Can the human mind learn to backward induce? A neural network answer.

Leonidas Spiliopoulos[*]

University of Sydney, Chippendale 2002, Sydney, NSW, Australia

Abstract

This paper addresses the question of whether neural networks, a realistic cognitive model of the human information processing, can learn to backward induce in a two stage game with a unique subgame-perfect Nash Equilibrium. The result that the neural networks only learn a heuristic that approximates the desired output and does not backward induce is in accordance with the documented difficulty of humans to apply backward induction and their dependence on heuristics.

[*] Email: lspi4871@usyd.edu.au

1. INTRODUCTION

Neural networks are widely considered to be the most plausible cognitive model of parallel information processing in the human mind. The psychology and artificial intelligence literature is full of neural network models of memory, face recognition, language acquisition and vision to name a few. In the fields of economics, finance and business most applications of neural networks have been in the field of forecasting, for a specific application to inflation forecasting see Nakamura (in press), or for a collection of papers see Smith and Jatinder (2002). This paper however seeks to differentiate itself from the forecasting applications of neural networks and instead attempts to model human learning of economic games though the paradigm of a neural network. This work has been pioneered by Sgroi and Zizzo (2002) in a paper where they examine whether a neural network can learn to play the unique Nash equilibria in 3X3 strategy games. This paper aims to extend this work by focusing on an interesting question, namely that of whether a neural network can learn backward induction. This is of particular interest as failures of backward induction reasoning in humans have been documented in many papers e.g. Binmore, McCarthy et al. (2002) and Johnson, Camerer et al. (2002).

2. METHODOLOGY

The class of games studied in this paper is 2-stage, 2 player, with 2X2 strategy spaces in each stage and unique Nash equilibria. Each payoff is drawn randomly from

a uniform distribution with support from 0 to 100. In both stages, players are faced with 2 strategies, with (down, right) leading to the second stage as per figure 1.

Figure 1

Stage 1

		Player 2	
		Left	Right
Player 1	Up	a,b	c,d
	Down	e,f	Stage 2

Stage 2

		Player 2	
		Left	Right
Player 1	Up	g,h	i,j
	Down	k,l	m,n

The neural network has the following architecture: it is single-layer feed-forward with 10 neurons in the hidden layer (all neurons use a tanh activation function), uses a backpropagation algorithm with step learning and no momentum parameter, as this is a simple and realistic learning rule from a biological aspect. Training consisted of repeated presentation of 1000 different games, with batch learning, until the MSE of the network's output was 0.05[1]. A cross-validation data set or some other early stopping procedure to avoid overfitting was not used as these procedures are not biologically plausible or realistic[2]. All testing was done on sets of 2500 never before seen games in order to test the networks ability to generalize to out of sample games. The input layer consisted of 14 neurons, with each neuron inputting each player's payoffs from each cell in the game matrices. The output layer consisted

[1] Other specifications were also experimented with such as more hidden layer neurons, other steps sizes, more hidden layers, but the results on the test sets were not significantly different indicating the robustness of this approach. In the end, a step size of 1 was used as this led to more rapid convergence.
[2] Training the same neural network with minimization of cross-validation MSE instead of training MSE lead to the same success rates on test results, so overfitting does not seem to be a problem even for this simple model.

of 7 neurons each one representing a cell in the payoff matrices and the desired output was simply a 7-tuple vector with values of one for the cells which were the NE and zero otherwise[3]. The final choice of the network was assumed to be the one whose output neuron had the maximum value compared to the rest[4]. Five different networks were trained with different initial weights, so as to emulate five different players with different (random) priors.

The networks' performances were documented for three different test sets as presented in figure 2. The first test set is simply new games of the same structure as the ones presented during the training session. The other test sets are inspired by Harsanyi and Selten's (1988) separation of backward induction into rationality, subgame consistency and truncation consistency. The truncated test set seeks to look at the performance of the network when the game is truncated so that it is directly given the NE of the second stage game as the payoffs of g, h with i, j, k, l, m, n all set to zero. The last stage truncated test set is the same as the previous set except that all payoffs are transferred to the second stage matrix of the game (a, b, c, d, e and f are set to zero) allowing us to test for subgame consistency and framing effects. It will also give us an indication as to whether the neural network learned to solve for the NE of the second stage of the game, a necessary step if the network has learnt to backward induce.

3. RESULTS

[3] Note that this is a stricter prediction criterion than asking the neural network to simply play the NE strategy.

[4] A more realistic decision process could introduce some error in making the final decision, or imperfect discrimination between output signals but would not affect the qualitative results of this paper, only the quantitative success rates. As such the quantitative results in this paper should be viewed as upper bounds on performance.

Table 1 highlights some of the main results from the analyses of the neural networks output. This data comes from the averaging of the performance of the five trained neural networks. The average success rate on the training set was indeed high at 91%, which theoretically could be improved to 100% according to the universal approximation theorem which guarantees that there exists a set of network weights that will allow the neural network to achieve perfect performance (Hornik, Stinchcombe et al. 1989). As expected the performance on the never seen before test set falls significantly recognizing the Nash equilibrium in roughly 76% of games never encountered before.

Table 1 Success rates (%) of predicting NE in specific cells

	ab	cd	ef	gh	ij	kl	mn	Average
Training set	96.0	88.0	83.0	91.0	90.0	95.0	92.0	91.0
Test set	85.0	65.0	70.0	81.0	79.0	75.0	76.0	76.0
Truncated game set	72.0	57.0	78.0	81.0	-	-	-	72.0
Last stage truncated set	-	-	-	92.0	72.0	74.0	56.0	73.0

As half of the work has been done by solving for the Nash equilibrium of the second stage, it would be reasonable to expect the performance to be better on the truncated game test than on the standard test set yet it is virtually the same. Despite this, specific choices of NE differed indicating that the network suffers from truncation inconsistency. Testing instead on the last stage truncated test set had no significant impact on average performance, although success rates for specific cells differed indicating that the network treats payoffs differently depending on their position in the game matrix, thereby violating subgame consistency. Performance is again sub-

optimal indicating that the network has not learnt to solve for the NE of the last game and therefore has not strictly learnt to backward induce.

Different priors or experience do not seem to significantly affect the training set and the full game test success rates as the standard deviation of the success rates given in Table 2 is quite small.

Table 2 Standard dev. of success rates in predicting NE across the five neural network types

	ab	cd	ef	gh	ij	kl	mn	Average
Training set	0.7	2.2	3.0	3.7	1.5	2.2	1.6	0.5
Test set	6.2	4.1	5.4	4.7	5.1	3.5	2.5	3.5
Truncated game set	23.0	21.3	20.4	7.0	-	-	-	5.0
Last stage truncated set	-	-	-	3.7	15.2	7.9	11.9	7.1

However the standard deviation increases significantly with respect to predicting on the other sets with especially large differences in the success rates of predicting specific NE, although this large dispersion across types disappears when comparing average success rates in the last column. Hence, the different neural networks do exhibit some individualism as far as specific choices are concerned but their average performance is not very different.

Neural networks are usually regarded as “black boxes” by their critics because it is difficult to extract the learning rules they have endogenously learned as knowledge is distributed across the weights of the neural network. This paper opts to attempt to peer directly inside the black box by performing sensitivity analyses that explore how the output of the final layer neurons varies with respect to the values of the input neurons, whilst holding all other inputs at some fixed value (in this case at their means).

Across all the sensitivity analyses performed there was one very consistent result. The output of a particular output neuron increases when either of the two inputs associated with that cell increases. For example, the output of the neuron associated with the cell with payoffs a,b increases when either a or b increases. This very simple heuristic can be quite effective for the following reason. A cell is more likely to be the Nash equilibrium the higher its payoffs are because this increases the probability that the strategies associated with that cell are best responses to opponents' strategies. For example cell a,b has a higher probability of being the unique NE the higher a and b are. In similar spirit, the lower c,d,e and f are the higher the probability of a,b being the NE and this effect is also apparent from the sensitivity analysis although the magnitude of this effect on the output neurons is smaller and not consistently present. Also, such a heuristic has the advantage of working equally well when applied to a subset of the whole game, such as a single stage, as it does when applied to the whole game. This simple and reasonable heuristic gives the neural networks a performance of 76% compared to an expected success rate of 14.3% for random choices[5].

4. CONCLUSION

This paper has shown that solving for the NE in a backward induction games of more than one stage is not trivial even for the processing capabilities of a neural network. Although perfect performance is theoretically possible it is dependent on the learning rule's capability to bypass local minima in the error function and reach the

[5] In fact, a simple heuristic that predicts NE in the cell with the maximum sum of payoffs, i.e. the cell that maximizes social welfare, yields a success rate of about 60%, which is also significantly different from chance. Such a heuristic was also able to predict roughly 60% of the neural network's choices.

global minimum, a task that cannot be expected to be achieved successfully by a simple learning algorithm. Despite not discovering the global minimum, the neural network does learn a very sensible heuristic method for finding the Nash equilibrium, with an out of sample success rate of roughly 76%. Moreover, this heuristic appears to have great portability and robustness as it is capable of working equally well in environments where backward induction is or is not necessary i.e. single and multiple-stage games, and therefore is a cost and resource effective heuristic for solving many types of games, giving it added credence and plausibility as a cognitive model of NE selection in humans.

Concluding, the neural network does not seem to learn to backward induce as it exhibits imperfect performance, truncation and subgame inconsistencies, and implements a heuristic which does not seem to be significantly affected by the structure or even existence of stages in games. The preliminary results of this paper indicate that the use of neural networks may be an important tool in the game theory learning literature especially when addressing the issue of how people learn to generalize when faced with games of similar structure but differing payoffs from game to game.

References

Binmore, K., J. McCarthy, G. Ponti, L. Samuelson, and A. Shaked, 2002, A Backward Induction Experiment, Journal of Economic Theory 104(1), 48-88.

Harsanyi, J. C. and R. Selten, 1988, A General Theory of Equilibrium Selection in Games, Cambridge, MA, MIT Press.

Hornik, K., M. B. Stinchcombe and H. White, 1989, Multi-layer feedforward networks are universal approximators, Neural Networks 2, 359-366.

Johnson, E. J., C. F. Camerer, S. Sen, and T. Rymon, 2002, Detecting failures of backward induction: Monitoring information search in sequential bargaining experiments, Journal of Economic Theory 104, 16-47.

Nakamura, E., Inflation forecasting using a neural network, Economics Letters, In Press, Corrected Proof.

Sgroi, D. and D. J. Zizzo, 2002, Strategy Learning in 3x3 Games by Neural Networks, Department of Applied Economics, University of Cambridge.

Smith, K. and G. Jatinder, 2002, Neural Networks in Business: Techniques and Applications, Idea Group Publishing.