

Neural networks compete with expert human players in solving the Double Dummy Bridge Problem

Jacek Mańdziuk, *Member, IEEE*, and Krzysztof Mossakowski

Abstract—Artificial neural networks, trained only on sample bridge deals, without presentation of any human knowledge as well as the rules of the game, are applied to solving the Double Dummy Bridge Problem (DDBP). The problem, in its basic form, consist in estimation of the number of tricks to be taken by one pair of bridge players.

The efficacy of knowledge-free neural network approach is compared with the case of applying human estimators of bridge hands' strengths (typically used by professional players) as additional input information.

Furthermore, a comparison with the results obtained by 24 professional human bridge players - members of The Polish Bridge Union - on sample test sets is presented, leading to interesting observations about suitability of particular types of deals for artificial systems and for human bridge players.

I. INTRODUCTION

Computational Intelligence methods, when applied to games, are mainly focused on the aspect of learning in the game playing systems. The baseline for these approaches is autonomous development of the game playing skills based on appropriately designed training procedure. CI learning attempts often assume that *a priori* knowledge is either completely absent or restricted to the game rules only.

The most popular learning frameworks of this type include artificial neural networks, evolutionary methods and reinforcement learning systems. In particular, neural networks in example-based training regime are often considered, mainly due to their excellent generalization capabilities and after-training high-speed.

In this paper neural networks are applied to the so-called Double Dummy Bridge Problem. The efficacy of proposed knowledge-free learning scheme is compared with the results of neural nets training which includes hand strength estimators used by professional bridge players in tournament situations. Furthermore, a selected group of professional bridge players is tested on small subsets of test deals for another comparison and results assessment.

The reminder of the paper is organized as follows. The DDBP is introduced in section II. The next section briefly describes the GIB Library, from which training and testing deals were extracted. The overview of application of neural networks in solving the DDBP is presented in section IV, which includes the results of both "pure" zero-domain-knowledge approach and the use of hand strength estimators as supportive training data. The results of human bridge

professionals on the subsets of test deals are discussed in section V. The next section is devoted to presentation of sample deals which present strong and weak points of neural networks (versus humans) in solving the DDBP. Section VII concludes the paper.

Due to space limits only a very basic introduction to the game of bridge is provided in the paper. The interested reader may be willing to consult [1] regarding this issue. Likewise, for other applications of neural nets to bridge related problems (especially the bidding phase), the reader may refer to [2], [3], [4], [5].

II. THE DOUBLE DUMMY BRIDGE PROBLEM

Contract bridge, usually known simply as bridge, is a trick-taking card game. The game is played by four players in two fixed pairs. Partners sit facing each other. Players are referred to according to their position at the table as *North(N)*, *East(E)*, *South(S)*, and *West(W)*, so the competing pairs are composed by (*N*, *S*) and (*W*, *E*) players. A standard 52 card pack is used. In each suit card's rank from highest to lowest is the following: *Ace(A)*, *King(K)*, *Queen(Q)*, *Jack*, 10, ..., 3, 2. The whole deck is dealt out, so each player receives 13 cards.

Next an auction to decide who will be the declarer takes place. A bid specifies a number of tricks and a trump suit (or that there will be no trumps). 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 "notrump" which is the term for contracts played without a trump. After three consecutive passes, the last bid becomes the contract. The team who made the final bid will now try to make the contract. The first player of this team who mentioned the denomination (suit or *notrump*) of the contract becomes the declarer. The declarer's partner is known as the dummy.

The player to the left of the declarer leads to the first trick. Immediately after this opening lead, the dummy's cards are exposed. The play proceeds clockwise. Each player must, if possible, play a card of the suit led. A player with no card of the suit led may play any card. A trick consists of four cards, and is won by the highest trump in it, or if no trumps were played, by the highest card of the suit led. The winner of a trick leads to the next.

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. Details of scoring depends on the variant of the game [1], [6].

Jacek Mańdziuk is with the Faculty of Mathematics and Information Science, Warsaw University of Technology, Warsaw, POLAND (email: mandziuk@mini.pw.edu.pl). Krzysztof Mossakowski is a part-time lecturer at the same faculty.

The Double Dummy Bridge Problem (DDBP) is an auxiliary bridge problem closely connected to the bidding phase of the game. More specifically the problem consists in answering the following question: “How many tricks are to be taken by one pair of players assuming perfect play of all four sides, with all four hands being revealed?”.

There is an important difference between solving DDBP and the real bridge playing, since in the latter, the exact placement of most of the cards is unknown. The DDBP can, however, be effectively used as supporting tool in real bridge playing, based on the following reasoning: Estimating hand’s strength is a crucial aspect of the bidding phase of the game of bridge, since the contract bridge is a game with incomplete information and during the bidding phase each player can see only their cards and has to make several assumptions about placement of other cards. This incompleteness of information forces considering many variants of a deal (cards distributions). The player should take into account all these variants and quickly estimate the expected number of tricks to be taken in each case. In actual play among professional players the location of crucial cards can be partly inferred from the bidding phase. Considering that it is worth to note, that assuming any particular variant of cards’ location is equivalent to the case of having all four hands revealed, *ergo* the DDBP.

The above observation, first discussed in [7], led to the following idea: use the assumption of particular cards’ placements combined with Monte Carlo simulations and a fast DDBP solver in order to estimate the most probable (the most effective) contract [7]. This idea was used by Ginsberg in his computer world champion playing program [8].

Neural networks were not very popular in solving the DDBP. To the authors knowledge the only relevant attempt was published by Gambäck and Rayner [9], [10], who proposed to use two networks, one for *notrump* contracts and one for suit contracts. Each network took as the input the sets of cards from four hands, and output a set of 14 real numbers, representing the estimated probabilities that 0 to 13 tricks will be collected by the playing pair. The authors concluded that the networks in their purest form (using only raw input data, without any preprocessing of the input or adding human knowledge in any form) were not able to successfully serve as estimators of the number of tricks. Adding some input data which humans consider important in the domain (called “pre-computed feature points” by the authors) improved the results. Unfortunately no numerical results are presented in [9], [10], which does not allow for any direct comparisons.

On the contrary to the above cited articles the results of experiments performed by Mossakowski and Mańdziuk [11], [12] suggest that with appropriate representation of a deal, neural networks can be very effective in solving the DDBP relying solely on example-based training and are capable of extracting necessary information from raw data, without any human intervention or use of human knowledge about the game of bridge.

III. THE GIB LIBRARY

The data used in solving the DDBP was taken from the GIB Library [13], which was created with the help of the Ginsberg’s Intelligent Bridgeplayer [8] - the above mentioned computer bridge champion in 1998 and 1999.

The GIB Library includes over 700,000 deals and for each of them provides the numbers of tricks to be taken by the *NS* pair for each combination of the trump suit (including *notrump* contracts) and the hand which makes the opening lead. Together there are 20 numbers for each deal (5 trump suits by 4 sides). All these numbers were calculated by the GIB program under the assumption of a perfect play of all players.

In all experiments reported in this paper, 100,000 deals from the library were used for training and another 100,000 ones were used for testing. Preliminary tests confirmed no further improvement in the case of using bigger data sets.

IV. APPLICATION OF NEURAL NETWORKS TO SOLVING THE DDBP

Four types of neural architectures have been tested in the experiments differing mainly by the way a deal was represented in the input layer. Since description of these architectures as well as detailed discussion on their advantages and weak points were presented in our previous papers [11], [12], we only focus here on the two types of networks, which were actually used in the competition against human bridge players.

A. (52x4) input coding

Neural architectures with the 52x4 input coding appeared to be superior among all tested architectures. In this deal representation, 208 input neurons were divided into 4 groups, one group per hand, respectively for *N*, *E*, *S* and *W* players. Four input neurons (one per hand) were assigned to each card from a deck. The neuron representing a hand to which the card actually belonged received input value equal to 1.0. The other three neurons (representing the remaining hands) were assigned input values equal to 0.0. This way, a hand to which the card was assigned in a deal was explicitly pointed out.

In this representation one suit on one hand was represented by 13 input neurons. The number of inputs which were equal to 1.0 determined the length of this suit in the hand, so the networks had a “straightforward” chance to find out the suit lengths in respective hands, in particular to determine the shortnesses (which are very important in bridge), especially voids (no cards in a suit) and singletons (one card in a suit).

There were 4 groups of neurons in the first hidden layer, each of them gathering information from 52 input neurons representing one hand. This data was further compressed in another one or two fully connected hidden layers. All neurons from the last hidden layer were connected to a single output neuron, which denoted the number of tricks to be taken by *NS*. The decision boundaries were defined *a priori* and target ranges for all possible numbers of tricks (from 0 to 13) were of pairwise equal length within the output interval [0.1, 0.9].

A sample network using this way of coding a deal, $(52 \times 4) - (13 \times 4) - 13 - 1$, is presented in Fig. 1. The 3-hidden layer architecture was, for example, represented by the network $(52 \times 4) - (26 \times 4) - 26 - 13 - 1$.

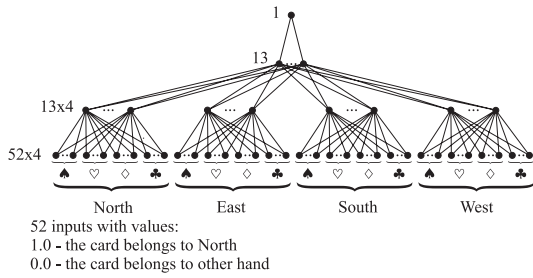


Fig. 1. Example of neural architecture implementing the (52×4) input coding.

B. 52 input coding

In the other way of deal representation considered in this paper, a different idea was utilized. Each input neuron was assigned to a particular card from a deal and a value presented to this neuron determined the hand to which the respective card (assigned to this input) belonged.

There were 52 input values, each representing one card from a deck. 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: $2\spadesuit, 3\spadesuit, \dots, K\spadesuit, A\spadesuit, 2\heartsuit, \dots, A\heartsuit, 2\diamondsuit, \dots, A\diamondsuit, 2\clubsuit, \dots, A\clubsuit$ (see Fig. 2).

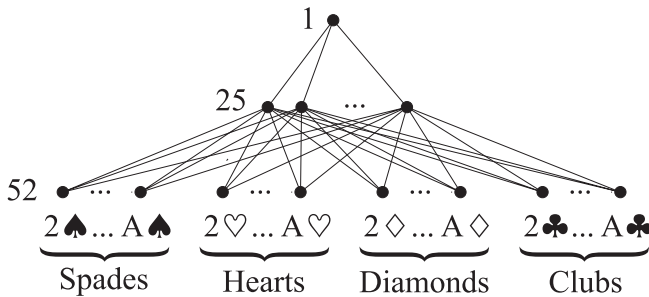


Fig. 2. Example of neural architecture implementing the 52 input coding. In this coding there were no dedicated groups of neurons in the hidden layers. Layers were fully connected, e.g. in the $52 - 25 - 1$ network all 52 input neurons were connected to all 25 hidden ones, and all hidden neurons were connected to a single output neuron.

A value presented to the input neuron denoted the hand to which a given card belonged. In the majority of the initial experiments the following coding was applied: 1.0 for *North*, 0.8 for *South*, -1.0 for *West*, and -0.8 for *East*. Interestingly, as came out from further experiments, using the same input value (-1.0) for both *West* and *East* and the common value (1.0) to denote both *North* and *South* hands improved the results. Hence, all experiments related to the 52 input representation presented in this paper were conducted with the use of the above mentioned two-state input representation. In the hidden and output layers,

sigmoidal neurons with the same sigmoidal transfer functions as in the case of 52×4 , were used.

C. DDBP-4 and DDBP-2

Strictly bipolar coding used in the 52 input representation is equivalent to hiding the information about exact cards' assignment in the *NS* and *WE* pairs, leaving only the *sum of both hands* available for the networks.

In practice, from the point of view of bridge rules, a situation in which the exact location of the cards is partly unknown, is more realistic, than the case of revealing all four hands. It seems reasonable to assume that during the bidding phase an experienced player can learn about the strengths and weaknesses of his partner's hand quite precisely (under the assumption that the players have a great deal of experience in playing together). Thus the assumption that partner's cards are known to the player is to some extent justified. The remaining half of a deal is at the opponent's disposal. Certainly some information concerning cards distribution in the opponents hands can also be inferred from the bidding phase, but the hands of the opponents should be treated, in principle, as covered.

In the reminder of the paper in order to distinguish between these two versions of the DDBP they will be referred to as DDBP-4 (or simply DDBP) in case of fully uncovered version of the problem and DDBP-2 in case of the opponents' hands being hidden. The former one will be implemented by 52×4 input coding and the latter variant by the $52 - 25 - 1$ architecture, with bipolar input representation.

Since no reference to the DDBP-2 have been found in the literature, in order to assess the networks' results in this case, a specially designed experiment with human professional bridge players has been performed (see section V).

D. Numerical results

The networks were trained for a few (up to 20) thousand epochs, starting with random weights, using the Rprop algorithm with the following set of parameters: initial and maximum values of an update-value factor were equal to 0.1 and 50.0 respectively, and weight decay parameter was equal to $1E - 4$. The unipolar sigmoid activation function ($f(x) = \frac{1}{1+e^{-x}}$) was used in all neurons.

The best results obtained by both types of networks respectively for *suit* and *notrump* contracts are presented in Table I. In each field of the table the three values $A | B | C$, denote respectively the fraction in percent of perfectly solved problems (C), the fraction of problems for which there was at most one trick error (B), and the fraction of the problems with the error not exceeding two tricks (A). For example the result, 99.80 | 95.54 | 50.91, means that the network answered perfectly in 50.91% of testing deals and was wrong by more than one trick only in 4.46% of the problems and by more than two tricks in 0.2% of testing deals.

The last column of the table refers to the experiments in which, in the training phase, each deal was presented twice with changing the hand making the opening lead. Since in about 7% of test deals the estimation of the number of

tricks to be taken depends on the choice of the opening lead hand, allowing the network to consider both possibilities improved the basic result by another 2.2% in the case of suit contracts (raising the result to 53.11%). The details of the experiments performed with changing the opening lead hand were presented in [14].

E. Analysis of weight patterns in the trained networks

The in-depth analysis of connection weights of trained networks revealed the existence of weight patterns “responsible” for particular aspects of the overall solution of the problem. These patterns can be easily recognized and explained by experienced human bridge players. In other words it turned out that the representation of problem specific knowledge gained through the learning process, although redundant, is highly specialized and several aspects indispensable for high playing competency in the game of bridge can be explicitly pointed out in the network’s weight space.

Analysis of connection weights of trained networks revealed patterns which can be explained using human knowledge about the game. The most common patterns (found practically in all networks) are: preference for honors with special attention put to *Aces*, favoring trump suit cards, and gradual importance of cards from *two* to the *Ace* in each suit. Specialization of particular neurons in the above features is very clear. Weight patterns are repeatable for the whole ensemble of randomly initialized networks. All of the above relationships are crucial and well-known even for novice human players. All of them, on the other hand, were discovered by the networks themselves in the blind example-based training regime, without explicitly adding any domain knowledge.

The investigation of weight patterns of the trained networks was a subject of the authors’ previous works [15], [16].

F. Extension of the input by adding human bridge estimators

All results presented in previous sections were accomplished by artificial neural networks trained only on examples, without presentation of human knowledge about the game of bridge in any form.

In this section, for comparison purposes, results of training *with* additional human knowledge are summarized (the details of implementation were presented in our previous papers [17], [14]). The human knowledge was represented by various numerical estimators of hand’s strength used by experienced human bridge players in order to declare the optimal possible contract. These estimators were added to previously used deal representation through additional input neurons.

Human estimators of hand’s strength can be divided into two categories: point count methods and distributional points methods.

1) *Point Count Methods*: Human point count methods are based on calculating the strength of a hand as a sum of single cards’ strengths [1]. In these methods, the value of each card depends only on card’s rank. The most widely used points

TABLE II
HUMAN POINT COUNT METHODS

Method	A	K	Q	J	10
Work Point Count	4	3	2	1	0
Bamberger Point Count	7	5	3	1	0
Collet Point Count	4	3	2	0.5	0.5
Four aces points	3	2	1	0.5	0
Polish points	7	4	3	0	0
Reith Point Count	6	4	3	2	1
Robertson Point Count	7	5	3	2	1
Vernes Point Count	4	3.1	1.9	0.9	0
AKQ points	4	3	2	0	0

counting system is called Work Point Count (WPC), which scores 4 points for an *Ace*, 3 points for a *King*, 2 points for a *Queen*, and 1 point for a *Jack*. Table II presents other popular human point count methods which were used in the experiments.

2) *Distributional Points Methods*: The other category of human hand’s strength estimators contains the so-called distributional points [1]. These methods score patterns which can be found in a set of cards assigned to one hand. The most important patterns are: suits’ lengths and existence of groups of honors in one suit. Even novice bridge players know, that a void (lack of cards in a suit) or a singleton (single card in a suit) are very valuable in suit contracts (certainly it does not regard the trump suit), so almost all distributional points methods award such shortness.

Another very important pattern which is appreciated is a group of honors in one suit located in the cards of both players in a pair. Having a group of top honors in a suit allows to predict more precisely the number of tricks available in this suit.

The following eight distributional points methods were used in the experiments (please consult [1] or [14] for their formal definition): Honor Trick Count; Playing Trick Count; Losing Trick Count; Asset System; Stayman Point Count; Rule of three and four; *Moins-value*; Plus-value.

Various combinations of 52×4 input representation and the selected point count or distributional points methods were verified in extensive experiments. The best overall combination was obtained by putting together all 9 point count methods (36 additional inputs) with all 8 distributional points methods (32 inputs) and with the information about the lengths of all suits in all hands (16 inputs). In the resulting architecture: $(52 \times 4 + 84) - (13 \times 4 + 21) - 26 - 1$ each of the basic 52 neurons describing particular hand was fully connected to the dedicated set of 13 neurons in the first hidden layer and the additional 84 input neurons were fully connected to the separate set of 21 first hidden layer nodes. The second hidden layer was composed of 26 neurons fully connected to the first hidden layer nodes. These second hidden layer nodes were connected to the single output. This architecture scored $97.37 \mid 84.99 \mid 38.78$ and $99.84 \mid 96.12 \mid 52.47$, respectively for *notrump* and suit contracts. It is

TABLE I
COMPARISON OF RESULTS OBTAINED FOR *Notrump* CONTRACTS AND FOR SUIT CONTRACTS WITH AND WITHOUT CHANGING A HAND MAKING THE OPENING LEAD

The Network	<i>Notrump</i> Contracts	<i>Suit(Spades)</i> Contracts	<i>Suit(Spades)</i> Contracts with Changing the Opening Lead Hand
52-25-1	96.07 80.88 34.66	98.77 88.00 40.13	98.49 87.15 39.29
(52x4)-(13x4)-13-1	97.34 84.31 37.80	99.78 95.00 50.03	99.79 95.49 50.62
(52x4)-(26x4)-26-13-1	96.89 83.64 37.31	99.80 95.54 50.91	99.88 96.48 53.11

important to note, that extending the deal representation by adding human estimators did not improve the best overall result accomplished by “pure” $52x4$ architecture in the case of suit contracts and only slight improvement was notified in the case of *notrump* contracts. This observation suggests that the relevance of additional information related to suit lengths and points distribution in particular hands has been autonomously discovered by the best $52x4$ networks during the training process.

V. COMPETING WITH HUMAN BRIDGE PLAYERS

A group of 24 professional players, the members of The Polish Bridge Union took part in the DDBP experiment through the Internet. Depending on time availability each of them solved between 27 and 864 instances of the problem grouped into 27 deal chunks¹. The players, based on their professional accomplishments (the bridge titles and the ranks of competitions they took part in at the international and national level) were divided into two groups. The first group was composed of 10 upper-classified players (four Grand Masters, three International Masters and three Masters, playing in the First or the Second Polish Bridge League) including one player from the top-10 players in Poland in the 2007 ranking. The remaining 14 players (members of the lower-ranked bridge teams, each having a professional bridge title) composed the second group. These groups of players will be denoted by “Group-1” and “Group-2”, respectively.

The participants of the experiment were faced with both types of DDBP deals, i.e. DDBP-4 and DDBP-2 ones. In each 27-deal chunk only deals of one type were served. For each deal the information about the type of contract (*notrump* or *spades*) and the hand making the opening lead was provided. The *NS* pair was either the declaring pair or the defending one. There was a 30 second slot allotted for each test problem. This amount of time was chosen after some preliminary experiments and seems to be sufficient for experienced players. Before starting the experiment the players had a chance for some training in order to get used to the environment and the rules of the experiment.

Various statistics were collected from the experiment. The overview of the main results is presented in Table III.

¹In fact there were more players involved in the experiment. After the contest the results were restricted to the players having at least the lowest possible professional bridge title and at the same time the ones that solved at least one full chunk of deals.

Generally speaking, human players were able to outperform neural networks only in the case of DDBP-4 and *notrump* contracts, where the score of the Group-1 players is respectful. In the remaining three categories the respective networks ($52x4$ in case of DDBP-4 and 52 in case of DDBP-2) are comparable to those of humans, especially when the one-trick and two-trick margins are considered.

Analysis of results leads to two specific observations: first of all, humans are visibly better at solving the *notrump* contracts than the suit ones. Second of all, the opposite conclusion is valid in the case of neural networks.

The reason for human better performance in *notrump* contracts than in the suit ones is twofold. Firstly, the tournament statistics show that a distribution of contracts being played is approximately equal to 60, 35 and 5%, respectively for *notrump*, *spades* and *hearts*, and *diamonds* and *clubs* [18]. Hence, humans are more used to playing *notrump* contracts than the suit ones. Secondly, *notrump* contracts are easier to be played than the suit ones (roughly speaking playing the suit contract includes all the techniques used for playing the *notrump* contract and additionally several other maneuvers related to the use of trump cards) [18].

The opposite effect observed in the case of neural networks can be attributed to the fact that human way of solving the DDBP is very different from that of neural networks. Humans, when analyzing a deal, despite scoring the hands and locating the honors also try to virtually simulate the play phase. Neural networks are restricted to thorough analysis of cards distribution, which includes location of honors and lengths of suits, but does not include play phase simulations. Due to a different specificity of *notrump* vs. suit contracts, the point count methods and distributional estimators (when used alone) are more effective in the case of suit contracts than the *notrump* ones, which require some amount of roll-out simulations [18]. In this sense, the suit contracts are “better suited” for simulation-free estimations made by neural networks than the *notrump* contracts.

VI. SAMPLE DEALS

In this section five examples of deals from the testing set were chosen in order to illustrate strong and weak points of trained networks. For each deal also the fraction of correct answers given by top human players (Group-1) taking part in the experiment is provided. Please note that since the order of deals in the human contest was partly random,

TABLE III
RESULTS OF HUMAN PLAYERS VS. SELECTED NEURAL ARCHITECTURES

Type of the player	DDBP-4			DDBP-4			DDBP-2			DDBP-2		
	<i>Notrump</i> Contracts			<i>Spades</i> Contracts			<i>Notrump</i> Contracts			<i>Spades</i> Contracts		
Group-1	94.74	88.30	73.68	88.34	81.63	53.06	93.17	79.18	43.32	93.68	81.20	38.63
Group-2	92.94	84.71	60.78	93.87	82.95	48.66	84.00	69.71	34.86	88.46	73.59	30.59
(52x4)-(26x4)-26-13-1	96.89	83.64	37.31	99.88	96.48	53.11						
52-25-1							96.07	80.88	34.66	98.77	88.00	40.13

the numbers of answers for each of the deals were not pairwise equal. The first three deals (sections VI-A - VI-C) were chosen before the human contest was organized and hence the criteria of choosing them were mainly to illustrate the differences between neural architectures. The last two examples (presented in section VI-D) were chosen so as to show the possible advantage of using neural networks in this task. All human and networks answers concern the DDBP-4 version of the problem.

A. Only 3 Tricks in Defence Despite 25 WPC and two Singletons

♠ K 9	♠ A J 7 4
♥ A 9 5 3 2	♥ 10 8 7
♦ 7	♦ 10 9 8 3 2
♣ A 7 4 3 2	♣ 6
♠ Q 10 3 2	♠ 8 6 5
♥ 6 4	♥ K Q J
♦ A K J 6 5 4	♦ Q
♣ 9	♣ K Q J 10 8 5

N
W E
S

Correct number of tricks	3
Networks' estimations	
52 - 25 - 1	5
$(52x4) - (13x4) - 13 - 1$	5
$(52x4) - (26x4) - 26 - 13 - 1$	3
$(52x4 + 84) - (13x4 + 21) - 26 - 1$	4
Group-1 human players (% of correct answers)	80%

Fig. 3. The 1st sample deal. The estimations of a number of tricks to be taken by the *NS* pair in *spades* contract with *North* opening lead.

In the first example, presented in Fig. 3, *WE* can hold 10 tricks in *spades* contract, so *NS* are able to hold 3 tricks only. Most of the network architectures (not presented in this paper) claimed 5 tricks being wrong by 2 tricks. The $(52x4 + 84) - (13x4 + 21) - 26 - 1$ architecture overestimated 1 trick, and only the $(52x4) - (26x4) - 26 - 13 - 1$ one answered properly.

The power of *NS* pair (25 WPC including the *King* of trumps and two singletons) promises more than 3 tricks.

Closer analysis shows some weaknesses. $K♠$ is not able to take a trick because it is placed on the *N* hand and will be beaten by $A♠$ placed on the *E* hand. Also $Q♦$ cannot take a trick because it is a singleton. Twenty points in *hearts* and *clubs* can take only 3 tricks due to shortnesses on the opponents hands.

Humans managed to do quite well in this task - the correct answers were provided in 80% of the players.

B. The Number of Tricks Depending on which Hand Makes the Opening Lead

♠ J 7	♠ A 10 9 8 4	♠ 6 5 3 2
♥ Q J 8 4	♥ 9 2	♥ K 6 5 3
♦ K 8 4	♦ 9 6 5 2	♦ A 7
♣ J 10 6 5	♣ A K	♣ Q 3 2
	♠ K Q	
	♥ A 10 7	
	♦ Q J 10 3	
	♣ 9 8 7 4	

N
W E
S

Correct number of tricks	Opening lead	
	North	South
Networks' estimations	4	3
52 - 25 - 1	4	4
$(52x4) - (13x4) - 13 - 1$	3	4
$(52x4) - (26x4) - 26 - 13 - 1$	4	3
$(52x4 + 84) - (13x4 + 21) - 26 - 1$	3	4
Group-1 human players (% of correct answers)	80%	50%

Fig. 4. The 2nd sample deal. The estimations of a number of tricks to be taken by the *NS* pair in *spades* contract.

In the second example (presented in Fig. 4), the number of tricks to be taken by the *NS* pair in the *WE* contract in *spades* depends on which hand makes the opening lead. If it is *North*, the *NS* pair is able to hold 4 tricks (1 in $♠$, 2 in $♥$, and 1 in $♦$). When *S* makes defender's lead, *NS* pair can hold only 1 trick in $♥$, so 3 in total.

The $(52x4) - (26x4) - 26 - 13 - 1$ was the only network which correctly estimated the number of tricks in both cases.

The remaining architectures were mistaken in at least one of the cases. The case with the opening lead from S was generally more difficult.

The results among human players follow similar pattern. In the variant with opening lead from N , 80% of the players were correct, but in the case of the rotated opening lead (from the S side), the correct answer was given by only half of the players.

C. Undefended Grand Slam

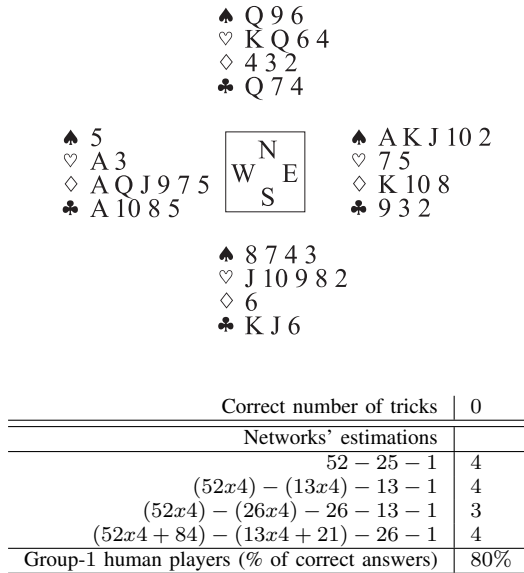


Fig. 5. The 3rd sample deal. The estimations of a number of tricks to be taken by the NS pair in *spades* contract with *North* opening lead.

The next example (shown in Fig. 5) is one of the deals for which all networks, regardless applied way of coding, made significant error. Perfectly fitted WE hands are able to hold all tricks (the grand slam) thanks to very favorable distribution of *spades* on NS hands. The strength of the NS pair is noticeable - 14 WPC, 7 trumps (including the *Queen*) and a singleton in *diamonds*, but still not enough to hold any trick in this deal.

Under these circumstances networks' estimations seem to be somehow "justified", but certainly wrong.

In this example the number of correct answers given by humans highly "diverged" between DDBP-4 (80%) and DDBP-2 (0.0%). The partly hidden variant of the deal appeared to be highly demanding. The closest answer was 3, which means three tricks error! Surprisingly, the problem was relatively easy in its fully uncovered version.

D. Advantage of networks' estimations

The following two examples were chosen among those that appeared to be relatively easy for neural networks (all tested architectures predicted correct numbers of tricks), but at the same time quite demanding for human players. In the first example, presented in Fig. 6 it is quite easy to point

out 8 tricks for NS pair (3 in ♠, 3 in ♥ and 2 in ♣), and "8 tricks" was the most frequent answer given by humans. The correct number of tricks is 9. Since the WPC estimation also suggests 8 tricks (the NS pair plays on 25 points) it is quite interesting that the networks were able to "find" this "missing" trick (in ♦).

The other example of networks' "supremacy" over humans is presented in Fig. 7, where the correct answer is 4 (1 trick in ♥, 2 in ♦ and 1 in ♣), but humans were largely inclined to estimate 5 tricks (additional one in ♣), possibly due to 16 WPC points on the NS hand.

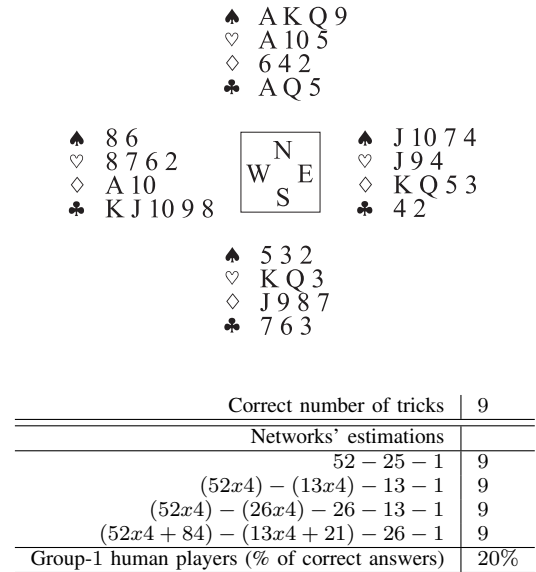


Fig. 6. The 4th sample deal. The estimations of a number of tricks to be taken by the NS pair in *spades* contract with *South* opening lead.

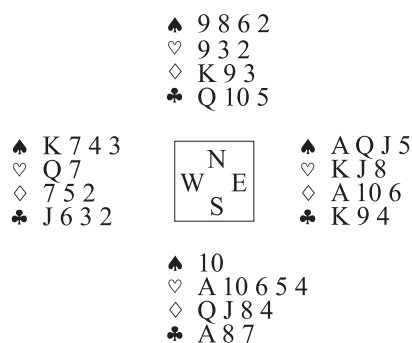
Based on the above analysis of example deals it can be concluded that even though the results accomplished by selected neural networks are comparable with those of humans (in case of *spades* contracts) there clearly exist deals in which human players are surpassed by neural networks and *vice versa*. It should be interesting to explore this issue in more detail.

VII. CONCLUSIONS

Artificial neural networks turned out to be very effective in estimating the number of tricks to be taken by one pair of players in the Double Dummy Bridge Problem.

The quality of attained results strongly depends on the way of coding a deal in the input layer. The best tested architectures were capable of discovering knowledge concerning the game based exclusively on sample training deals. The process of training was so effective that adding explicit human bridge knowledge, in the form of well-known human estimators of hands' strength, did not cause further improvement.

In several DDBP deals providing the proper answer is a really difficult task even for experience bridge players, despite revealing all cards. In some deals the answer depends



Correct number of tricks	4
Networks' estimations	
52 - 25 - 1	4
$(52x4) - (13x4) - 13 - 1$	4
$(52x4) - (26x4) - 26 - 13 - 1$	4
$(52x4 + 84) - (13x4 + 21) - 26 - 1$	4
Group-1 human players (% of correct answers)	25%

Fig. 7. The 5th sample deal. The estimations of a number of tricks to be taken by the NS pair in spades contract with West opening lead.

on the location of the defender's lead hand. Despite these difficulties, the most efficient neural network $((52x4) - (26x4) - 26 - 13 - 1)$ trained exclusively on example deals, without any human knowledge or awareness of nuances of the play (e.g. finesses), and with no information about the rules of the game, achieved respectful result: in suit contracts it was perfectly right in 53.11% of test deals and mistaken by more than one tricks in only 3.52% out of 100,000 test cases. The results for *notrump* contracts were equal to 37.80% and 16.36%, respectively.

The final assessment of the efficacy of proposed neural approach was made through a comparative experiment organized among professional bridge players. The top-10 of them were holding international titles (four Grand Masters, three International Masters and three Masters) and were playing in either the First or the Second Polish Bridge League. This selective group of players visibly outperformed neural network approach in case of *notrump* contracts, but accomplished comparable (or even slightly worse) result in the case of *suit* contracts. What is more, the results of neural networks were comparably close to those of human masters in the case of DDBP-2 in both *notrump* and *suit* contracts.

In the near future we plan to consider the problem of extraction of rules from the trained networks. Since neural networks appeared to be efficient in solving the DDBP and several human-type patterns were found in the networks' weights, it will be interesting to formally define and numerically quantify the bridge-specific features underlying their high performance. These extracted numerical features may be compared with human hand scoring systems and potentially lead to development of some new ideas in human bridge playing and be helpful for novice and semi-professional players in improving their bridge skills.

ACKNOWLEDGMENTS

The authors are grateful to Mr Piotr Dybicz the International Bridge Master for fruitful discussions regarding human vs. computer bridge playing styles and for his great help in organizing the human DDBP Contest.

REFERENCES

- [1] H. Francis, A. Truscott, and D. Francis, *The Official Encyclopedia of Bridge*, 5th ed. Memphis, TN: American Contract Bridge League Inc, 1994.
- [2] M. Köhle and F. Schönbauer, "Erfahrung mit einem Neurenalen Netz, das Bridge spielen lernt," in *Proceedings of the 5th Austrian Meeting on Artificial Intelligence*, J. Retti and K. Leidlmaier, Eds. Berlin: Springer-Verlag, 1989, pp. 224–229.
- [3] H. Yo, Z. Xianjun, Y. Yizheng, and L. Zhongrong, "Knowledge acquisition and reasoning based on neural networks – the research of a bridge bidding system," in *Proceedings of the INNC-90*, Paris, 1990, pp. 416–423.
- [4] M. Sarkar, B. Yegnanarayana, and D. Khemani, "Application of neural network in contract bridge bidding," in *Proc. of National Conf. on Neural Networks and Fuzzy Systems*, Anna University, Madras, 1995, pp. 144–151.
- [5] B. Yegnanarayana, D. Khemani, and M. Sarkar, "Neural networks for contract bridge bidding," *Sadhana*, vol. 21, no. 3, pp. 395–413, June 1996.
- [6] W. H. Root, *The ABCs of Bridge*. Three Rivers Press, 1998.
- [7] D. N. Levy, "The million pound bridge program," in *Heuristic Programming in Artificial Intelligence: The First Computer Olympiad*, D. Levy and D. Beal, Eds. Ellis Horwood, Chichester, 1989, pp. 95–103.
- [8] M. L. Ginsberg, "GIB: Imperfect information in a computationally challenging game." *Journal of Artificial Intelligence Research*, vol. 14, pp. 303–358, 2001. [Online]. Available: citeseer.ist.psu.edu/ginsberg01gib.html
- [9] B. Gambäck and M. Rayner, "Contract Bridge as a micro-world for reasoning about communication agents," Swedish Institute of Computer Science, Tech. Rep. SICS/R-90/9011, 1990. [Online]. Available: [ftp://sics.se/pub/SICS-reports/Reports/SICS-R-90-11-SE.ps.Z](http://sics.se/pub/SICS-reports/Reports/SICS-R-90-11-SE.ps.Z)
- [10] B. Gambäck, M. Rayner, and B. Pell, "Pragmatic reasoning in Bridge," University of Cambridge, Computer Laboratory, Tech. Rep. 299, April 1993. [Online]. Available: citeseer.ist.psu.edu/gamback93pragmatic.html
- [11] K. Mossakowski and J. Mańdziuk, "Artificial neural networks for solving double dummy bridge problems," in *Artificial Intelligence and Soft Computing - ICAISC 2004*, ser. Lecture Notes in Artificial Intelligence, L. Rutkowski, J. H. Siekmann, R. Tadeusiewicz, and L. A. Zadeh, Eds., vol. 3070. Springer, 2004, pp. 915–921.
- [12] —, "Neural networks and the estimation of hands strength in contract bridge," in *Artificial Intelligence and Soft Computing ICAISC 2006*, ser. Lecture Notes in Artificial Intelligence, L. Rutkowski et al., Eds., vol. 4029. Springer, 2006, pp. 1189–1198.
- [13] M. L. Ginsberg. Library of double-dummy results. [Online]. Available: <http://www.cirl.uoregon.edu/ginsberg/gibresearch.html>
- [14] K. Mossakowski and J. Mańdziuk, "Learning without human expertise. a case study of the Double Dummy Bridge Problem," *IEEE Transactions on Neural Networks*, vol. 20, no. 2, pp. 278–299, 2009.
- [15] J. Mańdziuk and K. Mossakowski, "Looking inside neural networks trained to solve double-dummy bridge problems," in *5th Game-On International Conference on Computer Games: Artificial Intelligence, Design and Education (CGAIDE 2004)*, Reading, UK, 2004, pp. 182–186.
- [16] K. Mossakowski and J. Mańdziuk, "Weight patterns in the networks trained to solve double dummy bridge problem," in *Issues in Intelligent Systems. Paradigms*, O. Hryniewicz et al., Eds. Exit, 2005, pp. 155–165.
- [17] J. Mańdziuk and K. Mossakowski, "Example-based estimation of hands strength in the game of bridge with or without using explicit human knowledge," in *Proceedings of the IEEE Symposium on Computational Intelligence in Data Mining (CIDM 2007)*. Honolulu, Hawaii, USA: IEEE Press, 2007, pp. 413–420.
- [18] P. Dybicz, International Bridge Master and trainer, *Private communication*, 2008.