Point Count Systems in Imperfect Information Game

Dr.M Dharmalingam[1]
[1]Associate *Professor,Department of Computer Science*
Nandha Arts and Science College, Erode,Tamil Nadu,India

Abstract:The contract bridge is an intelligent game, which enhances the creativity with various skills and quest to acquire the intricacies of the game, because no player knows exactly what moves other players are capable of during their turn. The Bridge being a game of imperfect information is to be equally well defined, since the outcome at any intermediate phase is purely based on the decision made on the immediate prior stage. One along with the architectures of Artificial Neural Networks (ANN) is applied by training on sample deals and used to estimate the number of tricks to be taken by one pair of bridge players is the key idea behind Double Dummy Bridge Problem (DDBP) implemented with the neural network paradigm. This paper mainly focuses on Cascade-Correlation Neural Network (CCNN) in which is used to solve the bridge problem by using Back-Propagation (BP) algorithm. The proposed systems are Work Point Count System (WPCS) and Bamberger Point Count System (BPCS) are an exclusive, most important and popular systems in which are used to bid a final contract in bridge game.

Key words: *ANN, CCNN, BP algorithm, Contract Bridge, DDBP, Bidding, Playing, WPCS, BPCS.*

1. INTRODUCTION

The bridge is a game which requires some amount of intelligence and it increases the creativity of the human in decision making and there are extremely powerful Artificial Neural Network (ANN) approaches are available in which playing agents are equipped with carefully designed evaluation functions. In the game playing domain, the most popular Computational Intelligence (CI) disciplines are Neural Networks (NN), Evolutionary Methods (EM), and Supervised Learning (SL) [1]. The ANN is a computational structure capable of processing information in order to finish a given task. A Neural Network is composed of many simple neurons each of which receives inputs from selected other neurons, and performs basic operations on these input information and sends its response to other neurons in the network. An ANN models can therefore be regarded as roughly a simplification and abstraction of biological networks. The ANN has been successfully applied to various recognition, classification problems [2] and games [3-5].

Artificial neural networks are classified under a broad spectrum of Artificial Intelligence (AI) that attempts to imitate the way a human brain works and the Cascade-Correlation Neural Network (CCNN) is most common type of neural network in use and these are often trained by the way of supervised learning supported by Back-Propagation (BP) algorithm [6-10] and they have been formalized in a best defense model, which presents the strongest possible assumptions about the opponent. This is used by human players because modeling the strongest possible opponents provides a lower bound on the pay off that can be expected when the opponents are less informed. The new heuristics of beta-reduction and iterative biasing were introduced and represents the first general tree search algorithm capable of consistently performing at and above expert level in actual card play. The effectiveness of these heuristics, particularly when combined with payoff-reduction mini-maxing results in iprm-beta algorithm. The problems from the game of bridge, iprm-beta actually makes less errors than the human experts that produced the model solutions. It thus represents the first general search algorithm capable of consistently performing at and above expert level on a significant aspect of bridge card play [11].

The forward pruning techniques may produce reasonably accurate result in bridge game. Two different kinds of game trees viz., N-Game trees and N-Game like trees were used to inspect, how forward pruning affects the probability of choosing the correct move. The results revealed that, mini-maxing with forward pruning did better than ordinary mini-maxing, in cases where there was a high correlation among the mini-max values of sibling nodes in a game tree. The result suggested that forward pruning may possibly be a viable decision-making technique in bridge games [12].The Bridge Baron is generally acknowledged to be the best available commercial program for the game of contract bridge. The bridge baron program was developed by using domain dependent pattern-matching techniques which has some limitations. Hence there was a need to develop more sophisticated AI techniques to improve the performance of the bridge baron which was supplemented by its previously existing routines for declarer play with routine based on Hierarchical Task-Network (HTN) planning techniques. The HTN planning techniques used to develop game trees in which the number of branches at each node corresponds to the different strategies that a player might pursue rather than the different cards the player might be able to play [13].

The GIB is a production program, expected to play bridge at human speeds. A GIB used Monte Carlo methods exclusively to select an action based on the double dummy analysis. All other competitive bridge-playing programs have switched their card play to similar methods, although GIB's double dummy analysis is substantially faster than most of the other programs and its play are correspondingly stronger. If the bidding simulation indicates that the opponents are about to achieve a result much inferior than what they might achieve if they saw each other's cards, that is evidence that there may be a gap in the database. An unfortunately, it is also evidence that GIB is simply effectively troublesome its opponents efforts to bid accurately. The GIB's bidding is substantially better than that of earlier programs but not yet of expert caliber [14].

Among the various neural networks, in this paper we mainly focus cascade-correlation neural network for training and testing the data. The back-propagation algorithm is used in the network to train the data for solving double dummy bridge problems in contract bridge. A point count method and distributional point methods are the two types of hand strength in human estimators. The structure of this paper is organized as follows. Section 2 and Section 3 gives a brief description of contract bridge game and data representation respectively. Section 4 discuss about briefing artificial neural networks and BP algorithm. Our proposed double dummy bridge problem and problem implementations are discussed in Section 5 and 6. Section 7 gives about the results and discussion with example and Section 8 discussed about the conclusion and future links of our research.

2. The contract bridge game

The contract bridge, usually known simply as bridge, is a trick - taking card game. There are four players in two fixed partnerships (Pairs). Partners sit facing each other. It is established to refer to the players according to their position at the table as North (N), East (E), South (S) and West (W), so N and S are partners playing against E and W. Example shown in Fig. 1.

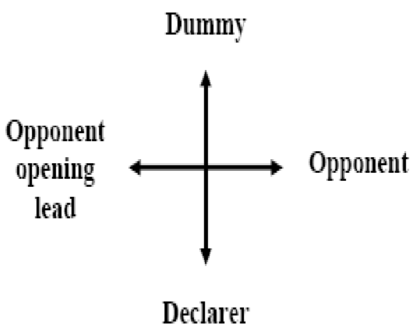

Fig. 1. Game Disposition.

A standard 52 card pack is used. The cards in each suit rank from the highest to the lowest as Ace (A), King (K), Queen (Q), Jack (J), 10, 9, 8, 7, 6, 5, 4, 3, 2. The dealer deals out all the cards one at a time so that each player receives 13 of them. The game then proceeds through a bidding and playing phase. The purpose of the biding phase is to identification of trumps and declarer of the contract. The playing phase consists of 13 tricks, with each player contributing one card to each trick in a clockwise fashion with another level bid to decide who will be the declarer. The side which bids highest will try to win at least that number of tricks bid, with the specified suit as trumps. There are 5 possible trump suits: *spades* (♠), *hearts* (♥), *diamonds* (♦), *clubs* (♣) and "*no-trump*" which is the term for contracts played without a trump. After three successive passes, the last bid becomes the contract. The team who made the final bid will at the moment try to make the contract. 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 shown in Fig. 2.

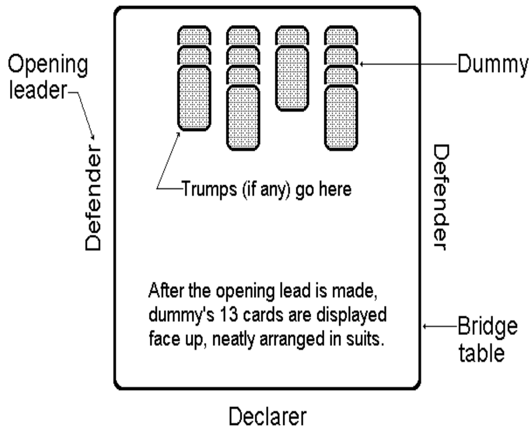

Fig. 2. Bridge Table

The player to the left of the declarer leads to the first trick and instantly after this opening lead, the dummy's cards is showing. The aim of the declarer is to take at least the number of tricks announced during the bidding phase. The players of the opposite pair try to prevent him from doing it [15,16]. In bridge, special focus in game representation is on the fact that players cooperate in pairs, thus sharing potentials of their hands [17].

2.1. Double Dummy Bridge Problem

To estimate the number of tricks to be taken by one pair of bridge players is the basis in double dummy bridge problem. A bridge problem is presented for entertainment, in which the solver is presented with all four hands and is asked to determine the course of play that will achieve or defeat a particular contract. The partners of the declarer, whose cards are placed face up on the table and played by declarer. The dummy has few rights and may not participate in choices concerning the play of the hand. Estimating hands strength is a decisive aspect of the bidding phase of the game of bridge, because the contract bridge is a game with incomplete information and during the bidding phase. This incompleteness of information might allow for many variants of a deal in cards distribution. The player should take into account all these variants and quickly approximation the predictable number of tricks to be taken in each case [18].

2.2. The Bidding

The bidding phase is a conversation between two cooperating team members against an opposing partnership. It aims to decide who will be the declarer. An each partnership uses an established bidding system to exchange information and interpret the partner's bidding sequence. Each player has knowledge of his own hand and any previous bids only. A very interesting characteristic of the bidding phase is cooperation of players in a North with South and West with East. In each, player is modeled as an autonomous, active agent that takes part in the message process. The agent-based algorithm to use of achieve in appropriate learning, a bidding ability close to that of a human expert [19-22].

2.3. The Playing

In the game, the play phase seems to be much less interesting than the bidding phase. ANN approaches tried to imitate the human strategy of the play by using some tactics. The new system was able to find a strategy of play and additionally a human explanation of it [23]. The play proceeds clockwise and each of the other three players in turn must, if potential, play a card of the same suit that the person in charge played. A player with no card of the suit led may play any card of his selection. A trick consists of four cards, one from each player, and is winning by the maximum trump in it, or if no trumps were played by the maximum card of the suit led. The winner of a trick leads to the subsequently and may lead any card. The dummy takes no lively part in the play of the hand and is not permitted to offer any advice or observation on the play. At any time it is dummy's turn to play, the declarer should say which of dummy's cards is to be played, and dummy plays the card as inculcated. Finally, the scoring depends on the number of tricks taken by the declarer team and the contract [24,25].

2.4. No-trump and Trump-suit

A trick contains four cards one contributed by each player and the first player starts by most important card, placing it face up on the table. In a clockwise direction, each player has to track suit, by playing a card of the alike suit as the one led. If a heart is lead, for instance, each player must play a heart if possible. The only if a participant doesn't have a heart he can discard. The maximum card in the suit led wins the trick for the player who played it. This is called playing in no-trump. A No-trump is the maximum ranking denomination in the bidding, in which the play earnings with no-trump suit. The No-trump contracts seem to be potentially simpler than suit ones, because it is not possible to ruff a card of a high rank with a trump card. Though it simplifies the rules, it doesn't simplify the strategy as there is no guarantee that a card will take a trick, still Aces are ineffective in tricks of other suits in no-trump contracts. The success of a contract often lies in the hand making the opening lead. Hence even knowing the location of all cards may sometimes be not sufficient to indicate cards that will take tricks [17]. A card that belongs to the suit has been chosen to have the highest value in a particular game, since a trump can be any of the cards belonging to any one of the players in the pair. The rule of the game still necessitates that if a player can track suit, the player must do so, otherwise a player can no longer go at the rear suit, on the other hand, a trump can be played, and the trump is higher and more influential than any card in the suit led [18].

2.5. Work Point Count System

The Work Point Count System (WPCS) which scores 4 point for Ace, 3 point for King, 2 point for Queen and 1point for a Jack is followed 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 WPCS for getting final contract and out of thirteen tricks in contract bridge, there is a possibility to make use of eight tricks by using WPCS [26].

2.6 Bamberger Point Count System

The Bamberger Point Count System (BPCS) is an exclusive, most important and 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 is followed in which no points are counted for 10 and below.

3. The data representation of GIB Library

The data used in this game of DDBP was taken from the Ginsberg's Intelligent Bridge (GIB) Library. The data created by Ginsberg's intelligent bridge player [14]. The GIB library includes 7,00,000 deals and for each of them provides the number of tricks to be taken by N S pair for each combination of the trump suit and the hand which makes the opening lead. [27].

4. Artificial neural network

The artificial neural network consists of several processing units which are interconnected according to some topology to accomplish a pattern classification task. An artificial neural network is configured for a precise application, such as pattern recognition or data classification through learning process. Artificial neural networks are non-linear information processing devices, which are built from organized elementary processing devices called neurons. In artificial neural network following the supervised learning; each input vector requires a matching target vector, which represents the desired output. The input vector along with the target vector is called training couple. In supervised learning, a supervisor is necessary for error minimization. The consequently network trained by this method is said to be using supervised learning methodology. In supervised learning, it is assumed that the correct target output values are known for each input pattern [28-30].

4.1. Cascade-correlation neural network architecture

The cascade-correlation architecture was introduced by [31] defined with number of input neurons, output neurons represented in the input layer and output layer respectively and hidden neurons are added to the network depends on the necessity of the accuracy of the results. The cascade-correlation begins with a minimal network, then mechanically trains and adds new hidden units one by one, creating a multi-layer configuration. Once a new hidden unit has been added to the network, its input-side weights are frozen. The new hidden neuron is added in each training set and weights are adjusted to minimize the magnitude of the correlation between the new hidden neuron output and the residual error signal on the network output that has to be eliminated. The cascade-correlation architecture has many rewards over its counterpart, as it learns at a faster rate, the network determines its own dimension and topology, it retains the structures it had built, still if the preparation set changes, and it requires no back-propagation of error signals through the associations of the network.

During the learning process, new neurons are added to the network one by one as in Fig.3 and each one of them is placed into a new hidden layer and connected to all the preceding input and hidden neurons. Once a neuron is added to the network and activated, its input connections become frozen and do not change anymore.

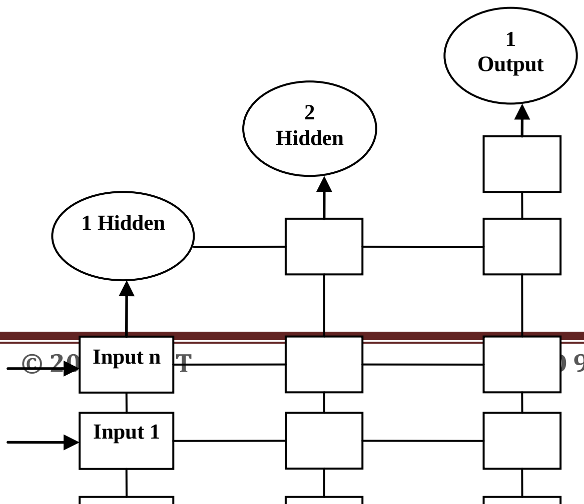

Fig. 3 The architecture of Cascade-Correlation Neural Network (CCNN)

The neuron to be added to the existing network can be made in the following two steps: (i) The candidate neuron is connected to all the input and hidden neurons by trainable input connections, but its output is not connected to the network. Then the weights of the candidate neuron can be trained while all the other weights in the network are frozen. (ii) The candidate is connected to the output neurons and then all the output connections are trained. The whole process is repeated until the desired network accuracy is obtained. In equation (1), the correlation parameter 'S' defined as below is to be maximized.

$$S = \sum_{0=1}^{O} \left| \sum_{p=1}^{P} (V_p - \bar{V})(E_{po} - \bar{E_o}) \right| \qquad (1)$$

where 'O' is the number of network outputs, 'P' is the number of training patterns, 'V_p' is output on the new hidden neuron and 'E_{po}' is the error on the network output. In the equation (2) the weight adjustment for the new neuron can be found by gradient descent rule as

$$\Delta_{w_i} = \sum_{0=1}^{O} \sum_{p=1}^{P} \sigma_o \left(E_{po} - \bar{E_o}\right) f_p' \, x_{ip} \qquad (2)$$

The output neurons are trained using the generalized delta learning rule for faster convergence in Back -Propagation algorithm. Each hidden neuron is trained just once and then its weights are frozen. The network's learning process is completed when satisfied results are obtained. The cascade-correlation architecture needs only a forward sweep to compute the network output and then this information can be used to train the candidate neurons.

4.2 Back-Propagation algorithm

The cascade correlation neural network is a widely used type of architecture consisting of an input layer, a hidden layer, an output layer and two levels of adaptive connections [32]. It is also fully interconnected, i.e. each neuron is connected to all the neurons in the next level. The overall idea behind back propagation is to make large change to a particular weight, 'w', the change leads to a large reduction in the errors observed at the output nodes. In equation (3), let *'y'* be a smooth function of several variables x_i, and it is required to know how to make incremental changes to initial values of each x_i, so as to increase the value of y as fast as possible. The change to each initial x_i value should be in proportion to the partial derivative of *'y'* with respect to that particular *'x_i'*. Suppose that *'y'* is a function of a several intermediate variables *'x_i'* and that each *'x_i'* is a function of one variable *'z'* and we want to know the derivative of *'y'* with respect to *'z'*, then using the chain rule.

$$\Delta x_i \infty \frac{\partial y}{\partial x_i} \tag{3}$$

$$\frac{dy}{dz} = \sum_i \frac{\partial y}{\partial x_i}\frac{dx_i}{dz} = \sum_i \frac{dx_i}{dz}\frac{\partial y}{\partial x_i} \tag{4}$$

The standard way of measuring performance is to pick a particular sample input and then sum up the squared error at each of the outputs. We sum over all sample inputs and add a minus sign for an overall measurement of performance that peaks at o.

$$P = -\sum_s \left(\sum_z (d_{sz} - o_{sz})^2 \right) \tag{5}$$

Where 'P' is the measured performance, S is an index that ranges over all sample inputs, Z is an index that ranges overall output nodes, d_{sz} is the desired output for sample input 's' at the z^{th} node, o_{sz} is the actual output for sample input 's' at the z^{th} node. The performance measure P is a function of the weights and the idea of gradient ascent can be deployed if one can calculate the partial derivative of performance with respect to each digit. With these partial derivatives in hand, one can climb the performance hill most rapidly by altering all weights in proportion to the corresponding partial derivative. The performance is given as a sum over all sample inputs. We can compute the partial derivative of performance with respect to a particular weight by adding up the partial derivative of performance for each sample input considered separately. The equation (6) each weight will be adjusted by summing the adjustments derived from each sample input. Consider the partial derivative

$$\frac{\partial P}{\partial w_{i\to j}} \tag{6}$$

where the weight is a weight connecting i^{th} layer of nodes to j^{th} layer of nodes. The equation (7) our goal is to find an efficient way to compute the partial derivative of P with with respect to . The effect of on value P, is through the intermediate variable o_j, the output of the j^{th} node and using the chain rule, it is express as

$$\frac{\partial P}{\partial w_{i\to j}} = \frac{\partial P}{\partial o_j}\frac{\partial o_j}{\partial w_{i\to j}} = \frac{\partial o_j}{\partial w_{i\to j}}\frac{\partial P}{\partial o_j} \tag{7}$$

Determine o_j by adding up all the inputs to node *'j'* and passing the results through a function.

$$o_j = f\left(\sum_i 0_i w_{i\to j} \right) \tag{8}$$

Hence,
where is a threshold function. Let

$$\sigma_j = \sum_i 0_i w_{i\to j} \tag{9}$$

We can apply the chain rule again.

$$\frac{\partial o_j}{\partial w_{i\to j}} = \frac{df(\sigma_j)}{d\sigma_j}\frac{\partial \sigma_j}{\partial w_{i\to j}} \tag{9}$$

$$\frac{\partial P}{\partial o_j} \quad \frac{df(\sigma_j)}{d\sigma_j} o_i \quad \frac{\partial P}{\partial o_k} \tag{10}$$

$$\frac{\partial P}{\partial w_{i\to j}} = o_i \quad \frac{df(\sigma_j)}{d\sigma_j} \quad \frac{\partial P}{\partial o_j} \tag{11}$$

Substituting Equation (8) in Equation (5), we have

$$\frac{\partial P}{\partial o_j} = \partial w_{i\to j}\frac{df(\sigma_k)}{d\sigma_k} \quad \frac{\partial P}{\partial o_k} \tag{12}$$

$$\frac{\partial P}{\partial w_{i\to j}} = o_i \quad \frac{df(\sigma_j)}{d\sigma_j} w_{j\to k} \quad \frac{df(\sigma_k)}{d\sigma_k}\frac{\partial P}{\partial o_k} \tag{13}$$

Thus, the two important consequences of the above equations are, 1) The partial derivative of performance with respect to a weight depends on the partial derivative of performance with respect to the following output. 2) The partial derivative of performance with respect to one output depends on the partial derivative of performance with respect to the outputs in the next layer. The system error will be reduced if the error for each training pattern is reduced. The equation (14) and (15) thus, at step's+1' of the training process, the weight adjustment should be proportional to the derivative of the error measure computed on iteration's'. This can be written as

$$\Delta w(s+1) = -n\partial P / \partial w(s) \tag{14}$$

$$\left[\Delta w(s+1) = -n\frac{\partial P}{\partial w} + a\Delta w(s) \right] \tag{15}$$

where η is a constant learning coefficient, and there is another possible way to improve the rate of convergence by adding some inertia or momentum to the gradient expression, accomplished by adding a fraction of the previous weight change with current weight change. The addition of such term helps to smooth out the descent path by preventing extreme changes in the gradient due to local anomalies. 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.

5. Neural network in double dummy bridge problem

There are several neural network architectures have been used to solving the double dummy bridge problem. In this paper we focus cascade-correlation neural network architecture 52(13x4) for solving the DDBP in contract bridge.

5.1. 52 (13x4) Representation

In this architecture, positions of cards in the input layer were fixed, i.e. from the leftmost input neuron to the rightmost one the following cards were represented: 2♠, 3♠, . . . , *K*♠, *A*♠, 2♥, . . . , *A*♥, 2♦, . . . , *A*♦, 2♣, . .. , *A*♣ Fig. 4. This way each of the 52 input neurons was assigned to a particular card from a deck and a value presented to this neuron determined the hand to which the respective card belonged, i.e. 1.0 for *North*, 0.8 for *South*, −1.0 for *West*, and −0.8 for *East*.

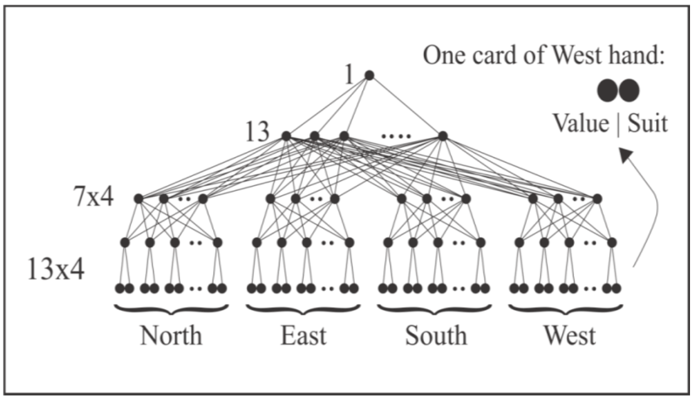

Fig. 4. Neural Network Architecture with 52 input neurons

Layers were fully connected, i.e., in the 52 – 25 – 1 network all 52 input neurons where connected to all 25 hidden ones, and all hidden neurons were connected to a single output neuron.

6. WPCS and BPCS with implementation

The human point count methods 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. Though there are many human point count methods such as, Collet point count, Four aces points, Polish points etc., are available, work point count method and Bamberger point count are employed in our discussion, because these are the most widely used point counting systems. A work point count system, which scores 4 points for Ace, 3 points for a King, 2 points for a Queen and 1 point for a Jack and Bamberger point count systems scores about 7 points for an Ace, 5 points for a King, 3 points for a Queen and 1 point for a Jack . 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's lengths and existence of groups of honors in one suit. The another important pattern is a group of honors in one suit located in the cards of both players in a pair, since having a group of top honors in a suit allows predicting more precisely the number of tricks available in this suit.

6.1. Input Layer

52 cards were used in input layer. Each member was received 13 cards. The card values are determined in rank card (2, 3, K, A) and suit card (♠ (S), ♥ (H), ♦ (D), ♣(C)). The rank card is transformed using a uniform linear transformation to the range from 0.10 to 0.90. The Smallest card value is 2(0.10) and highest card value is A (0.90). The suit cards are a real number of using the following mapping: Spades (0.3), Hearts (0.5), Diamonds (0.7) and Clubs (0.9).All combination cards value rank and suit cards represented by one hand.

6.2. Hidden Layer

There is a middle layered of hidden and internal representation 25 neuron were fully connected. The basically 4 suits, the power of trump suit, the weight of a rank card, the highest of *Ace* and lowest is *two*. The neuron representing a hand to which the card actually received input value equal 1.0. The other three neurons were assigned input values equal to 0.0.

6.3. Output Layer

In this layer only one output was received and getting the result, decision boundaries were defined within the range of (0.1 to 0.9). The results were defined a priori and target of ranges from 0 to 13 for all possible number of tricks was the use of a linear transformation. A Gradient descent training function was used to train the data and gradient descent weight/bias learning function was used for learning the data. For training and learning the data, CCNN is used in hyperbolic tangent sigmoid function.

7. Results and discussion

In this paper sample deals data were used for training 5000 and testing 2500 in MATLAB 2011a. Together there are 20 numbers of each deal i.e. 5 trump suits by 4 sides. Here 5 trump suits are No-trumps, spades, Hearts, Diamonds and Clubs, No-trump which is the term for contracts played without trump. The Four sides are West, North, East and South. The North and South are partners playing against East and West. The results presented in the Fig. 5 and Fig. 6 shown that the comparison of target tricks along with CCNN. While comparing the train and test data along with target data, the result indicated that, train and test data shown significantly better results in both methods, which minimized the mean square error (MSE).

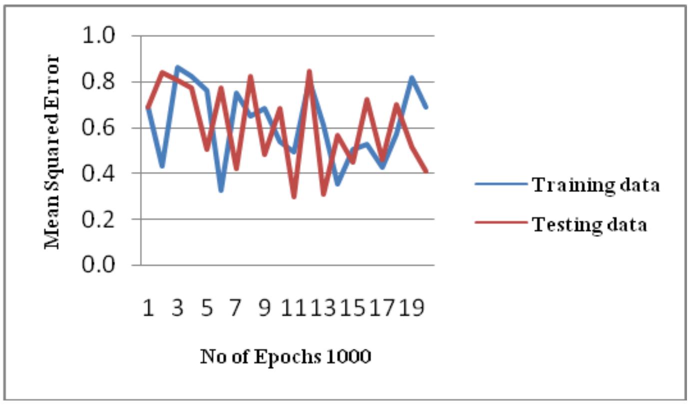

Fig.5 Mean squared error (MSE) during training and testing phase of bamberger point count system

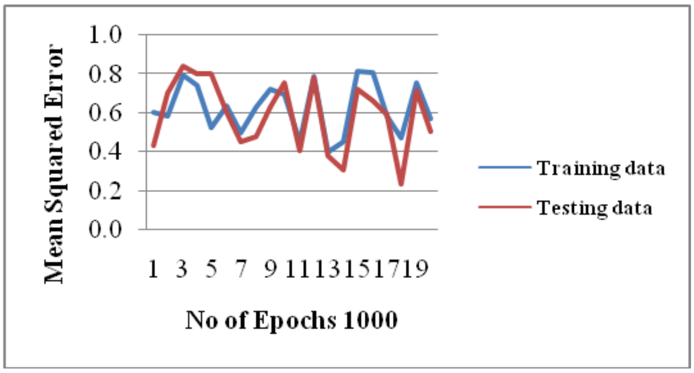

Fig.6 Mean squared error (MSE) during training and testing phase of work point count system

The data trained and tested through this CCNN shows better performance and the time taken for training and testing the data were relatively minimum which also converged to the error steadily during the whole process. The WPCS and BPCS were compared with each other and BPCS was given significantly superior results than WPCS. The during bidding phase of contract bridge, hyperbolic tangent sigmoid function was used in CCNN architecture in BP algorithm to take best BPCS for getting final contract in bridge game.

7.1. Sample deals 7♠with 41 points using BPCS

In the first example, since A K Q represented in ♠ amounts to the maximum points in North side, hence found in NS pair, it is aimed to score 41 points, which is possible with 8 tricks to all players, irrespective of their level of mastery of bridge game in Fig.7. While incorporating Bamberger point count method, it is possible to find the missing trick i.e., ◆ also, using the Cascade-Correlation neural network model discussed in this paper.

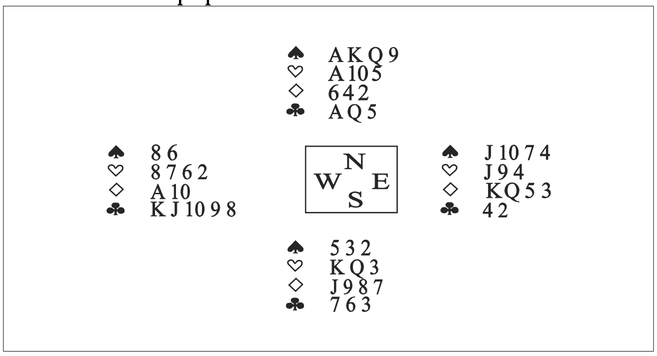

Fig. 7. A Sample deal (NS pair in ♠contract with South opening lead)

8. Conclusion and future work

In this paper, the artificial neural networks which were used to estimate the number tricks to be taken by one pair of players in the double dummy bridge problem in contract bridge. In cascade-correlation neural network architecture, during training process new hidden nodes are added to the network one by one. For each new hidden node, the correlation magnitude between the new node output and the residual error signal is maximized. The during the time when the node is being added to the network, the input weights of hidden nodes are frozen, and only the output connections are trained repeatedly. The WPCS and BPCS are excellent, even though both systems the WPCS and BPCS produced better results; BPCS has given significantly superior result than WPCS. The BPCS used in BP algorithm which produced better results and used to bid a final contract bridge. The BPCS is a good information system and it provides some new ideas to the bridge players and helpful for beginners and semi professional players also in improving their bridge skills. Furthermore we would enlarge the hybrid architecture and different algorithms to solve DDBP more efficiently and effectively.

References

[1] Mandziuk J (2010) Knowledge-free and Learning –Based methods in Intelligent Game Playing, Springer.

[2] Mandziuk J (2007) Computational Intelligence in Mind Games, In: Studies in Computational Intelligence, Springer.

[3] Frank I, Basin D A (2001) A Theoretical and Empirical Investigation of Search in Imperfect

Information Game, Theor.Comput.Sci.vol.no252, pp.217-256.
[4] Mandziuk J (2008) Some thoughts on using Computational Intelligence methods in classical mind board games, In: Procedings of the 2008 International Joint Conference on Neural Networks (IJCNN 2008), Hong Kong, China, pp.4001-4007.
[5] Mandziuk J, Mossakowski K (2004) Looking Inside Neural Networks Trained to Solve Double-Dummy Bridge Problems, Int.Proceeding 5th Game – On Computer Games: Artif. Intell., U.K. pp.182-186.
[6] Sivanandam S N , Deepa S N (2007) Principles of Soft Computing,Wily.
[7] Mandziuk J, Mossakowski K (2009) Neural networks compete with expert human players in solving the double dummy bridge problem Proc. of 5th Int. Conf. on Computational Intelligence and games, pp.117-124.
[8] Mossakowski K, Mandziuk J (2004) Artificial neural networks for solving double dummy bridge problems, In: AI and Soft computing vol.no 3070, Springer, pp.915-921.
[9] Sarkar M, Yegnanarayana B, Khemani D (1995) Application of neural network in contract bridge bidding, in Proc. of National Conf. on Neural Networks and Fuzzy Systems, Anna University, Madras, pp. 144-151.
[10] Dharmalingam M, Amalraj R (2013) Neural Network Architectures for Solving the Double Dummy Bridge Problem in Contract Bridge, in Proce of the PSG-ACM National Conference on Intelligent Computing, pp 31-37.
[11] Frank I, Basin D A (1999) Optimal play against best Defence: Complexity and Heuristics in Lecture Notes in Computer Science Germany: Springer-Verlag, vol.no1558, pp 50-73.
[12] Smith S J J, Nau D S (1994) An Analysis of Forward Pruning, in Proce of the National Conference on AI, pp 1386-1391.
[13] Smith S J J, Nau D S, Throop T A (1998) Success in Spades: Using AI Planning Techniques to Win the World Championship of Computer Bridge, in Proce of the National Conference on AI, pp.1079-1086.
[14] Ginsberg M L (2001) GIB: Imperfect Information in a Computationally Challenging Game, Journal of Artificial Intelligence Research, vol.14, pp.303-358.
[15] Francis H, Truscott A, Francis D (2001) The Official Encyclopedia of Bridge, 6th ed. Memphis, TN: American Contract Bridge League.
[16] Root W H (1998) The ABCs of Bridge, Three Rivers Press.
[17] 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. Comput. Intell. Data mining, pp.413-420.
[18] Mossakowski K, Mandziuk J (2009) Learning without human expertise: A case study of Double Dummy Bridge Problem, IEEE Transactions on Neural Networks, vol.20, no.2, pp.278-299.
[19] Amit A, Markovitch S (2006) Learning to bid in bridge Machine Learning, vol.63, no.3, pp.287-327.
[20] Ando T, Uehara T (2001) Reasoning by agents in computer bridge bidding, in Computers and Games, vol. 2063, pp.346-364.
[21] 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, vol. 86, no.12, pp.76-86.
[22] Ando T, Sekiya Y, Uehara T (2000) Partnership bidding for Computer Bridge, Systems and Computers in Japan, vol.31,no.2, pp.72-82.
[23] Khemani D (1994) Planning with thematic actions, in AIPS, pp.287-292.
[24] Frank I, Basin D A (1999) Strategies explained in Proceeding 5th Game programming Workshop in Japan, pp.1-8.
[25] Awada A, Dehayni M, Yaacoub A (2011) An ATMS-Based Tool for Locating Honor Cards in Rubber Bridge, Journal of Emerging Trends in Computing and Information Sciences, vol. 2 no.5, pp.209-218.
[26] Amalraj R, Dharmalingam M (2015) A work point count system coupled with back-propagation for solving double dummy bridge problem, Neurocomputing, vol.168, pp.160-178
[27] Mossakowski K, Mandziuk J (2006) Neural networks and the estimation of hands strength in contract bridge, in Artificial Intelligence and Soft Computing ICAISC vol.4029. Springer, pp.1189-1198.
[28] Yegnanarayana B (2010) Artificial Neural Networks, Pretince Hall.
[29] Yegnanarayana B, Khemani D, Sarkar M (1996) Neural networks for contract bridge bidding, Sadhana,vol.21,no.3, pp.395-413.
[30] Sivanandam S N, Paulraj M (2011) Introduction Artificial Neural Networks, Vikas.
[31] Fahlman S E, Lebiere V (1990) The cascade-correlation learning architecture, Advances in Neural Information Processing, pp.524-532.
[32] Rumelhart D E, Hinton G E, Williams R J (1986) Learning internal representations by error back propagation, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol.1, pp.533-536.

BIOGRAPHIE

Dr. M Dharmalingam received his Under-Graduate, Post-Graduate and Master of Philosophy degrees from Bharathiar University, Coimbatore in the years 2000, 2004 and 2008 respectively. He has completed his PhD degree in Computer Science in 2015, at Bharathiar University, Coimbatore. Currently he is working as an Associate Professor in Computer Science, Nandha Arts and Science College, Erode. He has published several research papers in reputed National and International journals and the broad field of his research interest is soft computing.