Imperfect Information Game of Contract Bridge Using Double Dummy Bridge Problem

Dr. Dharmalingam Muthusamy,
Assistant Professor,
Department of Computer Science,
Bharathiar University Arts and Science College,
Modakkurichi, Erode - 638104,
Tamil Nadu, India
Email: emdharma@gmail.com

Abstract- The game contract bridge is one of the most popular card games in the world wide. Bridge game is many interesting aspects such as bidding and playing, winning the number of trick including estimation of human hand strength. The decision made on several stage of the game is purely based on the decision- making that was made on the gradual preceding stage. The imperfect information bridge game is the real spirit of the card game in proceeding further deals. The elman neural network architecture with supervised learning implemented through resilient back-propagation algorithm and back-propagation algorithm to trained data was tested with bamberger point count method and work point count method. The research results reveal that bamberger point count method which was implemented in resilient back-propagation algorithm in elman neural network architecture yields better results than work point count method

Index Terms- **Elman neural network architecture, Resilient back-propagation algorithm, Double dummy bridge problem, Bamberger point count method, Work point count method.**

I. INTRODUCTION

The contract bridge is a trick-taking card game the opposite side first competes in a bidding auction for the right to set up the contract for that deal, the side smart the auction being known as the declaring side. The contract is an exchange of the right to set up which suit is a trump for a responsibility to win at least the number of tricks specified by the highest bid [47]. There are significantly increased for the side that is vulnerable, whether one's side is vulnerable affects its strategy for both bidding and play [2].

The artificial neural networks are based on non-linear activation function approximations which make them suitable for most of the applications especially in the Computational Intelligence (CI) games. There are many feed-forward neural network architectures are available which are trained in bridge game [33,34,16] and have been formalized in the best defense model, which also presents the strongest possible assumptions about the opponent [4,3,5,9]. There are two types of human point count methods known as the Point Count Method (PCM) [22, 36, 17] and the Distributional Point Method (DPM) [42, 36].

The paper dealt with section wise as follows. Section 2 provides the problem description of the game of bridge and definition of double dummy bridge problem. Artificial neural network and elman neural network architecture are discussed in section 3 & 4 respectively. In Section 5 dealt the elman neural network architecture and resilient back propagation algorithm with diagrammatical representation. The process flow and application of point count method in contract bridge discussion in section 6. Implementation of elman neural network architecture 52-26-1 briefly dealt in Section 7. Section 8 reported the experimental results and discussion with defined architecture with sample data. Section 9 concludes the findings of the experiment.

II. PROBLEM DESCRIPTION

The game - playing models are interesting and intelligent area for Computational Intelligence (CI) methods [30,31,18]. The bridge game is a partnership game requiring four players, each player sits opposite to his partner and it is traditional to refer to the players according to their position at the table. Such as North, West, South and East, so North and South are partners playing against East and West. It is played with a standard deck of 52 playing cards, where one of the players deals all of the cards, 13 to each player, in clockwise rotation, establishment with the player to the left of the declares.

In the bridge games, basic representation includes value of each card as (Ace (A), King (K), Queen (Q), Jack (J), 10, 9, 8, 7, 6, 5, 4, 3, 2) and suit as (♠ (Spades), the highest, ♥ (Hearts), ♦ (Diamonds), ♣(Clubs), the lowest). This is assignment of cards into particular hands and into hidden subsets, depending on the bridge game rules and regulations. The double dummy should be arranged neatly, separated into suits and the cards in each suit should be in order of rank and overlapped, with the rank of each card clearly visible [10,15].

A. The game of contract bridge

The contract bridge is four players in two fixed partnerships as pair facing each other [43]. The first player of this group who mentioned the value of the contract becomes the declarer. The declarer's partner is well-known as the dummy [35]. The winner of a trick leads to the subsequently stage and the aim of the declarer is to take at least the number of tricks announced for the duration of the bidding phase [19,32].

B. The bidding and playing phases

During the bidding and playing phases are the rule of biding phase is the classification of trumps and declarer of the contract. The playing phase consists of 13 tricks with each player contributing one card [44]. The side which bids the highest will try to win at least that number of tricks bid, with the specified suit as trumps. There are 5 possible trump suits are spades (♠), hearts (♥), diamonds (♦), clubs (♣) and 'no-trump' which is the term for contracts played without a trump[12].

The bidding phase is a conversation between two cooperating team members against an opposing partnership which aims to decide who will be the declarer. The each partnership uses a traditional bidding system to exchange information and understand the partner's bidding sequence. The bridge player has knowledge of his own hand and interesting aspect of the bidding phase is the cooperation and communication of players in North with South and West with East [8]. The play phase seems to be much less interesting than the bidding phase. The player to the left of the declarer leads to the first trick and may play any card and instantaneously after this opening lead, the dummy's cards are exposed [7,27,45,46].

C. Bamberger Point Count Method

The proposed system, Bamberger Point Count Method (BPCM) is a fashionable and most important human popular system which is used to bid a final contract in bridge game. The bamberger is a point count system that requires 52 points to produce a probable slam on power alone. The bamberger point count system which scores 7 point for Ace, 5 point for King, 3 point for Queen and 1point for a Jack, in which no points are counted for 10 and below.

D. Work Point Count Method

The Work Point Count Method (WPCM) which scores 4 points for Ace, 3 points for King, 2 points for Queen and 1 point for a Jack, in which no points are counted for 10 and below. During the bidding phase of contract bridge, when a team reaches the combined score of 26 points, they should use WPCM for getting final contract and out of thirteen tricks in contract bridge, there is a possibility to make use of eight tricks by using WPCM.

III. THE GAME OF BRIDGE IN ARTIFICIAL NEURAL NETWORK

There is game point of view as an artificial neural network, since many of factors in the game of bridge. In this satisfy the definition of it such as the inaccurate data, imperfect information game of bridge through which the fact is to ascertained etc. The two completely different phases of the game of bridge - the bidding and playing are both should be played optimally well-known to gain the best possible result can be matched with that of the training and testing phases. [23,24,29].

IV. THE ELMAN NEURAL NETWORK ARCHITECTURE

The elman neural network architecture is similar to that of the feed-forward neural network architecture shown in Fig.1. This is supervised learning is used in elman neural network architecture, which is used to solve the double dummy bridge problem in contract bridge.

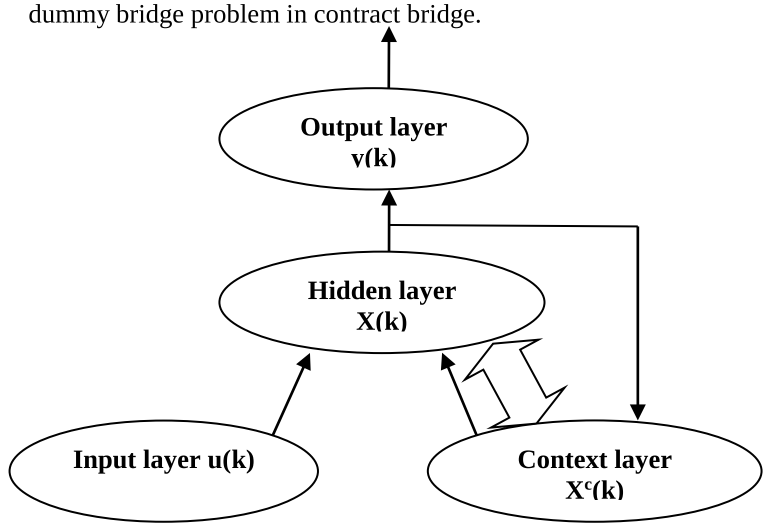

Fig.1 The architecture of the elman neural network

It is a special kind of feed-forward neural network in elman architecture, which has extra local memory neurons and feedback loop [20]. The elman neural network is capable of approximating a nonlinear system without an explicit physical model. A elman neural network has four kinds of layers input layer, hidden layer, context layer and output layer [21]. The context layer is utilized to constitute the back-forward loop, from which the hidden layer selects input. In comparison with other forms of feed forward neural network, the elman neural network is sensitive to history of input data by this mechanism.

The advantage of elman neural network architecture is feedback path which allows the elman neural network to recognize and generate temporal patterns and spatial patterns. After training, interrelations between the current input and internal states are processed and produced the output. However, since the elman neural network usually uses the resilient back-propagation based algorithm to deal with the various signals [6,26,28]. At the same time, the efficiency of the elman neural network is limited to high order system due to the sufficient memory capacity when resilient back-propagation algorithm is employed.

The elman neural network architecture can be trained with gradient descent back-propagation and optimization methods, similar to feed-forward neural network. In addition to this, long training sessions are often required in order to find an acceptable weight solution because of the well known difficulties inherent in gradient descent optimization [13,25].

A. The work flow of elman neural network

The elman network architecture as a consist of input layers, hidden layers, output layer and there are also context layers. The input and output layers interact with the outside environment, where as the hidden and context layers do not. The input layer is only a buffer layer which passes the signals without changing them. The hidden layers can have non-linear activation functions. The context layers are used only to memorize the previous activations of the hidden layers and can be considered to function as one-step time delays. The feed forward network connections are adjustable; the recurrent connections are permanent and because the recurrent connections are permanent [38].

At a specific time 'k', the previous activations of the hidden layer (at time '*k-1*') and the current input (at time 'k') are used as inputs to the network. At this stage, the network acts as a feed-forward neural network and propagates these inputs forward to produce the output. The standard back-propagation learning rule [37] can then be employed to train the network. After this training step the activations of the hidden layers at time 'k' are sent back through the recurrent links to the context layers and saved there for the next training step (time '$k+1$'). The beginning of the training process the activations of the hidden layers are unknown. Usually, they are set to one-half of their maximum range. For a sigmoidal activation function the initial values can be set to 0.5. For a hyperbolic tangent activation function they can be equated to 0.0.

The external input to the network is represented by '$u(k)$' and the network output '$y(k)$'. The total input to the 'i^{th}' hidden layer is denoted as '$v_i(k)$'. The output of the 'i^{th}' hidden layer is denoted as '$x_i(k)$'. The output of the 'j^{th}' context layer is '$x_j^c(k)$'. The following equations hold:

$$v_i(k) = \sum_{j=1}^{n} w_{i,j}^{x}\,(k-1)x_j^c(k) + w_i^u(k-1)u(k) \qquad (1)$$

$$x_i(k) = f(v_i) \qquad (2)$$

$$x_j^c(k) = x_j(k-1) \qquad (3)$$

$$y(k) = \sum_{i=1}^{n} w_i^{y}\,(k-1)x_i(k) \qquad (4)$$

Where $w_i^i(k), w_{i,j}^x(k)$ and $w_i^y(k), i,j = 1,2,\dots,n,$ are the weight of the links, respectively. This is input layer and the hidden layer, between the context layer and the hidden layer, and between the hidden layer and the output layer. f is a sigmoidal activation function. In particular, if input '$u(k)$' is delayed by one time step before it is sent to the input layer, '$x_j^c(k)$' is replaced by '$x_j(k-1)$', and the hidden layers are assumed to be linear, the above equations become

$$v_i(k) = \sum_{j=1}^{n} w_{i,j}^{x}\,(k-1)x_j(k-1) + w_i^u(k-1)u(k-1) \qquad (5)$$

$$x_i(k) = v_i(k) \qquad (6)$$

$$y(k) = \sum_{i=1}^{n} w_i^{y}\,(k-1)x_i(k) \qquad (7)$$

Eq.(5)-(7) are the state-space description of an 'n^{th}' order linear dynamic system, where 'n' is the dimension of $x(\mathrm{k}) = \{x_i(k)\}$, that is the number of hidden and context layers. The order of the model depends on the number of states, which is also the number of hidden layers. Eq.(5)-(7) can be expanded into the following:

$$\begin{aligned} y(k) = A_1 y(k-1) + A_2 y(k-2) \\ + \cdots + A_n y(k-n) \\ + B_1 u(k-1) + B_2 u(k-2) + \cdots \\ + B_n u(k-n) \end{aligned} \qquad (8)$$

An elman neural network is able the model an 'n^{th}' -order dynamic system if it can be trained to do so. In this model is a system represented eq. (8) using input- output data, '$2n$' input layers would be needed if a feed-forward neural network architecture is used. For an elman neural network, the input layer number is one or '$n+1$' if the context layers are regarded as input layers. An elman neural network will be significantly smaller in structure than a feed-forward neural network architecture when 'n' is large.

V. RESILIENT BACK-PROPAGATION TRAINING ALGORITHM WORKING IN ELMAN ARCHITECTURE

The resilient back-propagation algorithm is a local adaptive learning scheme and improving the performing of supervised learning in elman neural network architecture. The basic principle of resilient back-propagation algorithm is to eliminate the harmful influence of the size of the partial derivative on the weight step [40,41]. Only the sign of the derivative is considered to indicate the direction of the weight update. The eq. (9) for each weight, if there was a sign change of the partial derivative of the total error function compared to the last iteration, the update value for that weight is multiplied by a factor η−, where 0 <η− < 1. The update values are calculated for each weight in the above manner in finally each weight is changed by its own update value, in the opposite direction of that weight's partial derivative. This is to minimize the total error function.

The introducing for each weight w_{ij} its individual updates value Δ_{ij} (t), which exclusively determines the magnitude of the weight-update. This update value can be expressed mathematically according to the learning rule for each case based on the observed behavior of the partial derivative during two successive weight-steps by the following formula:

$$\Delta_{ij}(t) = \begin{cases} \eta^{+}.\Delta_{ij}(t-1), & if \quad \frac{\partial E}{\partial w_{ij}}(t).\frac{\partial E}{\partial w_{ij}}(t-1) > 0 \\ \eta^{-}.\Delta_{ij}(t-1), & if \quad \frac{\partial E}{\partial w_{ij}}(t).\frac{\partial E}{\partial w_{ij}}(t-1) < 0 \\ \Delta_{ij}(t-1), & else \end{cases} \tag{9}$$

Where 0<η^{-} <1<η^{+}.

A clarification of the adaptation rule based on the above formula can be stated. The eq. (10) it is evident that whenever the partial derivative of the equivalent weight w_{ij} varies its sign, which indicates that the last update was large in magnitude and the algorithm has skipped over a local minima, the update-value Δ_{ij} (t) is decreased by the factor η−. The eq. (11) that is if the derivative is positive, the weight is decreased by its update value, if the derivative is negative, the update-value is added.

$$\Delta w_{ij}(t) = \begin{cases} -\Delta_{ij}(t), & if \quad \frac{\partial E}{\partial w_{ij}}(t) > 0 \\ \Delta_{ij}(t), & if \quad \frac{\partial E}{\partial w_{ij}}(t) < 0 \\ 0, & else \end{cases} \tag{10}$$

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) \tag{11}$$

However, there is one exception. The eq. (12) if the partial derivative changes sign that is the previous step was too large and the minimum was missed, the previous weight-update is reverted

$$\Delta w_{ij}(t) = -w_{ij}(t-1),$$

$$if \ \frac{\partial E}{\partial w_{ij}}(t).\frac{\partial E}{\partial w_{ij}}(t-1) < 0 \tag{12}$$

due to that backtracking weight-step, the derivative is assumed to change its sign once again in the following step. In order to avoid a double penalty of the update-value, there should be no adaptation of the update-value in the succeeding step. In practice this can be done by setting $\frac{\partial E}{\partial w_{ij}}(t-1) = 0$ in the Δ_{ij} update-rule above.

The eq. (13) partial derivative of the total error is given by the following formula:

$$\frac{\partial E}{\partial w_{ij}}(t) = \frac{1}{2}\sum_{p=1}^{p}\frac{\partial E_p}{\partial w_{ij}}(t) \tag{13}$$

Hence, the partial derivatives of the errors must be accumulated for all training patterns. This indicates that the weights are updated only after the presentation of all of the training patterns [39,11,14].

VI. PROCESS FLOW OF POINT COUNT METHODS

In bridge games, though the basic demonstration includes value of each card as (Ace (A), King (K), Queen (Q), Jack (J), 10, 9, 8, 7, 6, 5, 4, 3, 2) for assignment of cards into selective hands. A uniform linear transformation in the range 0.10 through 0.90 where 0.10 is assigned to the smallest card value 2 with an increment of 0.067 to the next card value i.e., 3 and so on till 0.90 for the highest card value A is assigned. The suit cards such as (♠ (Spades), the highest, ♥ (Hearts), ♦ (Diamonds), ♣(Clubs), the lowest) are assigned a real number using the following mapping: Spades (0.3), Hearts (0.5), Diamonds (0.7) and Clubs (0.9). There are 52 input values and each value represents one card from the deck and the positions of cards in the input layer are fixed.

The human estimators of hand strength can be divided into two categories such as point count methods and distributional point methods. The human point count methods [1] are based on calculating the strength of a hand as a sum of single cards strength and the value of each card depends only on card's rank. The other category of human hand's strength estimators contains distributional points, in which the patterns are scored based on its existence in a set of cards assigned to one hand. The most important patterns are suit lengths and existence of groups of honors in one suit.

VII. IMPLEMENTATION OF 52-26-1 ELMAN NEURAL NETWORK ARCHITECTURE

In this paper, mainly focus on the elman neural network architecture with 52, (13x4) input neurons for solving the DDBP is attempted and the results are discussed. The 52 input card representation deals are implemented in the elman neural network architecture as shown in Fig.2 and Fig.3.

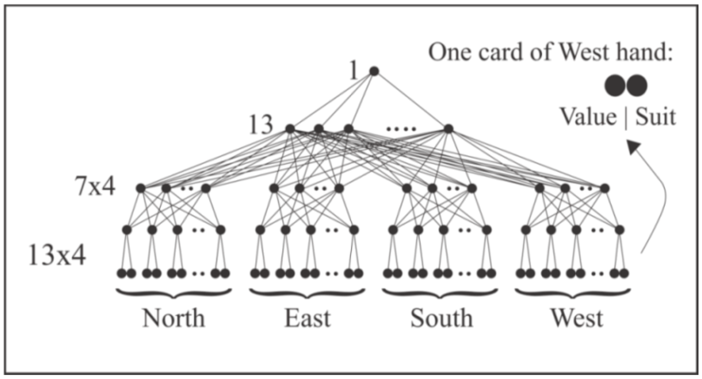

Fig. 2 The elman architecture with 13x4 input pattern

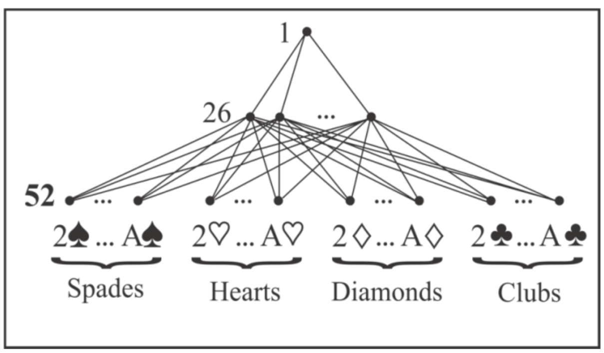

Fig. 3 elman architecture with 52-26-1 input patterns.

Layers are fully associated, i.e., in the 52- 26-1 network 52 input neurons are associated to all 26 hidden neurons and all hidden neurons are associated to a single output neuron. The number of hidden neurons to a conscientious problem is still decided by a rule of thumb. The number of neurons is minimum, the model may take too much of time to learn or may not be able to learn at all resulting in an underprivileged performance during the training session. The number of neurons in the hidden layer is equivalent to the input neurons, then the aim of the training phase itself may become obsolete and instead of learning during training session. The neural network might memorize the patterns, which will result, very badly in the testing phase of the network. Thus, it is decided to have half the size of the input neurons as a rule of thumb and in the implementation phase after a trial with 25 neurons, 26 neurons, 27 neurons, it is concluded to stick with 26 neurons since it is half the size of the input neurons. For training and learning the data, two activation functions viz., log sigmoid transfer function and hyperbolic tangent sigmoid functions are used. The resilient back propagation algorithm is used for training and testing through MATLAB 2015a.

VIII. EXPERIMENTAL RESULTS AND DISCUSSION

A total number of five thousand deals from the GIB library for training and two thousand five hundred among the trained data were used for testing on elman neural network architecture with fifty two input neurons, twenty six hidden neurons and one output neuron (52-26-1). There are 20 numbers for each deal i.e. 5 trump suits confidential as no-trumps, spades, hearts, diamonds and clubs by 4 sides. The mean squared error during the training phase and testing phase using log sigmoid as the activation function is illustrated in Fig.4, while the mean squared error during the training and testing phases using hyperbolic sigmoid as the activation function is illustrated in Fig.5.

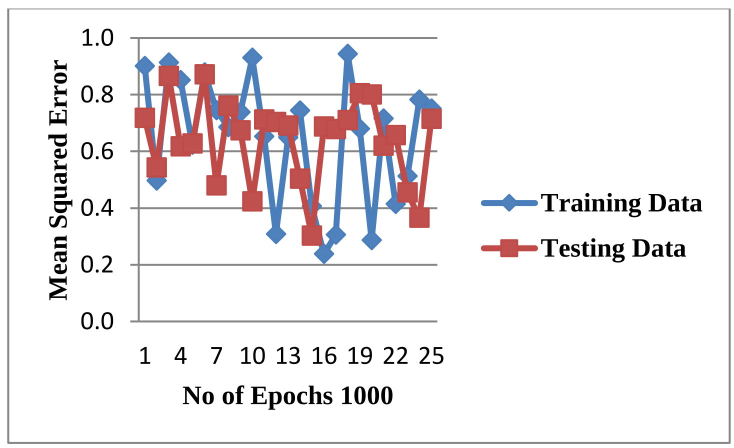

Fig.4 Mean Squared Error (MSE) during training and testing phase of log sigmoid function.

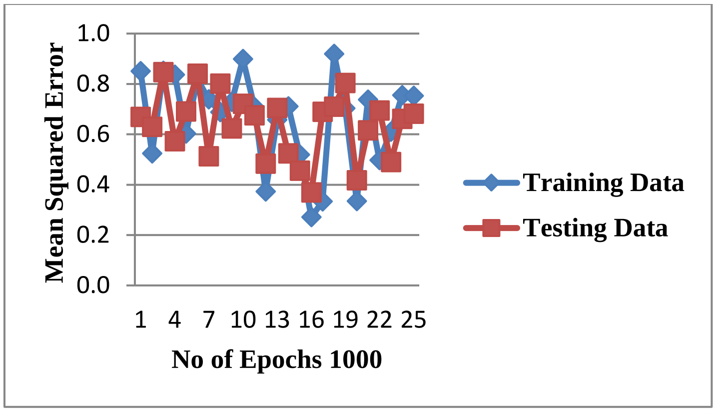

Fig.5 Mean Squared Error (MSE) during training and testing phase of hyperbolic sigmoid function.

In the elman neural network architecture, BPCM and WPCM are used as contract bridge game separately with resilient back- propagation algorithm, compared with each other and inferred that the BPCM produced better results when compared to WPCM. The results revealed that, the data tested through elman neural network architecture show better performance and the time taken for training and testing are relatively minimum which is converging towards the possible minimum error during the iterations.

A. Comparison of BPCM and WPCM

The sample deal representation by adding human estimation didn't improve the best overall result accomplished by pure 52-26-1 in the case of BPCM and only slight improvement was notified in the case of WPCM contracts. This observation suggested that the relevance of additional information related to suit lengths and point distribution in particular hands has been autonomously discovered by the best 52-26-1 elman neural network architecture during the training process. The human players are visibly better at solving the no-trump contracts than the suit ones and the opposite conclusion is also valid in the case of neural networks. The elman neural network architecture can be trained to capture the implicit reasoning used for bidding a hand in bridge.

Thus to validate the convergence of the algorithm in the elman neural network architecture with 52 (13x4) input neurons for solving the DDBP, the problem is attempted with resilient back-propagation algorithm. The performance during training and testing phases of the sample deal of BPCM and WPCM of the same data from GIB library used in this architecture is illustrated for the purpose of comparison in Fig 6.

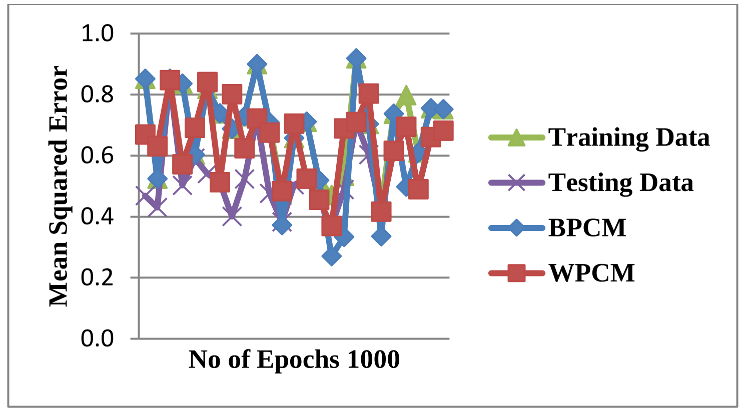

Fig. 6 Mean Squared Error during training and testing phase of BPCM and WPCM.

The result shows that, minimum eight tricks and maximum 12 tricks can be taken using WPCM during final bid of contract bridge. In BPCM minimum eight and maximum 13 tricks can be taken in final bid of contract bridge. Even though there are various point count methods are available, we have compared WPCM and BPCM in this paper. Among these two methods BPCM is the best method for getting maximum number of tricks (i.e. Slam (12) or Grand Slam (13)) in final bid of contract bridge.

IX. CONCLUSION

The elman neural network architecture used in contract bridge, the BPCM and WPCM are compared with each other during training process. The numbers of hidden nodes used are 25, 26 and 27, in elman neural network architecture. Among these, the 26 hidden neurons produced better results, which were compared with other two hidden neurons 25 and 27. The human point count methods, BPCM and WPCM are used to take maximum number of tricks in contract bridge. The result reported that, both the methods BPCM and WPCM minimized the Mean Square Error (MSE), reduced the time taken for playing and increase the number of tricks taken in DDBP. The result accomplished that elman neural network architecture with BPCM yields better results than WPCM.

The problem of representative a particular acquaintance gained through the learning process is extremely specialized and it is inferred that the estimated method bamberger point count method. In BPCM method that provides some new ideas to the bridge players. It also encourages basic and semi professional players as well as improving their bridge skills. Furthermore the bamberger point count method can be comprehensive to be taking into consideration of different human point count methods in contract bridge using different architectures and algorithms to solve DDBP more professionally and effectively.

REFERENCES

[1] Amalraj R, Dharmalingam M (2015) A work point count method coupled with back-propagation for solving double dummy bridge problem. Neurocomputing, 168:160-178

[2] Amit A, Markovitch S (2006) Learning to bid in bridge. Machine Learning, 63(3), pp 287-327

[3] Ando T, Kobayashi N, Uehara T (2003) Cooperation and competition of agents in the auction of Computer Bridge. Electronics and Communications in Japan, Part 3, 86(12):76-86

[4] Ando T, Sekiya Y, Uehara T (2000) Partnership bidding for Computer Bridge. Systems and Computers in Japan, 31(2), pp 72-82

[5] Ando T, Uehara T (2001) Reasoning by agents in computer bridge bidding. In: Computers and games, vol. 2063. Lecture Notes in Computer Science. Springer, Berlin, pp 346-364

[6] Chen Y., Qi W., and Zhao J.,(2008) A new Elman neural network and its dynamics properties, proceeding of international conference on cybernetics and intelligent system, pp 971- 975

[7] Dharmalingam M, Amalraj R (2013a) Neural Network Architectures for Solving the Double Dummy Bridge Problem in Contract Bridge. In Proceeding of the PSG-ACM National Conference on Intelligent Computing. 1(1), pp 31-37

[8] Dharmalingam M, Amalraj R (2013b) Supervised Learning in Imperfect Information Game. International Journal of Advanced Research in Computer Science, 4(2), pp 195-200

[9] Dharmalingam M, Amalraj R (2014a) A Solution to the Double Dummy Bridge Problem in Contract Bridge Influenced by Supervised learning module Adapted by Artificial Neural Network, ICTACT Journal on Soft Computing: Special Issue on Distributed Intelligent System and Applications, 5(01), pp 836 – 843

[10] Dharmalingam M, Amalraj R(2014b) Artificial Neural Network Architecture for Solving the Double Dummy Bridge Problem in Contract Bridge, International Journal of Advanced Research in Computer and Communication Engineering, 2(12): 4683- 4691

[11] Dharmalingam M, Amalraj R(2014c) Fast Supervised learning Architecture - A work point count system coupled with Resilient Back-propagation Algorithm for Solving the Double Dummy Bridge Problem, International Journal of Emerging Trends & Technology in Computer Science, 3(3):189-195

[12] Dharmalingam M, Amalraj R (2014d) Back-Propagation Neural Network Architecture for Solving the Double Dummy Bridge Problem in Contract Bridge, IEEE International Conference on Intelligent Computing Applications, pp 454-461

[13] Dharmalingam M, Amalraj R (2014e) Supervised Elman Neural Network Architecture for Solving the Double Dummy Bridge Problem in Contract Bridge, International Journal of Science and Research, 3(6):2745-2750

[14] Dharmalingam M, Amalraj R (2015a) Artificial Neural Network Architectures: Work Point Count System Coupled with Back-Propagation and Resilient Back - Propagation Algorithm for Solving Double Dummy Bridge Problem in Contract Bridge, International Journal of Innovative Science, Engineering and Technology, 2(8):142-157

[15] Dharmalingam M, Amalraj R(2015b) Artificial Neural Network Architectures for solving Double Dummy Bridge Problem in Imperfect Information Game, International Journal of Computational Intelligence and Informatics,05(03):188-197

[16] Dharmalingam M.(2015) Feed- Forward Neural Network Architecture for solving Double Dummy Bridge Problem in Contract Bridge. International Journal of Scientific Engineering and Applied Science, 1(6):192-202

[17] Dharmalingam M, (2016) Point Count Systems in Imperfect Information Game, International Research Journal of Engineering and Technology, 03(01):362-370

[18] Dharmalingam M, (2017) Artificial Neural Network Architectures for Solving the Contract Bridge, International Journal of Advanced Information and Communication Technology, 04(03):01-09

[19] Dharmalingam M, (2018) Double Dummy Bridge Problem in Contract Bridge: An Overview, International Journal Artificial Intelligent Systems and Machine Learning, 10(01):01-07

[20] Elman J.,(1990) Finding structure in time, Cognitive Science 14(3), pp 179-211

[21] Elman J, (1991) Distributed representation, simple recurrent neural networks, and grammatical structure, Machine Learning, 7, pp195 - 225

[22] Francis H, Truscott A Francis D (2001) The Official Encyclopedia of Bridge, American Contract Bridge League.(6th) Edition

[23] Frank I, Basin DA (2001) A Theoretical and Empirical Investigation of Search in Imperfect Information Game. Theoretical Computer Science, 252(1), pp 217-256

[24] Ginsberg ML (2001) GIB: Imperfect Information in a Computationally Challenging

Game, Journal of Artificial Intelligence Research, 14, pp 303-358
[25] Haykin S.,(1998) Neural Networks: A Comprehensive Foundation, Prentice - Hall, Englewood Cliffs, 1998.
[26] Hertz J., KroghA., and Palmer R.G.,(1991) Recurrent neural networks, Introduction to the Theory of Neural Computation
[27] Jamroga W (1999) Modeling Artificial Intelligence on a case of bridge card play bidding. In Proceedings of International Workshop on Intelligent Information System. pp 276-277
[28] Kremer S.C (1995) On the computational power of Elman-style recurrent neural networks, IEEE Tran. on Neural Networks, 6, pp 1000 – 1004
[29] Levy DN (1998) The million pound bridge program, In Heuristic programming in Artificial Intelligence, pp 95 - 103
[30] Mandziuk J (2007) Computational Intelligence in Mind Games. Springer
[31] Mandziuk J (2008) Some thoughts on using Computational Intelligence methods in classical mind board games, International Proceedings of the International Joint Conference on Neural Networks, Hong Kong, pp 4001 - 4007
[32] Mandziuk J (2010) Knowledge-free and Learning – Based methods in Intelligent Game Playing. Springer
[33] Mandziuk J, Mossakowski K (2007) Example - based estimation of hands strength in the game of bridge with or without using explicit human knowledge, In Proc. IEEE Symp, Computational Intelligence, Data Mining, pp 413 - 420
[34] Mandziuk J, Mossakowski K (2009) Neural Networks compete with expert human players in solving the double dummy bridge problem. In Proceedings of the Conference on Computational Intelligence and games, (5), pp 117-124
[35] Mossakowski K, Mandziuk J (2004) Artificial Neural Networks for solving double dummy bridge problems, 3070, pp 915-921
[36] Mossakowski K, Mandziuk J (2009) Learning without human expertise: A case study of Double Dummy Bridge Problem. IEEE Transactions on Neural Networks, 20(2), pp 278-299
[37] Pham D.T., and Liu X.,(1992) Dynamic system modeling using partially recurrent neural networks, Journal of Systems Engineering, 2, pp 90-97
[38] Pineda F.J.,(1987) Generalization of back-propagation to recurrent neural networks, Phys. Rev. Lett, 59, pp 2229 - 2232
[39] Riedmiller M., and Braun H.,(1992) Rprop-a fast adaptive learning algorithm, in International Symposium on Computer and Information Sciences, pp 279-285.
[40] Riedmiller M, Braun H (1993) A direct adaptive method for faster back propagation: the RPROP algorithm. In: Proceedings of the IEEE international conference on neural networks (ICNN 1993), pp 586- 591
[41] Riedmiller M.,(1994) Advanced supervised learning in multi-layer perceptrons- From backpropagation to adaptive learning algorithms, Computer Standards and Interfaces, 16 (5):265-278.
[42] Root HW (1998) The ABCs of Bridge. Three Rivers Press
[43] Sarkar M, Yegnanarayana B, Khemani D (1995) Application of neural network in contract bridge bidding. In Proceeding of National Conference on Neural Networks and Fuzzy System, pp 144-151
[44] Seifert B (1996) Encyclopedia of Bridge, Warsaw, Polish Scientific Publishers PWN
[45] Smith SJJ, Nau DS Throop TA (1998a) Computer Bridge - A Big Win for AI planning. Artificial Intelligence Magazine, 19(2), pp 93-106
[46] Smith SJJ, Nau DS, Throop TA (1998b) Success in Spades: Using AI Planning Techniques to Win the World Championship of Computer Bridge. In Proceeding of the National Conference on Artificial Intelligence, pp 1079-1086
[47] Yegnanarayana B, Khemani D, Sarkar M (1996) Neural Network for contract bridge bidding.21(3), pp 395-413

Author Biography

Dr. Dharmalingam Muthusamy is currently working as an Assistant Professor in the Department of Computer Science in Bharathiar University Arts and Science College at Modakkurichi in Erode, Tamil Nadu, India. He has published more than 20 research papers in reputed international and national journals. His main research work focuses on Neural Networks, Data Mining and Computational Intelligence based games. He has 15 years of teaching experience and 8 years of research experience.