And the winner is ... Chevalier de Borda: Neural networks vote according to Borda's Rule

Dávid Burka, Clemens Puppe, László Szepesváry and Attila Tasnádi

Abstract

We investigate whether neural networks are appropriate tools for selecting between prominent social choice functions. We find that neural networks can learn the unanimity principle and the Pareto property. Building on these two positive results, we train neural networks on the set of profiles possessing Condorcet winners, on the set of profiles possessing a unique Borda winner, and on the set of profiles possessing a unique plurality winner. We investigate which social outcome a neural network chooses if trained on the set of profiles possessing unique winners according to one or more social choice functions. We compare the choices obtained by trained neural networks with those chosen by the Borda count, the Copeland method, the Kemény-Young method, the plurality rule, and 2-approval voting. We find that the trained neural networks' behavior is the closest to the Borda rule, second closest to the Condorcet-consistent methods, and clearly, the furthest away from the plurality rule. By this approach we hope to give new insight on the problem of selecting the appropriate voting rule.

Approaches to single out a voting rule by investigating its properties failed so far, and there is no hope that the axiomatic approach so prominent in social choice theory would give us the ultimate answer. Since the impossibility theorems of Arrow (1951) and Gibbard-Satterthwaite (1973/1975) it is well known that there is no voting rule satisfying a small set of very reasonable axioms. Though there are some possibility results on some restricted domains, especially on single peaked-domains (Moulin, 1980) or their generalizations (see Barberà et al., 1993 and Nehring and Puppe, 2007 for the most general one), the problem of choosing the right voting method appears to be an evergreen problem.[1] There are various axiomatic characterizations of the Borda count (e.g. Smith, 1973), the Copeland method (e.g. Henriet, 1985), approval voting (e.g. Fishburn, 1978), the Kemény-Young method (e.g. Young and Levenglick, 1978), and of other methods. On one hand different meaningful characterizations support different voting rules, while on the other hand different voting rules exhibit different voting paradoxes and all of them (with the exception of the dictatorial one) suffer from strategic voting under the assumptions of the Gibbard-Satterthwaite theorem. Furthermore, neither the operations research type approach, by picking an objective function above the set of voting rules, turns out to be helpful since any voting rule can be obtained as a solution of an appropriately chosen optimization problem (e.g. Elkind, Faliszewski and Slinko, 2015).

In this paper we consider a different approach: It would be interesting to know which social outcomes humans would choose when confronted with an arbitrarily given voters' preference profile. One can expect that this task can be solved 'consistently' by a human only for a small set of alternatives and a small number of voters. Investigating the choice of individuals in function of a given preference profile in an experimental setting,[2] has its difficulties because it is not clear how the participants should be paid in function of

[1] For an overview on possibility results on restricted domains we refer to Gaertner (2002).

[2] For instance, Giritligil Kara and Sertel (2005) investigated in an experimental setting the connection between the Majoritarian Compromise-winner and the Borda-winner as well as the Condorcet-winner.

their choices, i.e. to incentivize them appropriately, which does not pose a problem when employing neural networks. However, instead of pursuing the experimental route, we replace humans by artificial neural networks, which have been successfully applied to problems like computer vision, handwriting recognition, and speech recognition,[3] mimicking human behavior. In particular, we will train the Multi-Layer Perceptron, henceforth MLP, by Rumelhart et al. (1986) on the set of profiles having Condorcet winners, unique Borda winners, and unique plurality winners (as well as in several combinations of these winners) and compare statistically the chosen outcomes by the trained MLP. We pose questions like how similar the voting rule specified by a trained MLP and some prominent voting rules are.

Neural networks have been widely used for forecasting and classification by econometricians, while there are only far less applications in economic theory. MLPs have been applied in economic theory for problems like investigating bidding behavior by Dorsey et al. (1994), addressing market entry by Leshno et al. (2002), and learning game playing rules by Sgroi and Zizzo (2009). Recently, a combination of neural networks have been successfully employed by Silver et al. (2016) to defeat the human European Go champion by 5 games to 0. It is worthwhile mentioning that Richards et al. (2006) employed in an opposite way some voting rules in the construction of new learning algorithms for winner-takes-all neural networks.

Considering the cases of three, four and five alternatives combined with the number of voters ranging from 7 to 11, we find that trained MLPs behavior is the closest to the Borda rule, second closest to the Condorcet consistent methods of Copeland and Kemény-Young, and differs the most from the plurality rule.

The plan of the paper is as follows. In Section 2 we introduce our framework, Section 3 describes our data generation process, Section 4 describes the results on the two axioms, Section 5 compares trained neural networks with social choice functions, and finally Section 6 concludes.

1 Basic Notation, Definitions and Notions

1.1 Voting rules

Let X be a finite set of alternatives with cardinality q, where when employing neural networks we will assume $q = 3$, $q = 4$ or $q = 5$. By $\mathcal{P}$, we denote the set of all linear orderings (irreflexive, transitive and total binary relations) on X. In addition, by $\mathcal{R}$ we denote the set of all weak orderings on X. Let $rk[x, \succ]$ denote the *rank* of alternative x in the ordering $\succ \in \mathcal{P}$ (i.e. $rk[x, \succ] = 1$ if x is the top alternative in the ranking $\succ$, $rk[x, \succ] = 2$ if x is second-best, and so on). We shall denote by $N = \{1, \ldots, n\}$ the set of voters.

Definition 1 (Voting rule)**.** A mapping $F : \mathcal{P}^n \to 2^X$ that selects the set of winning alternatives is called a *voting rule.*

Note that our definition of a voting rule allows for possible ties, since we do not want to deal with tie-breaking rules, which would unnecessarily complicate the description of our framework without additional gains.

We define the two basic properties that every sensible voting rule has to satisfy.

Definition 2. Voting rule F is *unanimous* if for all $(\succ_1, \ldots, \succ_n) \in \mathcal{P}^n$

$$(\exists x \in X : \forall i \in N : rk[x, \succ_i] = 1) \implies F(\succ_1, \ldots, \succ_n) = \{x\}.$$

[3]Humans are good in solving these types of problems, while they are hard to solve by the means of standard rule-base programing.

Unanimity requires that in every profile in which everybody ranks the same alternative on the top, this top alternative has to be chosen by the voting rule.

Definition 3. Voting rule F satisfies the *Pareto property* if for all $(\succ_1, \ldots, \succ_n) \in \mathcal{P}^n$

$$\forall x, y \in X : (\forall i \in N : x \succ_i y) \implies y \notin F(\succ_1, \ldots, \succ_n).$$

We have stated the Pareto property in a modified form by requiring that if there is an alternative x which is better than y for everybody, then y is dominated by x, and therefore y should not be chosen by a voting rule.

Now we turn to the definition the six voting rules we will investigate. We shall denote the *Borda score* of alternative $x \in X$ according to ordering $\succ$ by $bs\,[x, \succ] = q - rk[x, \succ]$.

Definition 4. A voting rule *Borda* is the *Borda count* if for all $(\succ_i)_{i=1}^n \in \mathcal{P}^n$

$$Borda\,((\succ_i)_{i=1}^n) = \arg\max_{x \in X} \sum_{i=1}^n bs[x, \succ_i].$$

For a given profile $(\succ_i)_{i=1}^n \in \mathcal{P}^n$ we say that alternative $x \in X$ *beats* alternative $y \in X$ if $\#\{i \in N \mid x \succ_i y\} > \#\{i \in N \mid y \succ_i x\}$, i.e. x wins over y by pairwise comparison. We shall denote by $l[x, (\succ_i)_{i=1}^n]$ the number of alternatives beaten by alternative $x \in X$ for given profile $(\succ_i)_{i=1}^n$.

Definition 5. A voting rule *Cop* is the *Copeland method* if for all $(\succ_i)_{i=1}^n \in \mathcal{P}^n$

$$Cop\,((\succ_i)_{i=1}^n) = \arg\max_{x \in X}\; l[x, (\succ_i)_{i=1}^n].$$

In order to define the Kemény-Young method for a given preference profile $(\succ_i)_{i=1}^n \in \mathcal{P}^n$ we shall denote by $v(x, y, (\succ_i)_{i=1}^n)$ the number of voters who prefer $x \in X$ to $y \in X$. For a given preference profile $(\succ_i)_{i=1}^n \in \mathcal{P}^n$ we define $\mathcal{D}_{KY}\,((\succ_i)_{i=1}^n) \subseteq \mathcal{P}$ as follows:

$$\mathcal{D}_{KY}\,((\succ_i)_{i=1}^n) = \arg\max_{\succ \in \mathcal{P}} \sum_{\{x, y \in X, x \succ y\}} v(x, y, (\succ_i)_{i=1}^n)).$$

Definition 6. A voting rule *Kem-You* is the *Kemény-Young method if it chooses the top ranked alternative(s) from the set of linear orderings in $\mathcal{D}_{KY}$, i.e*

$$x \in Kem{-}You\,((\succ_i)_{i=1}^n) \iff (rk[x, \succ] = 1 \text{ for some } \succ \in \mathcal{D}_{KY})\,.$$

We also investigate the plurality rule.

Definition 7. A voting rule *Plu* is the *plurality rule if for all* $(\succ_i)_{i=1}^n \in \mathcal{P}^n$

$$Plu\,((\succ_i)_{i=1}^n) = \arg\max_{x \in X}\; \#\,\{i \in N \mid rk[x, \succ_i] = 1\}\,.$$

Next we define k-approval voting.

Definition 8. A voting rule $k{-}AV$ is the *k-approval voting rule if for all* $(\succ_i)_{i=1}^n \in \mathcal{P}^n$

$$k{-}AV\,((\succ_i)_{i=1}^n) = \arg\max_{x \in X}\; \#\,\{i \in N \mid rk[x, \succ_i] \leq k\}\,.$$

Finally, we also consider plurality with runoff.

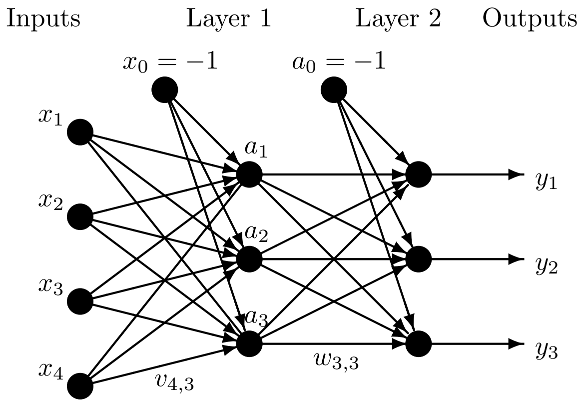

Figure 1: MLP

Definition 9. A voting rule $Plu{-}run$ is the *plurality with runoff rule* if for all $(\succ_i)_{i=1}^n \in \mathcal{P}^n$ in the first round only the top two alternatives based on the number of top positions compete in a second round. If there were more than two alternatives tied on the top, we try to reduce ourselves to top two alternatives by sequentially considering lower ranked positions.[4] If there is a unique plurality winner, then in case of ties on the second place ties have to be broken by only considering the tied alternatives. We determine in the second round the winner of the two alternatives by restricting the preferences of the voters to these two alternatives and the alternative beating the other in case of more voters becomes the final winner.[5]

1.2 A brief description of the MLP

We will consider a MLP with two layers. We shall denote by m, p and r the number of inputs, hidden neurons and output neurons, respectively. An illustration of a two layered perceptron can be found in Figure 1. The weight matrices $V \in \mathbb{R}^{(m+1)\times p}$ and $W \in \mathbb{R}^{(p+1)\times r}$ will be determined by the backpropagation algorithm of Rumelhart et al. (1986). Then a trained two-layered preceptron has gathered its knowledge in $\mathbf{V}$ and $\mathbf{W}$ from the training set, which will be in our case the set of those profiles with some common winners together with the respective common winner. Then for profiles without a winner we can obtain the choice of the trained neural network by determining the activation level

$$h_j := \sum_{i=0}^{m} v_{ij} x_i, \quad a_j := g(h_j) = \frac{1}{1 + e^{-\beta h_j}}, \tag{1.1}$$

for each hidden neuron, and thereafter, by determining the activation level

$$o_k := \sum_{j=0}^{p} w_{jk} a_j, \quad y_k := g(h_k) = \frac{1}{1 + e^{-\beta h_k}}; \tag{1.2}$$

for each output neuron. For more on neural networks we refer to Haykin (1999) and Marshland (2009).

[4] We omit here a specification of a tie-breaking rule since we will only employ the rule for a small number of alternatives.

[5] Ties do not emerge at this stage since we will investigate only cases with an odd number of voters.

2 Data generation

We consider cases in which the number of voters equals 7, 9 or 11, while the number of alternatives equals 3, 4 or 5. In case of 3 alternatives each individual can have 6 different linear orderings, which results in $6^7 = 279936$ profiles for 7 voters if we neglect symmetries. In case of 3 alternatives 1000 profiles and 5 neurons proved to be sufficient for learning. Moreover, these parameter values were also for 4 and 5 alternatives sufficient for verifying unanimity. For the case of 4 alternatives and 11 voters the number of profiles increases to $24^{11} = 1521681143169024 \approx 1.52 * 10^{15}$. Therefore, in order to keep the problem computationally tractable we took 3000 profiles and 5 neurons and 10000 profiles and 15 neurons for the cases of 4 and 5 alternatives, while training the MLP for a certain subset of winners. While verifying the Pareto property, we took 10000 profiles and 15 neurons in case of 4 alternatives and 20000 profiles and 30 neurons in case of 5 alternatives.

We code preference relations in the following way: Let $X = \{x_1, x_2, \dots, x_q\}$. If $x_{i_1} \succ x_{i_2} \succ \dots \succ x_{i_q}$, where $(i_1, \dots, i_q)$ is a permutation of $(1, \dots, q)$, then we store the respective pairwise comparisons in a vector corresponding to an upper triangular matrix $(a_{jk})_{j=1,k=j+1}^{q,q}$ in which $a_{jk} = 1$ in case of $x_j \succ x_k$ and $a_{jk} = 0$ otherwise. For example, $x_1 \succ x_4 \succ x_2 \succ x_3$ is coded by $(1, 1, 1, 1, 0, 0)$ obtained as the outcomes of the comparisons $x_1 \succ x_2$, $x_1 \succ x_3$, $x_1 \succ x_4$, $x_2 \succ x_3$, $x_2 \succ x_4$, $x_3 \succ x_4$. A profile is given by a row vector with $nq(q-1)/2$ entries. To arrive to a preference profile we generate the impartial culture (IC) in which the selection of each preference profile is equally like.[6]

To complete an input of a training set we have to specify an alternative randomly according to the uniform distribution, for which we use the so-called 1-of-N encoding, i.e. if alternative x_i is the respective target value, then it is represented by the indicator vector $(0, \dots, 0, 1, 0, \dots, 0)$, where only the ith coordinate equals 1.

When testing unanimity, we have picked an alternative, which served as the top alternative of each individual, thereafter we have randomly ordered the alternatives below the top alternative. The target value for a profile was its unanimous 'top alternative.'

In case of the Pareto property we have picked randomly two alternatives x and y for each profile, and then we have chosen for each voter independently the position of x randomly from the top $q-1$ positions, and the position of y randomly below x. This assured that there will be a Pareto dominated alternative in our profile. Thereafter, we have distributed the remaining $q-2$ alternatives randomly on the set of unfilled positions for each voter. If no other dominance relationship emerged, we took y as the target value for the generated profile. However, since other dominance relationships could emerge in the generated profile, we only kept a generated profile in our sample if there was a most dominated alternative, i.e. an alternative that was dominated by any other alternative involved in a dominance relationship. In that case we took the most dominated alternative as the target value, since this one definitely should not be chosen by a voting rule.

When training for winners or a set of winners, we considered five possibilities. First, we only trained on the set of profiles with Condorcet winners from the randomly generated 1000, 3000, and 10000 profiles in case of 3, 4, and 5 alternatives, respectively; second, we trained on the set of profiles with unique Borda winners; third we trained on the set of profiles with unique plurality winners; forth, we trained on the set of profiles on which the Condorcet winner was equal to the unique Borda winner; and fifth, we trained on the set of profiles on which the Condorcet winner was equal to the unique Borda winner and also equal

[6] We have checked that our results are robust with respect to generating the anonymous impartial culture (AIC) in which each anonymous preference profile is equally likely and for employing an encoding by pairwise comparison matrices. All results and the program can be found at https://github.com/dburka001/SocialChoiceNeuralNetwork. For more on the IC and AIC cultures we refer the reader to Eğecioğlu and Giritligil (2013).

to the unique plurality winner. The generation of profiles and of the training set was written in C#, thereafter we employed Marshland's (2009) MLP Python class to train our neural network and to 'predict' the winning alternatives for the profiles from our generated sample. The prediction was carried out by a trained neural network on an independent new sample (called simply an *input sample*) from which we have only taken the first 1000 profiles into account even if the generated set of profiles was larger (i.e. either 3000 or 10000). Finally, the statistical evaluation was carried out in Excel.

For each combination of alternatives and voters, as well as for the five investigated set of winners (altogether 45 cases), we have generated five random training seeds for generating the training samples and taken five random neuron seeds for the training procedure of the MLP. In particular, an alternative for a profile was selected by the five MLPs trained on the same training sample (i.e. generated by the same training seed) if the same alternative was chosen by at least 3 neuron seeds out of the 5 possible neuron seeds (i.e. selected by the plurality of trained MLPs with the same training seed). In this way we have determined the learning rates for the five training seeds, and thereafter we have calculated the average learning rates for any investigated combination of alternatives, voters, and set of winners. The results shown in the tables in Section 4 are the respective averages of the 5 random training seeds employed for generating the training samples. Altogether we have evaluated and aggregated 25 results for each of the 45 cases.

3 Testing whether MLP can learn our two basic properties

In case of unanimity the situation was quite straightforward: For 3, 4 and 5 alternatives and 7, 9, and 11 voters each trained neural network could select the unanimous top alternative with 100% accuracy for its newly generated input sample.

Continuing with the Pareto property, in case of 3 alternatives the trained MLPs could learn the Pareto dominated alternative on average with 99.54%, 99.66%, and 99.56% accuracy in case of 7, 9, and 11 voters, respectively, which can be considered as very high. However, in case of 4 alternatives the Pareto dominated alternative was learned with 93.90%, 95.10%, and 95.68% accuracy in case of 7, 9, and 11 voters, respectively. To achieve this result we had to increase our training sample to 10000 and the number of hidden neurons to 15. The results became less satisfactory for the case of 5 alternatives for which we obtained respective learning ratios of 61.74%, 68.18%, and 73.24% even for training samples of size 20000 and 15 hidden neurons. By increasing the number of hidden neurons to 30 we could increase the respective learning rates to 67.60%, 79.20%, and 84.68%. However, the very long training time even on a stronger computational device makes it difficult to experiment with different numbers of hidden neurons and larger samples. Clearly, the training sample size of 20000 profiles is extremely small compared with the huge number of all possible profiles having a Pareto dominated alternative in case of 5 alternatives (about 55.99 millions, 2.02 billions, 72.56 billions in case of 7, 9, and 11 voters, respectively, combined with a high input space dimension of 110 in the latter case). Consuming a week of computation time on a stronger machine, we determined the learning rates for five training sets of size 50000 for at least one network seed and found that the average learning rate increased to 88.78% (when taking 5 alternatives, 11 voters, and 30 hidden neurons). Therefore, we conjecture that the Pareto property for 5 alternatives can still be learned by the MLP, but this lies beyond our computational capacities.

To summarize, the MLP passed the unanimity test with 100% and the Pareto dominance test at a high level of accuracy for 3 and 4 alternatives. Furthermore, the unanimity test was passed at 100% accuracy and we conjecture that the Pareto dominance test can be also passed for 5 alternatives at a very high level of accuracy if the training sample size is sufficiently large (e.g. 100000). Based on these tests we consider neural networks as a useful tool for the evaluation of social choice functions.

4 Comparing MLP choices with the choices of several voting rules

First, we consider the case when we took profiles with Condorcet winners as our training sample. For the case of 3, 4, and 5 alternatives and 7, 9, and 11 voters the main results appear in Table 1. The table entries contain the average percentages of those cases in which a trained MLP selects a winner of the method appearing in the respective column heading. Focusing on the cases of 4 and 5 alternatives, surprisingly, though trained on the set of Condorcet winners, the Borda count 'outperforms' (in the sense of being more similar to the trained MLPs choice) the other methods. From the two investigated Condorcet consistent methods the Copeland method lies closer to the MLPs choice than the Kemény-Young method. The plurality and the plurality with runoff rules are clearly the furthest away from the MLP's choices. 2-approval voting is coming, compared with the other rules, relatively closer to the choices of the trained MLP as the number of alternatives is increasing. By considering the differences in percentages one should keep in mind that on many profiles the methods all agree so the relative differences on the profiles on which these methods disagree become even larger. In case of 3 alternatives the differences are smaller, only the plurality rule differs significantly more from the MLP. There is a separating line between the cases of 3, 4, and 5 alternatives in order to emphasize that we have employed an increasing training set size in the number of alternatives. It is worthwhile mentioning that the percentages in

Method	Cop	Kem-You	Borda	Plu	2-AV	Plu-run
q=3, n=7	98.42%	98.42%	98.19%	90.03%	83.61%	92.38%
q=3, n=9	96.34%	96.24%	97.56%	89.16%	81.17%	89.84%
q=3, n=11	91.36%	91.08%	96.81%	87.08%	81.99%	84.08%
q=4, n=7	91.83%	89.24%	96.96%	82.18%	87.32%	80.24%
q=4, n=9	90.30%	87.17%	96.01%	79.08%	85.46%	78.85%
q=4, n=11	88.49%	85.57%	93.62%	78.27%	84.32%	77.60%
q=5, n=7	88.73%	84.84%	97.18%	76.28%	85.75%	71.43%
q=5, n=9	88.03%	83.32%	95.10%	70.46%	82.58%	69.79%
q=5, n=11	87.62%	83.30%	93.69%	69.44%	81.11%	70.35%

Table 1: Trained on Condorcet winners

all Tables 1-5 are decreasing in the number of voters. However, this does not necessarily mean that the MLP learns voting rules in case of more alternatives and more voters with a lower percentage. Since we have an increase in the number of MLP inputs from 21 to 66 as we move from the $q = 3$, $n = 7$ case to the $q = 4$, $n = 11$ case, this would have required an increase in the sample size to partially offset the increase of the dimension of the input space. We can observe this when we look at the results for 5 alternatives. These differences in sample sizes do not affect our main findings.

We obtain basically the same ordering of the methods concerning their 'distances' from the trained MLPs if we take the profiles with unique Borda winners as the training set (see Table 2). Not surprisingly the MLP comes even closer to the choices of the Borda count since it is now intended to learn the Borda count. Once again the Kemény-Young method is further away from the MLP's choices than the Copeland method. The difference becomes again significant for 4 and 5 alternatives. The plurality rule is somewhat closer to the MLP's choices than plurality with runoff. 2-approval voting performs relatively better for a larger number of alternatives.

Method	Cop	Kem-You	Borda	Plu	2-AV	Plu-run
$q = 3, n = 7$	92.60%	92.60%	99.49%	87.56%	86.52%	86.10%
$q = 3, n = 9$	91.96%	91.92%	99.17%	87.74%	83.60%	84.73%
$q = 3, n = 11$	91.38%	91.22%	98.32%	87.79%	81.92%	83.89%
$q = 4, n = 7$	90.03%	86.94%	98.75%	81.40%	88.24%	78.08%
$q = 4, n = 9$	88.81%	85.65%	97.92%	78.28%	86.73%	76.75%
$q = 4, n = 11$	89.15%	86.03%	96.92%	78.18%	84.92%	77.63%
$q = 5, n = 7$	88.55%	84.44%	98.39%	75.10%	85.74%	70.50%
$q = 5, n = 9$	88.28%	83.48%	97.15%	70.41%	82.93%	69.43%
$q = 5, n = 11$	87.08%	82.65%	95.69%	68.62%	81.66%	69.62%

Table 2: Trained on Borda winners

Again, when taking the profiles with unique plurality winners as the training set (see Table 3), we obtain basically the same ordering of the methods concerning their 'distances' from the trained MLP. Now, surprisingly the MLP does not really learn the plurality rule and remains closer to the other methods, especially to the Borda count. If we compare only the columns for the plurality rule in Tables 1-3, surprisingly we can observe that for 3 alternatives the MLP comes closer to the plurality rule when not trained on the profiles with a unique plurality winner.

Method	Cop	Kem-You	Borda	Plu	2-AV	Plu-run
$q = 3, n = 7$	91.80%	91.78%	97.35%	87.38%	86.13%	85.98%
$q = 3, n = 9$	90.08%	89.92%	95.96%	87.33%	83.29%	84.25%
$q = 3, n = 11$	88.88%	88.32%	93.28%	85.76%	81.19%	82.15%
$q = 4, n = 7$	89.93%	87.67%	96.41%	82.31%	88.55%	79.53%
$q = 4, n = 9$	87.73%	85.11%	93.31%	79.32%	85.98%	77.96%
$q = 4, n = 11$	86.73%	84.12%	91.12%	78.89%	84.57%	77.75%
$q = 5, n = 7$	89.42%	86.17%	97.06%	78.52%	87.30%	74.22%
$q = 5, n = 9$	89.75%	86.22%	95.81%	75.17%	86.81%	75.51%
$q = 5, n = 11$	89.32%	85.94%	94.44%	72.29%	85.40%	74.10%

Table 3: Trained on plurality winners

Giving the Condorcet consistent methods and the Borda count the same chance of being learned, we take as the training sample the set of those profiles with identical Condorcet and unique Borda winners. As it can be seen from Table 4 the ordering of the methods did not change, and the Borda count is reinforced in some sense. The gap between the Copeland and the Kemény-Young methods remains approximately the same. The same observation holds true for the plurality rule, 2-approval voting, and plurality with runoff.

Continuing our analysis, we add the requirement of being even a unique plurality winner to the requirements of being a Condorcet and unique Borda winner at the same time. Table 5 strengthens our previous observations. Compared with Table 4 almost all percentages are slightly smaller.

For the cases in which we trained the MLP on the set of profiles having a Condorcet winner or having a unique Borda winner, we investigated the trained MLPs' choices on the

Method	Cop	Kem-You	Borda	Plu	2-AV	Plu-run
$q = 3, n = 7$	93.04%	92.98%	98.02%	86.92%	85.28%	86.00%
$q = 3, n = 9$	92.19%	91.89%	97.20%	86.63%	82.73%	85.26%
$q = 3, n = 11$	90.97%	90.39%	95.67%	86.52%	81.42%	83.60%
$q = 4, n = 7$	90.21%	87.33%	97.27%	81.12%	87.89%	78.53%
$q = 4, n = 9$	88.17%	84.84%	95.26%	78.15%	85.46%	76.53%
$q = 4, n = 11$	87.52%	84.34%	92.82%	77.63%	83.45%	76.41%
$q = 5, n = 7$	87.09%	83.12%	95.62%	74.66%	84.81%	69.65%
$q = 5, n = 9$	87.26%	82.56%	94.20%	69.83%	81.69%	68.89%
$q = 5, n = 11$	85.77%	81.41%	92.49%	68.51%	80.55%	68.89%

Table 4: Trained on Borda and Condorcet winners

Method	Cop	Kem-You	Borda	Plu	2-AV	Plu-run
$q = 3, n = 7$	92.61%	92.49%	97.32%	86.71%	85.26%	85.49%
$q = 3, n = 9$	91.37%	91.01%	96.15%	86.62%	82.39%	84.65%
$q = 3, n = 11$	88.80%	87.82%	92.60%	85.26%	79.99%	81.91%
$q = 4, n = 7$	88.59%	85.68%	94.78%	80.35%	87.26%	77.63%
$q = 4, n = 9$	86.36%	83.24%	91.72%	76.89%	84.18%	74.82%
$q = 4, n = 11$	84.90%	81.56%	88.98%	76.55%	82.31%	74.43%
$q = 5, n = 7$	86.14%	82.27%	93.03%	74.19%	83.93%	69.40%
$q = 5, n = 9$	85.76%	81.12%	92.01%	69.35%	81.10%	68.04%
$q = 5, n = 11$	84.48%	80.27%	90.08%	67.95%	80.04%	68.30%

Table 5: Trained on Borda, Condorcet and plurality winners

set of those profiles on which the Condorcet winner differs from the unique Borda winner. Out of the 1000 cases we had about 20 to 60 profiles of this kind in function of q and n. Though the number of such profiles is small our results are meaningful because we have investigated all the 25 generated input files with 1000 profiles for each $q \in \{3, 4\}$ and $n \in \{7, 9, 11\}$. Table 6 shows our results when we trained the MLPs on the set of profiles with Condorcet winners. We observe that on the subset of these profiles the MLP comes closer to the Borda count compared with most of the other methods if the number of voters or the number of alternatives is larger. However, MLP comes slightly closer to 2-AV than to Borda for 5 out of the 6 combinations of q and n. Plurality has higher percentages than the two investigated Condorcet consistent methods.

Method	Cop	Kem-You	Borda	Plu	2-AV	Plu-run
$q = 3, n = 7$	76.76%	76.76%	23.24%	85.16%	23.24%	76.76%
$q = 3, n = 9$	51.06%	51.06%	48.94%	71.83%	58.58%	51.06%
$q = 3, n = 11$	18.71%	18.71%	81.29%	64.32%	87.14%	19.85%
$q = 4, n = 7$	22.75%	22.75%	77.25%	58.47%	76.64%	24.42%
$q = 4, n = 9$	22.40%	22.40%	76.08%	58.01%	77.51%	25.86%
$q = 4, n = 11$	28.83%	28.83%	67.74%	60.64%	73.01%	33.08%

Table 6: Condorcet winner differs from unique Borda winner and trained on Condorcet winners

Table 7 contains our results when we have trained the MLPs on the set of profiles with unique Borda winners. On the subset of these profiles the MLP comes much closer to the Borda count compared with most of the other methods. In general we can say that the investigated scoring methods Borda, 2-AV and plurality are far closer to the MLPs' choices than those ones by our two Condorcet consistent methods. The values for the Borda count and 2-AV are quite close.

Method	Cop	Kem-You	Borda	Plu	2-AV	Plu-run
$q = 3, n = 7$	1.18%	1.18%	98.82%	50.90%	98.82%	1.18%
$q = 3, n = 9$	5.17%	5.17%	94.83%	58.73%	95.83%	5.17%
$q = 3, n = 11$	5.37%	5.37%	94.63%	66.64%	92.50%	7.62%
$q = 4, n = 7$	13.32%	13.32%	86.68%	55.39%	82.29%	16.77%
$q = 4, n = 9$	18.76%	18.76%	80.89%	58.02%	79.44%	24.06%
$q = 4, n = 11$	26.27%	26.27%	71.68%	61.05%	74.17%	30.22%

Table 7: Condorcet winner differs from unique Borda winner and trained on unique Borda winners

So far we have investigated whether the trained neural networks had chosen a winning alternative of another method. However, the investigated voting methods may differ in how often they select a unique winner, which might cause a bias. In Table 8 we gathered the averages on how frequently a method does not have a unique winner. We can observe that for more alternatives and also for more voters the bias in favor of the Borda count becomes less and less important. It is worthwhile mentioning that the plurality rule could not benefit from its significantly highest percentage of producing ties concerning the results in Tables 1-5. Nevertheless we also analyze the voting rules on the subset of profiles on which they are selecting a unique winner.

Method	Cop	Kem-You	Borda	Plu	2-AV	Plu-run
$q = 3, n = 7$	7.20%	6.84%	13.44%	18.84%	35.26%	1.96%
$q = 3, n = 9$	8.24%	7.32%	12.76%	18.12%	21.44%	7.72%
$q = 3, n = 11$	8.26%	6.84%	10.82%	24.52%	19.18%	0.88%
$q = 4, n = 7$	14.08%	14.38%	14.36%	24.96%	31.48%	9.82%
$q = 4, n = 9$	14.74%	13.90%	13.02%	27.38%	27.92%	2.74%
$q = 4, n = 11$	15.32%	14.72%	10.92%	22.30%	25.00%	6.42%
$q = 5, n = 7$	18.10%	22.42%	12.12%	37.78%	30.98%	10.14%
$q = 5, n = 9$	19.70%	22.82%	11.44%	27.30%	27.68%	5.80%
$q = 5, n = 11$	19.18%	21.10%	10.50%	32,08%	26.16%	6.60%

Table 8: How frequently does a method not have a unique winner?

It is worthwhile mentioning that the MLP itself does not always give us a unique answer, that is for some profiles the trained MLP is not able to choose a single alternative. Concerning that the rules we used to train the MLP themselves do not give us a unique answer in many cases, it is not surprising that the MLP might be indifferent between alternatives. Comparing Table 8 with Table 9 we can see that the trained MLP can give us for far more profiles a decisive answer than the respective voting rules. However, it is worthwhile mentioning that in the combined cases the trained MLP was less decisive, probably because there was no unique rule to be learned.

Trained on:	CW	BW	PL	CW+BW	CW+BC+PL
$q = 3, n = 7$	1.52%	1.34%	4.16%	0.28%	0.68%
$q = 3, n = 9$	1,52%	0.98%	4.60%	0.66%	1.22%
$q = 3, n = 11$	2.06%	0.88%	3.92%	1.18%	2.16%
$q = 4, n = 7$	3.36%	2.72%	11.44%	2.74%	3.90%
$q = 4, n = 9$	4.28%	3.12%	14.52%	3.42%	4.28%
$q = 4, n = 11$	4.12%	2.56%	16.86%	3.08%	4.92%
$q = 5, n = 7$	5.52%	3.22%	17.60%	3.64%	4.46%
$q = 5, n = 9$	5.28%	3.74%	26.84%	3.80%	4.66%
$q = 5, n = 11$	5.52%	3.06%	26.98%	3.58%	5.40%

Table 9: How frequently does MLP not have a unique winner?

5 Concluding remarks

If we interpret learning by neural networks also as a kind of complexity selection device, we could state that plurality is far too simple, and the two investigated Condorcet methods are too sophisticated. Or put it otherwise, when picking the winner, it makes sense to take more information into account than looking only at the top ranked alternatives, but it also makes sense not to carry out too complicated calculations. The learning percentages are also in line with the fact that the Kemény-Young method is more sophisticated than Copeland's method. We also observed that the plurality rule achieves higher percentages than plurality with runoff. Furthermore, concerning the percentages 2 approval voting is somewhere between the Condorcet consistent methods and the plurality rule. Our results show that on any training set the Borda count was learned far more closely and that the trained MLPs resemble the Borda count. Based on these results we conjecture that humans not trained on the theory of social choice would aggregate preferences to a social alternative most frequently in line with the Borda count. However, this has to be confirmed by well-designed human experiments in computer labs.

References

[1] K. J. Arrow. *Social Choice and Individual Values.* Wiley, 1951.

[2] S. Barberà, F. Gul, and E. Stacchetti. Generalized Median Voter Schemes and Committees. *Journal of Economic Theory*, 61(2):262–289, 1993.

[3] D. Black. *The Theory of Committees and Elections.* Cambridge University Press, 1958.

[4] R.E. Dorsey, J.D. Johnson, and M.V. Van Boening. The use of artificial neural networks for estimation of decision surfaces in first price sealed bid auctions. In W.W. Cooper and A.B. Whinston, editors, *New decisions in computational economics*, 19–40. Kluwer Academic Publishing, 1994.

[5] Ö. Eğecioğlu and A.E. Giritligil. The Impartial, Anonymous, and Neutral Culture Model: A Probability Model for Sampling Public Preference Structures. *Journal of Mathematical Sociology*, 37(4):203–222, 2013.

[6] E. Elkind, P. Faliszewski, and A. Slinko. Distance rationalization of voting rules. *Social Choice and Welfare*, 45(2):345–377, 2015.

[7] P.C. Fishburn. Axioms for approval voting: direct proof. *Journal of Economic Theory*, 19(1):180–185, 1978, corrigendum 45(1):212, 1988.

[8] W. Gaertner. *Domain Conditions in Social Choice Theory.* Cambridge University Press, 2002.

[9] A. Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41(4):587–601, 1973.

[10] A.E. Giritligil Kara and M.R. Sertel. Does majoritarian approval matter in selecting a social choice rule? An exploratory panel study. *Social Choice and Welfare* 25(1):43–73, 2005.

[11] S. Haykin. *Neural Networks: A Comprehensive Foundation*, 2nd Edition. Prentice-Hall, 1999.

[12] D. Henriet. The Copeland choice function: an axiomatic characterization. *Social Choice and Welfare*, 2(1):49–63, 1985.

[13] M. Leshno, D. Moller, and P. Ein-Dor. Neural nets in a group decision process. *International Journal of Game Theory*, 31(3):447–467, 2002.

[14] S. Marshland. *Machine Learning: An Algorithmic Perspective.* Chapman & Hall, CRC Press, 2009.

[15] H. Moulin. On Strategy-Proofness and Single-Peakedness. *Public Choice*, 35(4):437–455, 1980.

[16] K. Nehring and C. Puppe. The Structure of Strategy-Proof Social Choice. Part I: General Characterization and Possibility Results on Median Spaces. *Journal of Economic Theory*, 135(1):269–305, 2007.

[17] W. Richards, H.S. Seung, and G. Pickard. Neural voting machine. *Neural Networks*, 19(8):1161–1167, 2006.

[18] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning Internal Representations by Error Propagation. In: D.E. Rumelhart, J.L. McClelland, and the PDP research group, editors, *Parallel distributed processing: Explorations in the microstructure of cognition*, Volume 1: Foundations. MIT Press, 1986.

[19] M.A. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, 1975.

[20] D. Sgroi and D.J. Zizzo. Learning to play 3×3 games: Neural networks as bounded-rational players. *Journal of Economic Behavior & Organization*, 69(1):27–38, 2009.

[21] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature* 529(7587):484–489, 2016.

[22] J.H. Smith. Aggregation of preferences with variable electorate. *Econometrica*, 41(6):1027–1041, 1973.

[23] P.H. Young and A. Levenglick. A consistent extension of Condorcet's election principle. *SIAM Journal on Applied Mathematics*, 35(2), 285–300, 1978.

Dávid Burka
Department of Mathematics
Corvinus University of Budapest
Budapest, Hungary
Email: `dburka001@gmail.com`

Clemens Puppe
Department of Economics and Managament
Karlsruhe Institute of Technology
Karlsruhe, Germany
Email: `clemens.puppe@kit.edu`

László Szepesváry
Department of Operations Research and Actuarial Sciences
Corvinus University of Budapest
Budapest, Hungary
Email: `szepesvaryl@gmail.com`

Attila Tasnádi
MTA-BCE "Lendület" Strategic Interactions Research Group, Department of Mathematics
Corvinus University of Budapest
Budapest, Hungary
Email: `attila.tasnadi@uni-corvinus.hu`