Title
Improving a model of human planning via large-scale data and deep neural networks

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 44(44)

Authors
Kuperwajs, Ionatan
Schuett, Heiko H
Ma, Wei Ji

Publication Date
2022

Peer reviewed

Improving a model of human planning via large-scale data and deep neural networks

Ionatan Kuperwajs (ikuperwajs@nyu.edu)
Center for Neural Science, New York University
New York, NY, United States

Heiko Schütt (heiko.schuett@nyu.edu)
Center for Neural Science, New York University
New York, NY, United States

Wei Ji Ma (weijima@nyu.edu)
Center for Neural Science and Department of Psychology, New York University
New York, NY, United States

Abstract

Models in cognitive science are often restricted for the sake of interpretability, and as a result may miss patterns in the data that are instead classified as noise. In contrast, deep neural networks can detect almost any pattern given sufficient data, but have only recently been applied to large-scale data sets and tasks for which there already exist process-level models to compare against. Here, we train deep neural networks to predict human play in 4-in-a-row, a combinatorial game of intermediate complexity, using a data set of $10,874,547$ games. We compare these networks to a planning model based on a heuristic function and tree search, and make suggestions for model improvements based on this analysis. This work provides the foundation for estimating a noise ceiling on massive data sets as well as systematically investigating the processes underlying human sequential decision-making.

Keywords: neural networks; machine learning; sequential decision-making; planning; behavioral modeling

Introduction

The standard approach to computational modeling in psychology involves handcrafting a model and specifying free parameters that may be adjusted to produce behaviors consistent with empirical data (Busemeyer & Diederich, 2010; Daw et al., 2011). Model predictions are then evaluated using the parameter values that achieve the best match to the data. Based on these evaluations, the model is iteratively amended to reduce remaining errors. Whether a specific change is accepted or not is usually based on model comparison techniques, balancing the tradeoff between complexity and goodness of fit. While this methodology yields interpretable models because all innovations are implemented by the researcher, it provides no guidance for when to stop searching for candidate models or what changes to try. Thus, there is no way to distinguish whether the unexplained variance represents natural variability in human behavior or could be explained by a crucial change to the model. Even if it can be determined that the model needs improvement, adjustments are usually based on intuition and manual engineering.

Here, we address these limitations by using deep neural networks to fit large-scale human behavioral data. Deep neural networks make minimal assumptions about underlying cognitive mechanisms and have sufficient capacity to represent virtually any computational process (Siegelmann & Sontag, 1995; LeCun, Bengio, & Hinton, 2015). Training a network to predict human behavior in a particular task allows the network to detect patterns in the data without requiring human understanding of these patterns. After training, the network can be used to judge whether there is still room for model improvements and, if so, in which situations these improvements should be made. Similar logic has been applied in recent work that leverages deep learning to discover algorithms underlying human decision-making (Agrawal, Peterson, & Griffiths, 2020; Peterson, Bourgin, Agrawal, Reichman, & Griffiths, 2021) and categorization (Battleday, Peterson, & Griffiths, 2020). One potential problem with this approach is that neural networks are so flexible that they run the risk of overfitting. To circumvent this, we use a large data set for training. This contrasts with previous efforts that use more complicated recurrent neural networks in much simpler tasks with less data (Dezfouli, Griffiths, Ramos, Dayan, & Balleine, 2019) and thus require regularization methods to ameliorate this problem.

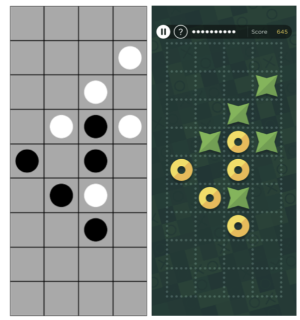

Figure 1: Example board position in 4-in-a-row. The left is the laboratory version of the task, while the right is the gamified version used on the Peak platform. Two players, black and white or yellow circles and green stars, alternate placing pieces on the board, and the first player to connect four pieces in any orientation wins the game.

We apply this approach to 4-in-a-row, a combinatorial game of intermediate complexity, to improve a model of human planning. First, we outline our methods for training these neural networks and show that our best network approaches a satisfactory upper bound on predictive power while matching human behavior well. We then compare the network to an interpretable cognitive model of human planning and discuss implications for improving this model based on our results.

In J. Culbertson, A. Perfors, H. Rabagliati & V. Ramenzoni (Eds.), *Proceedings of the 44th Annual Conference of the Cognitive Science Society*.

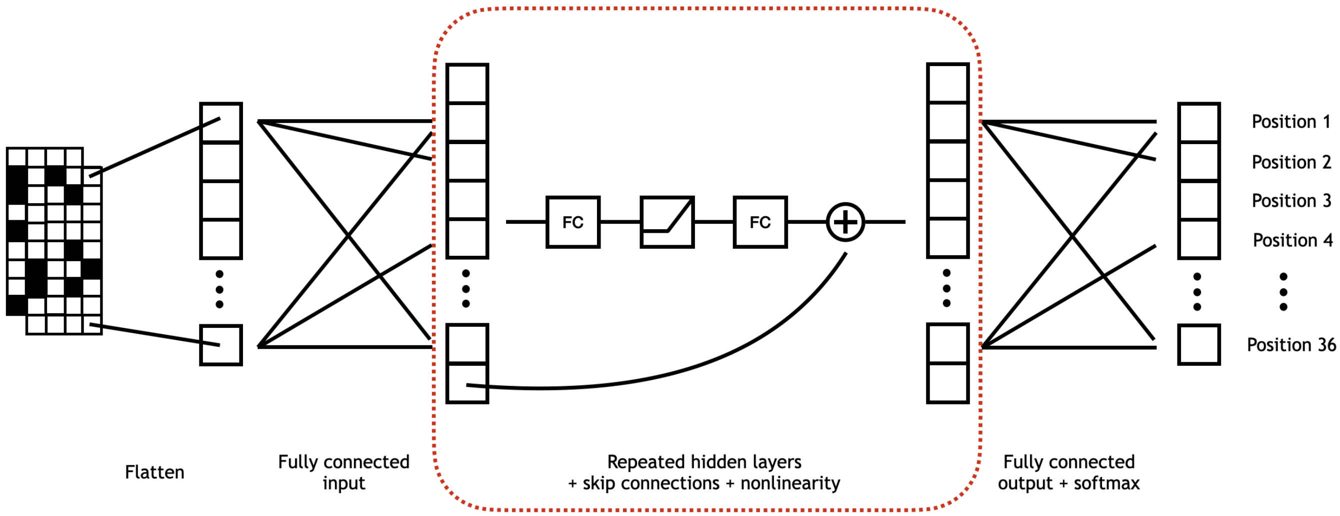

Figure 2: Neural network architecture. The board is represented as a $2 \times 4 \times 9$ tensor filled with zeros where there are no pieces and ones where there are pieces. One matrix encodes the user's pieces, and the second encodes the AI agent's pieces. The board representation is flattened to a 72 dimensional vector, and then passed into a series of hidden layers. Each hidden layer contains a fully connected layer, a nonlinearity, another fully connected layer, and then adds the input from the skip connections. Finally, the fully connected output layer has 36 units and is passed through a softmax function, which gives the probability that the model assigns to the human player selecting each position of the board. In addition to varying the number of hidden layers in the network, the number of units per fully connected layer is also varied when testing different networks.

Task and data set

Our task is a variant of tic-tac-toe, in which two players alternate placing tokens on a 4-by-9 board (Figure 1). The objective is to get four tokens in a row horizontally, vertically, or diagonally. From a broader cognitive science perspective, the game, which we call 4-in-a-row, is at a level of complexity which far exceeds tasks commonly used in psychology, providing rich human behavior for which computational modeling is still tractable (van Opheusden & Ma, 2019). The game has approximately $1.2 \cdot 10^{16}$ non-terminal states, and can be leveraged to study the interplay between different reinforcement learning systems (Kuperwajs, van Opheusden, & Ma, 2019), the nature of expertise (van Opheusden et al., 2021), or comparisons between human and machine learning. Importantly, a cognitive model exists for this task, which provides a strong starting point for further model development.

Additionally, we partnered with Peak, a mobile app company, to implement a visually enriched version of 4-in-a-row on their platform (`https://www.peak.net`), which users play at their leisure in their daily environment. We are currently collecting data at a rate of approximately 1.5 million games per month, and here we use a subset consisting of $10,874,547$ games from $1,234,844$ unique users collected between September 2018 and April 2019. In this version of the task, users always move first against an AI agent implementing a planning algorithm, with parameters adapted from fits on previously collected human-vs-human games (van Opheusden et al., 2021).

Methods

Data representation

Our networks take a tensor representation of the current board state and return a probability distribution for the next move over all board positions. The predictions for different board positions are independent of each other. We encode each board as two 4×9 binary matrices. The first matrix has ones indicating the location of the user's pieces, while the second 4×9 matrix has ones marking where the AI agent's pieces are located. Unoccupied locations contain a zero in both matrices. Thus, the input to each network is $2 \times 4 \times 9$, and the output of the network is a 36 dimensional vector, with each element representing a corresponding index of the board.

Network architecture

The architecture for our networks consists of an input layer that feeds into several hidden layers followed by an output layer (Figure 2). The input layer flattens the $2 \times 4 \times 9$ board into a 72 dimensional vector and projects it to the number of dimensions used by the hidden layers with a fully connected layer. Each hidden layer consists of two fully connected layers with a rectified linear function between them and skip connections. These skip connections add the input of the hidden layer to its output without transformation, and aid in avoiding the vanishing gradient problem (He, Zhang, Ren, & Sun, 2016). The output layer is a fully connected layer that projects from the dimensionality of the hidden layer to the output with 36 units corresponding to the log probabilities for each board position. During training, we systematically

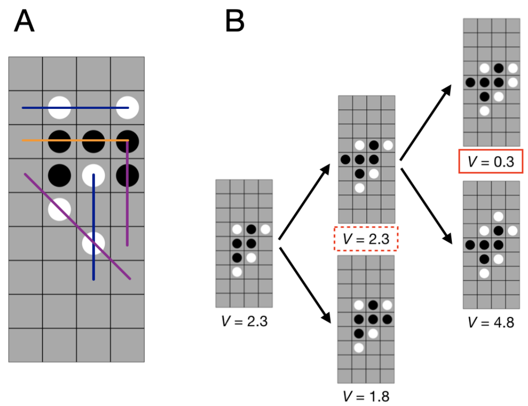

Figure 3: Planning model specification. **(A)** Features used in the heuristic function. Features with identical colors are constrained to the same weights. **(B)** Illustration of the best-first search algorithm. In the root position, black is to move. After expanding the root node with two candidate moves for black and evaluating the resulting positions using the heuristic function, the algorithm selects the highest value node ($V = 2.3$) on the second iteration and expands it with two candidate moves for white. The algorithm evaluates the resulting positions, and backpropagates the lowest value ($V = 0.3$), since white is the opponent. This will be compared against alternatives in each intermediate node of the tree to decide in which direction to expand the tree on the next iteration.

vary the number of hidden layers as well as the number of units in each fully connected layer.

On most boards, some moves are illegal because the corresponding square is occupied by pieces on the board. These moves can be eliminated as potential predictions a priori. To accomplish this, we subtract a large value from the output at occupied locations. The final softmax operator applied after this always sets the corresponding outputs to exactly 0 and normalizes the probability distribution over all open positions. This also nulls all gradients for the occupied positions such that the values at the these positions are ignored for gradient backpropagation and learning during training.

Prior work used deep convolutional networks to predict human moves in Go (Sutskever & Nair, 2008; Clark & Storkey, 2015), and we initially tested similar architectures in our task. However, we consistently found that the convolutional networks performed worse than the fully connected layers in preliminary training runs. Therefore, we decided to move forward using only fully connected networks.

Training

We partition the data into three sets: 90% for training (9,787,093 games), 5% for validation (543,727 games), and 5% for testing (543,727 games). The validation set was used to monitor learning and experiment with hyperparameters. We use stochastic gradient descent for training and reduce the learning rate by a factor of 10 if the loss associated with the validation set was stagnant for a few epochs. All layers had their biases initialized to 0 and weights drawn from a normal distribution with mean 0 and standard deviation 0.01, and we use a batch size of 128.

Planning model

The main goal of this work is to compare these neural networks with our current best interpretable planning model trained on the same task (van Opheusden et al., 2021), and subsequently make suggestions for how to improve the model. The model of interest combines a heuristic function (Figure 3A), which is a weighted linear combination of board features (Campbell, Hoane Jr, & Hsu, 2002), with the construction of a decision tree via best-first search (Figure 3B). Best-first search iteratively expands nodes on the principal variation, or the sequence of actions that lead to the best outcome for both players given the current decision tree (Dechter & Pearl, 1985). To allow the model to capture variability in human play and make human-like mistakes, we add Gaussian noise to the heuristic function and include feature dropout. For each move the model makes, it randomly omits some features from the heuristic function before it performs search. Such feature omissions can be interpreted cognitively as lapses of selective attention (Treisman & Gelade, 1980). During search, the model also prunes the decision tree by removing branches with low heuristic value (Huys et al., 2012).

While fitting the model, we estimate the log probability of a move in a given board position with inverse binomial sampling (van Opheusden, Acerbi, & Ma, 2020), and optimize the log-likelihood function with Bayesian adaptive direct search (Acerbi & Ma, 2017). To fit parameters for the entire training, we evaluate the log-likelihood on 10,000 trials, randomly sampled for each evaluation, which yields an unbiased and sufficiently precise estimate of overall performance. For testing, we ran 100 repetitions to estimate the log-likelihoods for each move and 200 simulations in each board position to get a probability distribution over potential moves.

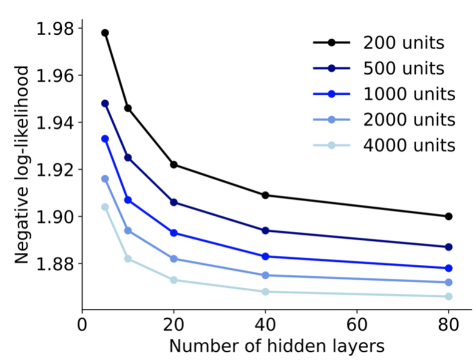

Figure 4: Scaling up the neural network achieves an upper bound on goodness of fit. Log-likelihood on the test data set as a function of the number of hidden layers and number of units per hidden layer in each network.

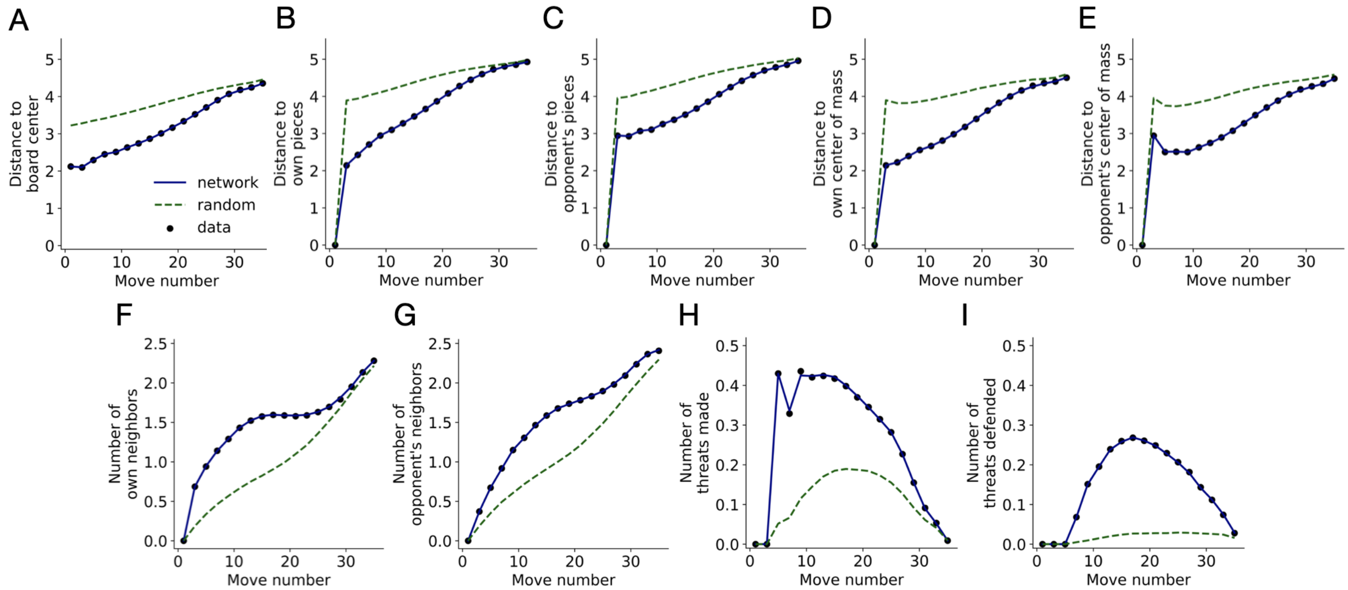

Figure 5: Summary statistics as validation of the neural network's performance. Each statistic is averaged by move number for moves made by users (black circles), the neural network (blue lines), or random moves (green dashed lines).

Results

Network evaluation

In order to ensure a reasonable estimate on the data's noise ceiling, we trained a total of 25 networks. Networks vary along two dimensions: the number of hidden layers and the number of units per layer, spanning a range from 5 to 80 layers and 200 to 4000 units. We continued scaling up the networks until the log-likelihood on the test data reached a plateau, meaning that additional increases in either dimension would not lead to significant increases in performance (Figure 4). This leads to the largest network achieving a negative log-likelihood of 1.866 per move and a prediction accuracy of 41.71% on the test data. As the largest network represents a satisfactory upper bound on predictability for the data set, we continue to analyze it in the remainder of the paper. We validate that this network's log-likelihoods per move are highly correlated with the networks that are one step smaller in either direction, further supporting our conclusion that our results would not radically change with larger networks.

Comparing the neural network's predictions to human choices is challenging because the data is high-dimensional and discrete, so we interrogate it via two methods. The first is the entropy of the network's output distributions, which we analyze by move number. Positions in the early game are harder to predict because they consist of fairly empty boards where no player can immediately win the game, and therefore result in higher entropy for the network's output distribution. Conversely, positions in the middle and late game are much easier to predict as there are less alternatives and more pieces to inform decision-making, leading to lower entropy for the network's output distribution. These positions are also more likely to contain winning moves.

Second, we compute a set of summary statistics that characterize human play in 4-in-a-row. For each move made by each user, we compute the distance from the chosen square to the center of the board, the distance to pieces owned by that user, the distance to pieces owned by the opponent, the distance to the center of mass of that user's pieces, the distance to the center of mass of the opponent's pieces, the number of that user's pieces on the 8 squares neighboring the chosen square, and the number of opposing pieces on neighboring squares. We also indicate whether with their chosen move, the user creates a threat to win on the next move or parries a threat from their opponent. We compute these statistics for moves made by the network in the same positions encountered by human players and for random moves. Figure 5 shows the average of these summary statistics aggregated across all users in the test set as a function of move number. This analysis probes systematic patterns in the time course of people's games, for example a tendency to start playing near the center of the board and gradually expand outwards. For all summary statistics, people deviate considerably from random, and the neural network matches the data almost exactly.

Model comparison and improvements

We compare predictions between the neural network and the planning model to search for potential improvements that will better capture human behavior. The model achieves a negative log-likelihood of 2.178 and accuracy of 34.06% on the test data, which is worse than the network's performance. We show that the network's predicted log-likelihood per move is typically higher than that of the model (Figure 6A) and that the model's average accuracy per move throughout the course

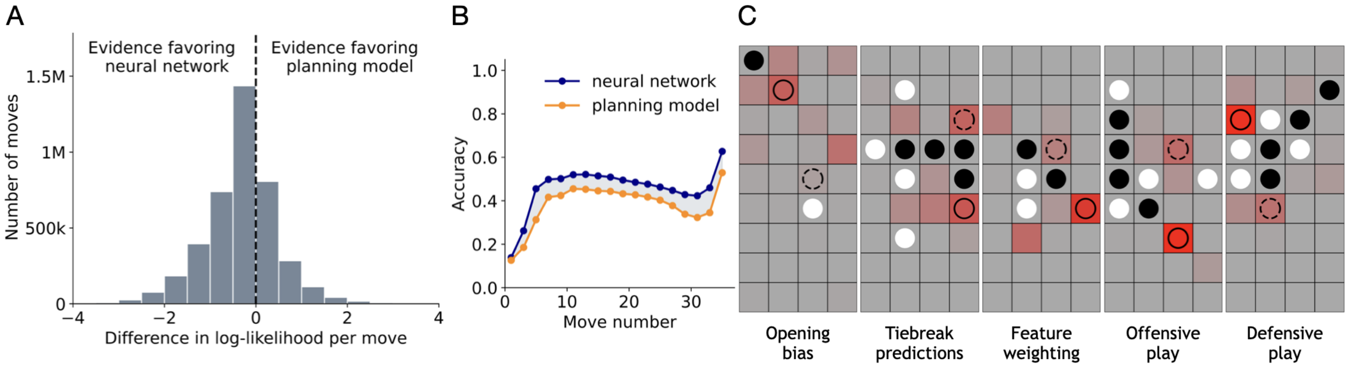

Figure 6: Comparing the neural network and a model of human planning. **(A)** Histogram of the difference in log-likelihood per move in the test set for the neural network and planning model (mean: $-0.31 \pm 3.91 \cdot 10^{-4}$). **(B)** Accuracy as a function of move number for the neural network (blue) and planning model (orange), averaged across the test set. The gray region denotes the net difference in prediction accuracy between the neural network and the planning model. **(C)** Representative board positions for discrepancies between the network and planning model, which could be used to improve mechanisms in the model. The red shading indicates the probability distribution of the network's move prediction, the open circle indicates the user's selected move, and the dashed circle indicates the planning model's predicted move.

of games is lower than the network's (Figure 6B). This provides us with a space of board positions to explore for model improvement, namely positions in which the network correctly predicts human moves while the model does not.

Inspecting these boards and their corresponding predictions yields several interesting patterns that the model cannot readily account for (Figure 6C). The first is an *opening bias*, where users are more likely to play on the left side of the board and in the corners. There is no strategic reason for making these types of moves, but the network detects these preferences nonetheless. A similar pattern is *tiebreak predictions*, where users select between two moves that are nearly equally good. In these positions, the network tends to have a preference for the move that was actually played by humans, whereas the model essentially assigns equal probability to both moves. A third difference is *feature weighting*, where users prefer creating certain features on the board to others. Feature weights are already a mechanism in the model, but some of the feature tradeoffs that the network can predict exhibit complex preferences: the example in Figure 6C shows a position in which a move connecting three pieces is selected over a move which connects two pieces in multiple directions. Finally, we find evidence that the network correctly predicts *offensive and defensive play*. That is to say, given a move which creates threats or features for the user and one which defends against an opponent's threats or blocks their features, the network often correctly predicts which move to make at a higher rate than the model.

One potential concern is that the neural network was not necessary to identify patterns that the planning model is overlooking, because direct comparisons between the model and human behavior was sufficient to detect these. However, such direct comparisons are not an expedient solution because human behavior contains substantial randomness. Even with the size of our data set, most board positions beyond the early game were only encountered once by human players. Thus, many of the 2,725,151 moves where the planning model predicted that a different move was more likely than the one that humans actually made represent unpredictable random human behavior rather than a failure of the planning model. To select moves that could have been predicted more accurately, prior work suggests the use of a powerfully predictive model like the neural network we use here (Agrawal et al., 2020). By pooling information across board positions, the neural network can produce a better estimate of the difference between the model and the true human policy and can thus give better guidance for model improvements. Indeed, the largest differences between the planning model and the neural network are more interpretable than the largest differences between the planning model and the data (Figure 7). Board positions that result in a high Kullback–Leibler (KL) divergence between the output distributions of the model and the network result in patterns that fit into the categories outlined in Figure 6C. Board positions that result in a low log-likelihood for the planning model are often not predicted well by the network either, and largely seem to be human errors in gameplay such as overlooking an immediate win or making a random move. Therefore, we argue that direct comparisons between model and data for individual positions are indeed noisy and that the network's predictions are a better guide for model improvement.

Discussion

In this paper, we trained deep neural networks to predict human moves in 4-in-a-row using a large-scale data set. We ensured that these networks estimate a reasonable upper bound on how well any model can explain human behavior by incrementally scaling up the networks, and then validated that

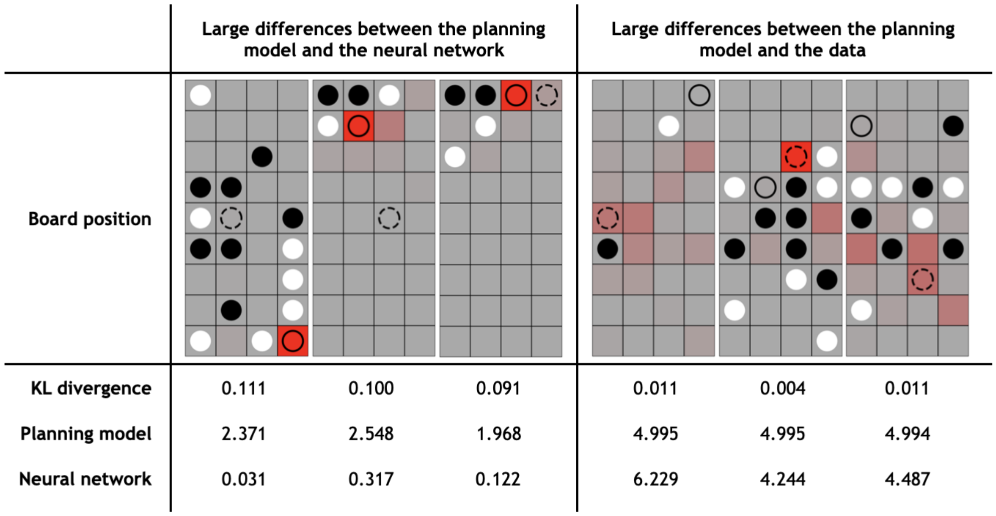

Figure 7: Representative residuals between the planning model and the neural network (left) and the planning model and the data (right). For each board position, we report the KL divergence between the output distributions of the model and the network, as well as the negative log-likelihood of the human move for the model and the network. The format for the board positions is the same as for Figure 6C.

the best network captures general trends in human play. This provided us with a model that was able to predict human decisions more accurately than an interpretable planning model without requiring manual engineering. We then explored the positions in which the neural network was more accurate than the planning model, leading to several candidate mechanisms that could be implemented in order to improve the model.

More generally, our work provides a framework that is useful for cognitive scientists to employ in model construction, particularly in large-scale experiments. The planning model described here has already undergone rigorous testing against alternatives. Human choices in 4-in-a-row seem to be consistent with a broad class of planning algorithms, as long as they contain a feature-based evaluation function, tree search, pruning, and attentional oversight. The current model is a representative of this class which balances simplicity with predictive power. The process for finding this model was based on manual adjustments and intuition for the game, and utilizing neural networks to discover missing features may have greatly expedited this search. Nonetheless, our results suggest mechanisms that have not been previously considered.

What do the mechanisms that we identify as potential model improvements tell us about planning and human cognition? One takeaway is that people have inherent biases, meaning that they consistently prefer one out of many equivalent solutions to problems when there is no rational reason to do so. Humans display such systematic biases in many tasks, and the literature on these biases and how to model them may be informative to structure the biases players show in 4-in-a-row (Griffiths, Chater, Kemp, Perfors, & Tenenbaum, 2010). Technically, such biases could be incorporated into the planning model fairly easily by simply increasing the probability that the tree search algorithm explores the alternatives humans prefer. Our feature weighting finding suggests that people's heuristic functions may be more sophisticated than a simple sum of features, accounting for complex tradeoffs between pieces on the board depending on the context of the board position. Further, we observed in earlier studies that individuals seem to evaluate positions differently, as feature weights vary when the planning model is fit to each participant. Adjusting the heuristic function to be more human-like and account for nuanced individual differences is a challenge, but the size of the data set paired with the neural network's predictions can guide this process. Finally, offensive and defensive play point to differences in planning strategies that could potentially be captured by scaling advantages for the player and the opponent relative to each other. The network is able to predict when humans play aggressively or not, but investigating how it does so remains a question for further exploration. While these specific features of gameplay are tied to 4-in-a-row, they point to the interaction between heuristic evaluations and forward search. These are fundamental aspects of human planning, and uncovering more sophisticated implementations for either process may provide principles that generalize across planning tasks. In future work, we plan to integrate these results into the planning model.

Acknowledgments

This work was supported by Graduate Research Fellowship number DGE183930 from the National Science Foundation, grant number IIS-1344256 from the National Science Foundation, and grant number R01MH118925 from the National Institutes of Health.

References

Acerbi, L., & Ma, W. J. (2017). Practical bayesian optimization for model fitting with bayesian adaptive direct search. In *Proceedings of the 31st international conference on neural information processing systems* (pp. 1834–1844).

Agrawal, M., Peterson, J. C., & Griffiths, T. L. (2020). Scaling up psychology via scientific regret minimization. *Proceedings of the National Academy of Sciences*, *117*(16), 8825–8835.

Battleday, R. M., Peterson, J. C., & Griffiths, T. L. (2020). Capturing human categorization of natural images by combining deep networks and cognitive models. *Nature communications*, *11*(1), 1–14.

Busemeyer, J. R., & Diederich, A. (2010). *Cognitive modeling*. Sage.

Campbell, M., Hoane Jr, A. J., & Hsu, F.-h. (2002). Deep blue. *Artificial intelligence*, *134*(1-2), 57–83.

Clark, C., & Storkey, A. (2015). Training deep convolutional neural networks to play go. In *International conference on machine learning* (pp. 1766–1774).

Daw, N. D., et al. (2011). Trial-by-trial data analysis using computational models. *Decision making, affect, and learning: Attention and performance XXIII*, *23*(1).

Dechter, R., & Pearl, J. (1985). Generalized best-first search strategies and the optimality of a. *Journal of the ACM (JACM)*, *32*(3), 505–536.

Dezfouli, A., Griffiths, K., Ramos, F., Dayan, P., & Balleine, B. W. (2019). Models that learn how humans learn: the case of decision-making and its disorders. *PLoS computational biology*, *15*(6), e1006903.

Griffiths, T. L., Chater, N., Kemp, C., Perfors, A., & Tenenbaum, J. B. (2010). Probabilistic models of cognition: Exploring representations and inductive biases. *Trends in cognitive sciences*, *14*(8), 357–364.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778).

Hochreiter, S., Younger, A. S., & Conwell, P. R. (2001). Learning to learn using gradient descent. In *International conference on artificial neural networks* (pp. 87–94).

Huys, Q. J., Eshel, N., O'Nions, E., Sheridan, L., Dayan, P., & Roiser, J. P. (2012). Bonsai trees in your head: how the pavlovian system sculpts goal-directed choices by pruning decision trees. *PLoS computational biology*, *8*(3), e1002410.

Kuperwajs, I., van Opheusden, B., & Ma, W. J. (2019). Prospective planning and retrospective learning in a large-scale combinatorial game. In *2019 conference on cognitive computational neuroscience. berlín, alemania* (pp. 13–16).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436–444.

Peterson, J. C., Bourgin, D. D., Agrawal, M., Reichman, D., & Griffiths, T. L. (2021). Using large-scale experiments and machine learning to discover theories of human decision-making. *Science*, *372*(6547), 1209–1214.

Siegelmann, H. T., & Sontag, E. D. (1995). On the computational power of neural nets. *Journal of computer and system sciences*, *50*(1), 132–150.

Sutskever, I., & Nair, V. (2008). Mimicking go experts with convolutional neural networks. In *International conference on artificial neural networks* (pp. 101–110).

Treisman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive psychology*, *12*(1), 97–136.

van Opheusden, B., Acerbi, L., & Ma, W. J. (2020). Unbiased and efficient log-likelihood estimation with inverse binomial sampling. *PLoS computational biology*, *16*(12), e1008483.

van Opheusden, B., Galbiati, G., Kuperwajs, I., Bnaya, Z., Ma, W. J., et al. (2021). Revealing the impact of expertise on human planning with a two-player board game.

van Opheusden, B., & Ma, W. J. (2019). Tasks for aligning human and machine planning. *Current Opinion in Behavioral Sciences*, *29*, 127–133.

Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z., Munos, R., . . . Botvinick, M. (2016). Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*.