for the Chinese chess situation predicting is proposed. A new  $L_{1/2}$  Sparsity-Constrained AE is also presented, which is an element of DSWAN.

##### A. DSWAN Framework

The deep stochastic weight assignment network (DSWAN) is a five-layer perception shown in Fig. 1 which is constructed from one input layer, one output layer, three hidden layers and parameters among them. The whole framework is divided into two different-acting phases [34-35]. In the former, a multi-layer AE is used to represent features of the data set. For the latter, the basic SWAN algorithm is utilized for the final decision making.

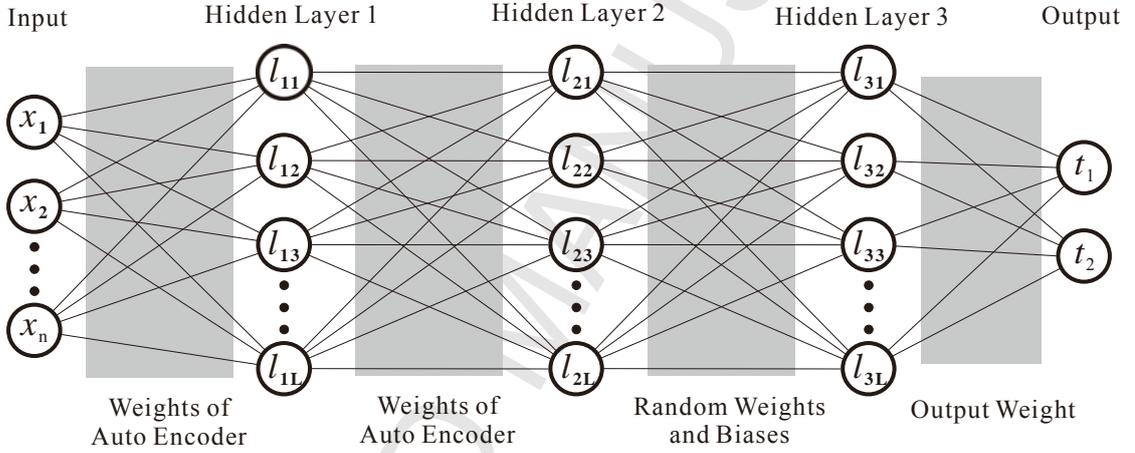


Fig. 1. The structure of DSWAN

Given the N-sample data set  $(x_i, t_i)$  where  $x_i \in R^n$  and  $t_i \in R^m$ . In the procedure of representation of data set features, the hidden neurons of each hidden layer of AE is L, i.e., the objective dimension of data set features to transform by AE is L. The new feature mapping of the original data set is N by L matrix  $X_2$  which is derived by (14, 15). The features of matrix  $X_2$  is more representative to approximate the class attributes data  $t_i$  than the original data set. The new training data set  $[X_2|T]$  is the substitution of the original N-sample data set. Then the basic SWAN algorithm is applied as a classifier in the final layer.

Unlike some traditional deep learning frameworks which the parameters need to be retrained iteratively using back propagation (BP) algorithm, the random feature mapping and universal approximation capability of SWAN are fully exploited both in AE and latter phase classifier of DSWAN.

##### B. $L_{1/2}$ Sparsity-Constrained AE

From Section III-B, the universal approximation capability of SWAN algorithm is applied in the AE. Then the  $L_{1/2}$  regularization [25] is added into the AE optimization to enhance the data set's feature sparsity. It has been proved by Xu [26] that the  $L_q$  regularization ( $0 < q < 1$ ) can

generate more sparse solutions than  $L_1$  regularization and meanwhile, the index  $1/2$  plays a representative role: whenever  $q \in [1/2, 1)$ , the smaller  $q$  is, the sparser solutions of  $L_q$  regularization will be obtained. Whenever  $q \in (0, 1/2)$ , the performance of  $L_q$  regularization has no obvious difference, but the solution of  $L_{1/2}$  regularization problem is easier to get than the  $L_0$  regularization problem. There have been some works [36] applying the  $L_{1/2}$  regularization to deal with sparsity problem. The  $L_{1/2}$  regularization problem is:

$$\min_{x \in \mathbb{R}^N} \{ \|H\beta - X\|^2 + \lambda \|\beta\|_{1/2} \} \quad (16)$$

where the  $X$  represents the input data,  $H$  denotes the random mapping output, and  $\beta$  is the hidden layer weight to be obtained. Because of the random mapping of the hidden layer feature representation,  $H$  is the output of hidden layer that it need not be optimized.

An iterative half thresholding algorithm [26] proposed by Xu provides a fast and effective solver for  $L_{1/2}$  regularization problem. It is defined as:

$$\beta_{n+1} = G_{\lambda_n \mu_n, 1/2}(\beta_n + \mu_n H^T (X - H\beta_n)) \quad (17)$$

where  $n$  denotes the iteration times and  $G_{\lambda_n, 1/2}$  is the half thresholding operator:

$$G_{\lambda_n, 1/2}(x) = \begin{cases} f_{\lambda_n, 1/2}(x), & |x| > \frac{\sqrt[3]{54}}{4} (\lambda_n)^{2/3} \\ 0, & \text{otherwise} \end{cases} \quad (18)$$

where

$$f_{\lambda_n, 1/2}(x) = \frac{2}{3} x \left( 1 + \cos \left( \frac{2\pi}{3} - \frac{2}{3} \varphi_{\lambda_n}(x) \right) \right) \quad (19)$$

with

$$\varphi_{\lambda_n}(x) = \cos^{-1} \left( \frac{\lambda_n}{8} \left( \frac{|x|}{3} \right)^{-\frac{3}{2}} \right) \quad (20)$$

According to the parameter-setting strategies in [26], Scheme 1 is applied:  $\mu_n = \mu_0$ ;  $\lambda_n$  is chosen by cross-validation. Because the sparsity of the Chinese chess situation feature data set is unknown. If the Chinese chess situation evaluation in this paper is a  $k$ -sparsity problem,  $\lambda_n$  can be derived by the method provided in section III-A in [26]. But the unknown nature of sparsity causes the cross-validation of  $\lambda_n$ , which is barely satisfactory. Here

$$\mu_0 = \frac{(1-\varepsilon)}{\|H\|^2} \text{ with any small } \varepsilon \in (0, 1) \quad (21)$$

The target output matrix of AE is equal to the input matrix  $X$ . So  $\varepsilon$  is equal to 0. Thus,  $\mu_0 = \frac{1}{\|H\|^2}$  where  $\|H\|^2 = \max(\text{eig}(H^T H))$ .

Begin the iteration by taking  $\beta_0 = 0$  and compute the hidden layer weights  $\beta$  iteratively utilizing (17), we could perfectly extract the representative and compact data set features. Moreover, compared with the traditional AE with least square method, the  $L_{1/2}$  regularization optimization is a better solution for the feature extraction and dimension transformation.

## V PERFORMANCE EVALUATION AND ANALYSIS

The experiments are conducted to evaluate the Chinese chess situation predicting performance of DSWAN which is compared to the basic SWAN algorithm. The experiments platform

configurations are all listed as follows: Desktop PC, Intel-i5 3.30G CPU, 4G DDR3L RAM, MATLAB R2013a.

#### A. Chinese Chess Data Set Establishment

Chinese chess game is a new field in Machine Learning. The data set of Chinese chess to machine game research job is still a gap. So, the establishment of Chinese chess data set is essential and meaningful. All of the Chinese chess manuals are collected from the Internet and Chinese chess books. Each sample is extracted from one Chinese chess situation which is the middlegame of a match using the method proposed in Section II. In a manual, only one middlegame situation can be chosen to extract features. The total sample size of data set is 50000. The victory and defeat of Chinese chess game should be balanced, so the samples' number whose class attribute are 1 or 2 are identical to avoid the imbalance problem in machine learning.

#### B. Comparison Between DSWAN and SWAN

In this section, we will evaluate the classification performance of DSWAN and SWAN on the Chinese chess situation data set. The major difference of DSWAN and basic SWAN is that before the SWAN-based classification step, DSWAN utilizes a multi-layer AE to gain the sparse representation of the input data. But, in the basic SWAN, the input data is used for classification directly. The hidden neurons in DSWAN and basic SWAN are both activated by the sigmoid function. All the experiments' results are obtained via 10-fold cross validation.

From Section III (A), the only specified parameter in the basic SWAN algorithm is the number of hidden neurons  $L$  which is set as  $\{100, 200, \dots, 4000\}$ .

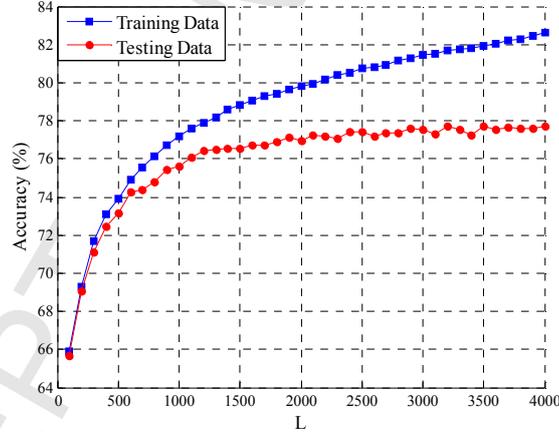


Fig. 2. Training and testing accuracy for the basic SWAN algorithm

Fig. 2 shows the training and testing accuracy of the basic SWAN applied in the  $L$  hidden neurons SLFNs. It can be seen that although training accuracy is increasing with the additive hidden neurons, the testing accuracy has nearly arrived the maximum value when the  $L$  is bigger than 2000.

To fully realize the  $L_{1/2}$  sparsity-constrained AE's potential capability in feature extraction, the parameter  $\lambda_n$  in Section IV (B) should be determined by cross validation. For the exploration of the best  $\lambda_n$ ,  $\lambda_n$  is set as  $\{10^{-10}, 10^{-9}, \dots, 10^9, 10^{10}\}$ . For eliminating the impact of classification step in DSWAN, the neuron number of the last hidden layer is fixed that  $l_3 = 1200$ . And the number of neurons of the former two hidden layers is equal that  $l_1 = l_2$  belong to

{100, 500, 1000, 1500, 2000}. The accuracy of DSWAN is showed in Fig. 3. It can be vividly illustrated that no matter which  $l_1(l_2)$  is chosen, the DSWAN can reach its peak testing accuracy when  $\lambda_n = 10^2$ . Hereinafter the experiments will all apply this  $\lambda_n$ .

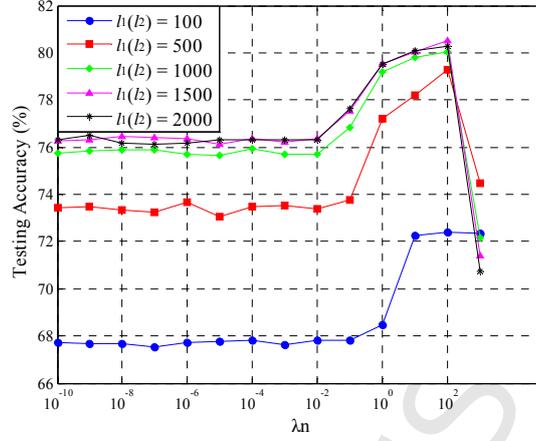


Fig. 3. Accuracy of DSWAN in terms of  $\lambda_n$

For the experiments conducted utilizing DSWAN, the hidden neurons number  $L$  is also a vital parameter that needs to be fixed. Aiming at maximizing the performance of DSWAN, before the evaluation of  $L$ , the relationship of the hidden neurons number of different hidden layers should be sought out. From Section IV (A), the former two layers of DSWAN is a multi-layer AE as a whole, whose two hidden layers have identical neurons. Then the relationship of the last one hidden layer and the former two hidden layers of DSWAN should be estimated. Suppose that the number of neurons of the former two hidden layers is equal that  $l_1 = l_2$  is the element of {100, 200, ..., 3000} and the last one layer  $l_3$  is set as {500, 1200, 2000}. The training and testing accuracy are generated on the Chinese chess situation data set respectively.

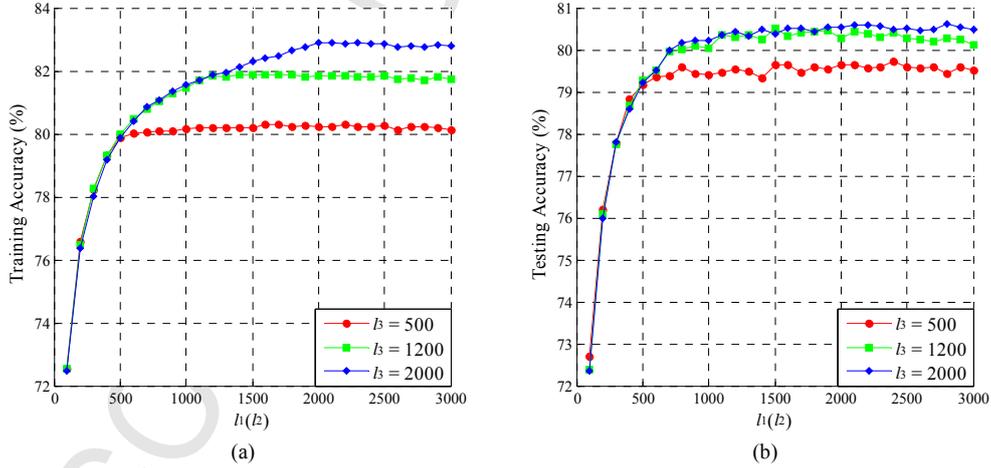
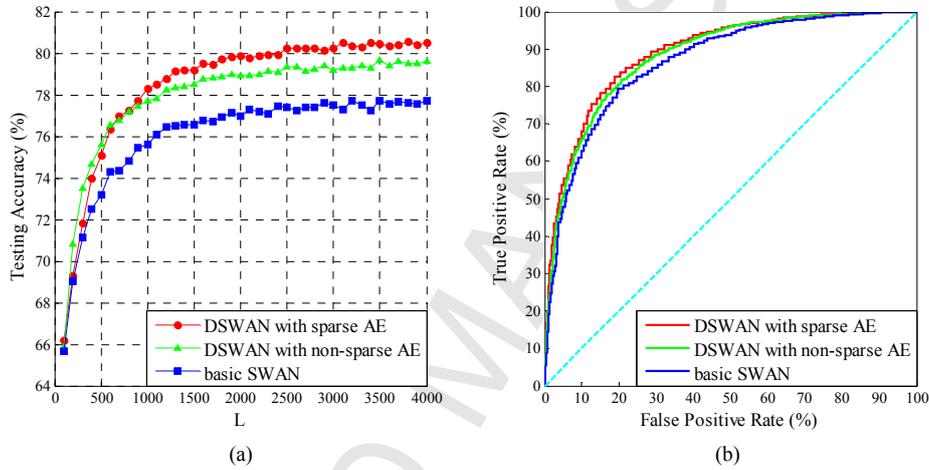


Fig. 4. Accuracy of DSWAN in terms of  $l_1$ ,  $l_2$  and  $l_3$ . (a) Training accuracy of DSWAN. (b) Testing accuracy of DSWAN.

Fig. 4(a) and (b) show the training accuracy and testing accuracy of DSWAN with different hidden layer arrangements respectively. Because the number of hidden neurons of  $l_1$  and  $l_2$  is equal, their numbers are both represented by  $l_1(l_2)$  below. Moreover,  $l_3$  is represented the number of hidden neurons of final hidden layer. As  $l_1(l_2)$  increases, the training accuracy of DSWAN changes before the occasion that  $l_1(l_2) = l_3$ . When  $l_1(l_2) > l_3$ , the training accuracy is approximately equal to the accuracy when  $l_1(l_2) = l_3$ . As for the testing accuracy of DSWAN, the accuracy curves with  $l_3 = 1200$  and  $l_3 = 2000$  respectively nearly coincide. So, testing accuracy of DSWAN has reached the convergence to the best performance before  $l_1(l_2)$  increased to 1200. On the contrary, testing accuracy curve of DSWAN with  $l_3 = 500$  is lower than the former two curves when  $l_1(l_2) > l_3$ . Apparently, the DSWAN has not achieved the optimal effect even when  $l_1(l_2) > l_3$ . The conclusion is that in DSWAN, when  $l_3$  is fixed, the structure that  $l_1(l_2)$  equals to  $l_3$  is the optimum choice for evaluating the best performance. Then in next experiments, the structures of DSWAN are all designed to contain the same size



hidden layers.

Fig. 5. Performance comparison of DSWAN with sparse AE, DSWAN with non-sparse AE, and basic SWAN. (a) Testing accuracy in terms of L. (b) ROC curve.

For comparing and contrasting with the basic SWAN, the total neurons of three hidden layers of DSWAN is set the same number as the basic SWAN. Fig. 5(a) and (b) depict the performance of DSWAN with sparse AE, DSWAN with non-sparse AE, and basic SWAN. From picture (a), the testing accuracy comparison of these three models is exhibited. From the testing accuracy comparison of these three models, we can conclude that DSWAN's classification capacity is better than the basic SWAN. The feature extraction function of multi-layer AE can really promote the classifier's work. But the DSWAN doesn't escape the overfitting problem that the testing accuracy increases slower with more hidden neurons. The  $L_{1/2}$  regularization added in the multi-layer AE can release the overfitting problem to a certain degree. When the number of hidden neurons is no more than 600, the testing accuracy of the sparse DSWAN is lower than the non-sparse DSWAN. But as the hidden neurons number increases, the testing accuracy of the sparse DSWAN is higher than the non-sparse DSWAN.

Referring to picture (b) in Fig. 5, a receiver operating characteristic (ROC) curve of these three models are painted. ROC curve represents an evaluation criterion of a binary classifier system as its discrimination threshold is varied. The ideal prediction classifier would generate a point in the upper left corner of the ROC picture which denotes all the samples are classified correctly. The

smaller distance between ROC curve and the upper left corner of the picture is, the better performance the classifier has. As we can see, the curve of DSWAN is closer to the upper left corner of ROC picture than the curve of basic SWAN. And the curve of sparse DSWAN is on the top left of the curve of non-sparse DSWAN. So, DSWAN can perform better than basic SWAN on Chinese chess situation data set. And the sparsity from  $L_{1/2}$  regularization is positive to the classifier.

## VI CONCLUSION

In this paper, we have proposed a set of Chinese chess situation features which can represent the advantages or disadvantages of chess situation to some extent. The Chinese chess situation data set has been founded utilizing these features. In experiments, it has been proved that the Chinese chess situation data set is representative and it fills the gap of lacking the Chinese chess relevant data set. A neoteric MLP which contains  $L_{1/2}$  sparsity-constrained AE and SWAN classifier is proposed. DSWAN can achieve the feature extraction and classification on the Chinese chess situation data set. The dimension of Chinese chess situation data set is so high that overfitting problem arises during the learning process.  $L_{1/2}$  regularization added in the AE can make mapped features of Chinese chess situation data set sparser. The dimension can be effectively reduced before the SWAN classification phase and overfitting problem is remitted.

## ACKNOWLEDGEMENT

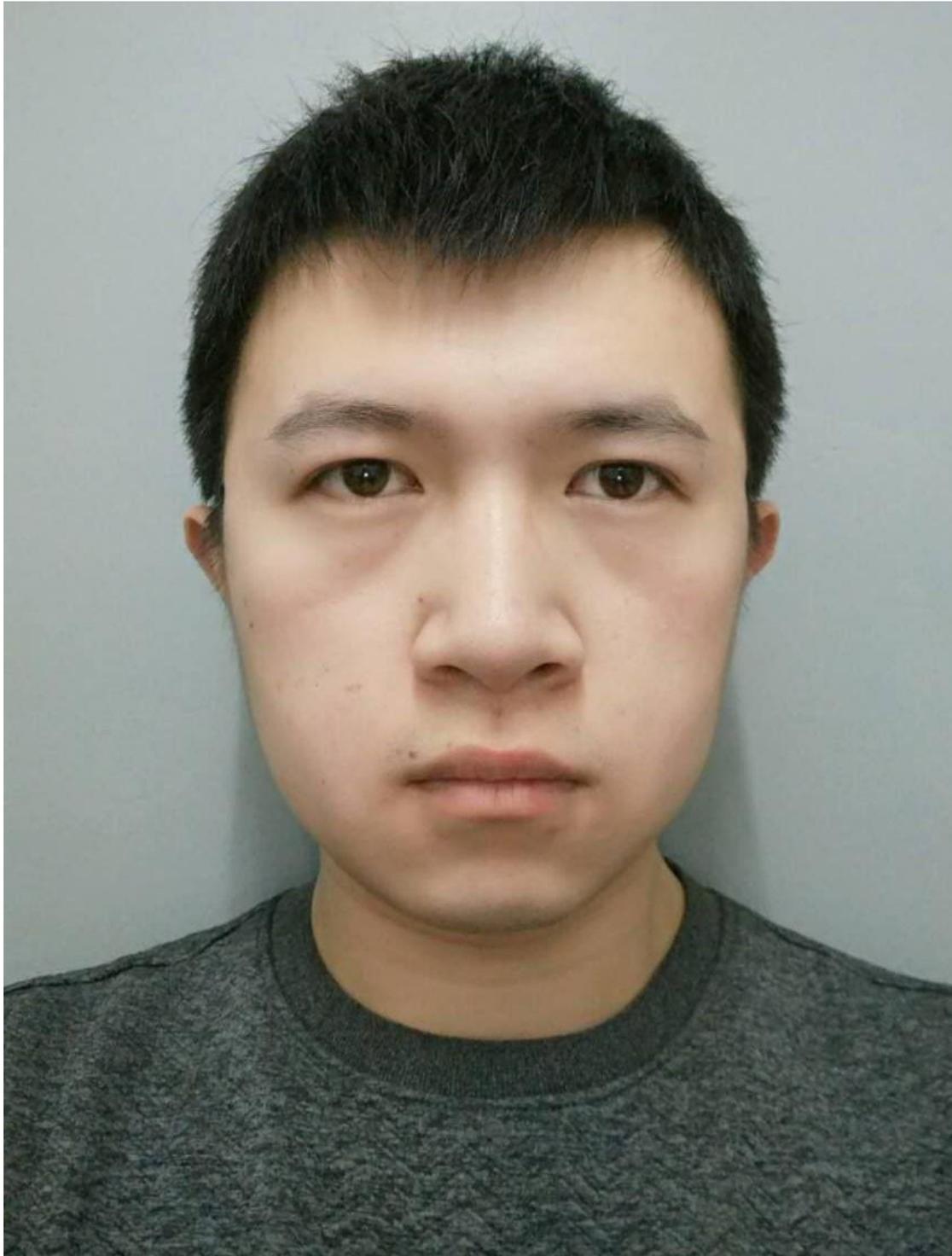
This work is supported by Basic Research Project of Knowledge Innovation Program in Shenzhen (JCYJ20150324140036825), and National Natural Science Foundation of China (61473194 and 71371063).

## REFERENCES

- [1] Murray Campbell, A. Joseph Hoane Jr, Feng-hsiung Hsu; Deep blue; Artificial Intelligence; Volume 134, Issue 1-2, January 2002, Pages 57-83.
- [2] David Silver, Aja Huang, Chris J. Maddison; Mastering the game of Go with deep neural networks and tree search; Nature; Volume 529, January 2016, Pages 484-489.
- [3] Shunqin Xu, Donghui Huang; Design of Chinese Chess Knowledge Base; Proceedings of the National Computer Conference in 1985; 1985, Pages 505-509.
- [4] Boris Stilman; Proximity reasoning for discoveries; International Journal of Machine Learning and Cybernetics; Volume 7, Issue 1, February 2016, Pages 53-84.
- [5] Xinhe Xu, Jiao Wang; Key technologies analysis of Chinese chess computer game; Journal of Chinese Computer Systems; Volume 27, Issue 6, June 2006, Pages 961-969.
- [6] Jinpeng Yue, Su Feng; Application of game tree search algorithm in computer Chinese chess;

- Computer Systems & Applications; Volume 18, Issue 9, January 2009, Pages 140-143.
- [7] Shen Cai; Improved pruning strategy of Chinese chess machine game; Foreign Electronic Measurement Technology; Volume 35, Issue 3, March 2016, Pages 47-49.
- [8] Shuying Liu, Yuanbiao Mu, Hong Li; Game design of Chinese chess based on alpha-beta pruning search algorithm; Information & Communications; Issue 8, 2015, Pages 47-48.
- [9] Qingang Wei, Jiao Wang, Xinhe Xu; A study and design of opening-book of computer chinese chess; Caa Transactions on Intelligent Systems; Volume 2, Issue 1, February 2007, Pages 85-89.
- [10] Qiang Gao, Chen Guo; Technology of hashing and its application research in hybrid game tree search engine of chinese chess; Science Technology & Engineering; Volume 8, Issue 17, 2008, Pages 4869-4872.
- [11] Shangbin Jiao, Ding Liu; Research on translation table heuristic algorithm; Computer Engineering & Applications; Volume 46, Issue 6, 2010, Pages 42-45.
- [12] Shouyu Qian, Minfang Ge; The Computer Aid Analysis for Game of Chinese Chess; Microelectronics & Computer; Issue 5, 1987, Pages 39-42.
- [13] Guangbin Huang, Qinyu Zhu, Chee-Kheong Siew; Extreme learning machine: a new learning scheme of feedforward neural networks; Proceedings of International Joint Conference on Neural Networks; Volume 2, 2004, Pages 985-990.
- [14] Guangbin Huang, Qinyu Zhu, Chee-Kheong Siew; Extreme learning machine: Theory and applications; Neuralcomputing; Volume 70, Issue 1, May 2006, Pages 489-501.
- [15] Guangbin Huang, H.A. Babri; Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions; IEEE Transactions on Neural Networks; Volume 9, Issue 1, January 1998, Pages 224-229.
- [16] Guangbin Huang; Learning capability and storage capacity of two-hidden-layer feedforward networks; IEEE Transactions on Neural Networks; Volume 14, Issue 2, March 2003, Pages 274-281
- [17] D.E. Rumelhart, G.E. Hinton, R.J. Williams; Learning Internal Representation by Back-Propagation Errors; Nature; Volume 323, Issue 323, 1986, Pages 533-536.
- [18] Yu-Shan Hsu, Sin-Jin Lin; An emerging hybrid mechanism for information disclosure forecasting; International Journal of Machine Learning and Cybernetics; Volume 7, Issue 6, December 2016, Pages 943-952.
- [19] S. Balasundaram, Deepak Gupta; On optimization based extreme learning machine in primal for regression and classification by functional iterative method; International Journal of Machine Learning and Cybernetics; Volume 7, Issue 5, October 2016, Pages 707-728.
- [20] Peng Liu, Yihua Huang, Lei Meng, Siyuan Gong; Two-stage extreme learning machine for high-dimensional data; International Journal of Machine Learning and Cybernetics; Volume 7, Issue 5, October 2016, Pages 765-772.
- [21] Jian Zhang, Shifei Ding, Nan Zhang; Incremental extreme learning machine based on deep feature embedded; International Journal of Machine Learning and Cybernetics; Volume 7, Issue 1, February 2016, Pages 111-120.
- [22] Nasser L. Azad, Ahmad Mozaffari; An optimal learning-based controller derived from Hamiltonian function combined with a cellular searching strategy for automotive coldstart emissions; International Journal of Machine Learning and Cybernetics; Volume 8, Issue 3, June 2017, Pages 955-979.

- [23] Yoshua Bengio; Learning Deep Architectures for AI; Foundations & Trends® in Machine Learning; Volume 2, Issue 1, 2009, Pages 1-127.
- [24] Shifei Ding, Nan Zhang, Jian Zhang; Unsupervised extreme learning machine with representational features; International Journal of Machine Learning and Cybernetics; Volume 8, Issue 2, April 2017, Pages 587-595.
- [25] Zongben Xu, Hai Zhang, Yao Wang;  $L_{1/2}$  regularization; Science China Information Sciences; Volume 53, Issue 6, June 2010, Pages 1159-1169.
- [26] Zongben Xu, Xiangyu Chang, Fengmin Xu;  $L_{1/2}$  regularization: a thresholding representation theory and a fast solver; IEEE Transactions on Neural Networks & Learning Systems; Volume 23, Issue 7, July 2012, Pages 1013-1027.
- [27] Iqbal Nouyed, Bruce Poon, M. Ashraful Amin; A study on the discriminating characteristics of Gabor phase-face and an improved method for face recognition; International Journal of Machine Learning and Cybernetics; Volume 7, Issue 6, December 2016, Pages 1115-1130.
- [28] Rita Rana Chhikara, Prabha Sharma; A hybrid feature selection approach based on improved PSO and filter approaches for image steganalysis; International Journal of Machine Learning and Cybernetics; Volume 7, Issue 6, December 2016, Pages 1195-1206.
- [29] Xiao-yun Sun, Feng-ning Kang, Ming-ming Wang; Improved probabilistic neural network PNN and its application to defect recognition in rock bolts; International Journal of Machine Learning and Cybernetics; Volume 7, Issue 5, October 2016, Pages 909-919.
- [30] Yu Cheng, Zhigang Jin, Hongcai Chen; A fast and robust face recognition approach combining Gabor learned dictionaries and collaborative representation; International Journal of Machine Learning and Cybernetics; Volume 7, Issue 1, February 2016, Pages 47-52.
- [31] K. S. Arun, V. K. Govindan; A context-aware semantic modeling framework for efficient image retrieval; International Journal of Machine Learning and Cybernetics; Volume 8, Issue 4, August 2017, Pages 1259-1285.
- [32] Neha Baranwal, G. C. Nandi; An efficient gesture based humanoid learning using wavelet descriptor and MFCC techniques; International Journal of Machine Learning and Cybernetics; Volume 8, Issue 4, August 2017, Pages 1369-1388.
- [33] Liyanaarachchi Lekamalage Chamara Kasun, Hongming Zhou, Guangbin Huang; Representational learning with extreme learning machine for big data; IEEE Intelligent Systems; Volume 28, Issue 6, December 2013, Pages 31-34.
- [34] Jiexiong Tang, Chenwei Deng, Guangbin Huang; Extreme Learning Machine for Multilayer Perceptron; IEEE Transactions on Neural Networks & Learning Systems; Volume 27, Issue 4, April 2016, Pages 809-821.
- [35] Yang Luo, Benqiang Yang, Lisheng Xu; Segmentation of the left ventricle in cardiac MRI using a hierarchical extreme learning machine model; International Journal of Machine Learning and Cybernetics; DOI 10.1007/s13042-017-0678-4; 2017.
- [36] Zhenni Li, Takafumi Hayashi, Shuxue Ding; Dictionary learning with the  $\ell_{1/2}$ -regularizer and the coherence penalty and its convergence analysis; International Journal of Machine Learning and Cybernetics; DOI 10.1007/s13042-017-0649-9; 2017.



Zhi Wang received the B.Sc. degrees from Hebei University, Baoding, China, in 2014. He is currently a M.Sc. candidate of the School of Computer Science and Technology, Hebei University. His main research interests include machine game and neural networks.



Xi-Zhao Wang received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1998. He is currently a Professor with the College of Computer Science and Software Engineering, Shenzhen University, Guangdong, China. From September 1998 to September 2001, he was a Research Fellow with the Department of Computing, Hong Kong Polytechnic University, Hong Kong. His current research interests include learning from examples with fuzzy representation, fuzzy measures and integrals, neuro fuzzy systems and genetic algorithms, feature extraction, multi-classifier fusion, and applications of machine learning.

1. A feature extraction method of Chinese chess middle game situation is proposed.
2. The deep stochastic weight assignment network for feature extraction and classification is proposed.
3. The auto encoder in DSWAN is constrained by  $L_{1/2}$  regularization to make data set more sparse.
4. The experiments on the middle game data set of Chinese chess situation applying DSWAN is conducted. The performance is remarkable while the work is very challenge.