Predicting Human Behavior Using Tic Tac Toe

CSCI 357 AI & Cognitive Science Final Project Research Paper

Christina Yu, Ian Herdt, Lawrence Li, Steven Iovine
Professor Christopher Dancy

Project Description

This project seeks to predict human behavior in the game of tic tac toe. More specifically, this project is interested in creating a neural network that is able to predict user decision-making based on their previous behavior. To train the network, a large number of training games must be played between AI and the human player, which can later predict human behaviors in the context of the game. Our main goal is to predict how human players may choose the slot based on the current game board state. In order to stay within the theme of AI & human behavior, the computer player will not use such predictions against the human player. Instead, only the individual deep learning network is able to predict. For the prediction phase, once the human player does its move, the neural model would output what it may have thought the user was going to do based on the current game state.

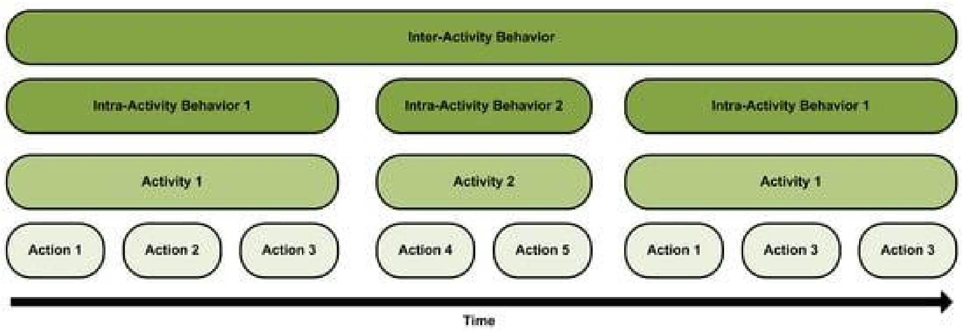
Initiatives

Predicting human behavior is one of the most interesting topics in artificial intelligence and learning. However, researchers cannot make predictions about human behavior in the same way a physicist can make predictions in a lab. Like the weather prediction, there are complex patterns and lots of undesirable or unpredictable factors that can significantly impact the accuracy. Nevertheless, in theory, if scientists had all the information about all the nature and nurture motivations, we could predict human behavior with accuracy.

This is why we chose the game of tic tac toe as our topic in predicting human behavior. The tic tac toe is one of the best examples that we can have all the information about player motivations. The goal of tic tac toe is to win the game by getting 3-in-a-row, and we can gather details on how to win the game by strategies. There are strategic ways to win in tic tac toe that resemble a smarter and more creative way of thinking. We ultimately wanted to observe whether the neural net would be able to predict our specific strategy and method of winning.

Research Background

Tic Tac Toe is a game that can help you guide your own choices and realize the impact your actions have on others. This is related to human behavior and activity evaluation. The idea behind machine learning techniques is to learn and predict behaviour and activity models. These are usually presented as a supervised learning approach, for which different techniques have been used to learn behaviours and activities from collected data. The following diagram shows how human behavior is modelled.



Elements of the user behavior. (Almeida & Azkune, 2018)

As shown, user behaviours are a large collection of human activities, making them a complex structure. According to the human behavioral prediction paper, human behavior can be specified into three key components: actions, activities and behaviours. Actions describe the simplest conscious movements, whereas behaviours describe the most complex conduct. In this project, we are dealing with human actions in each round of tic tac toe gameplay. Dividing human behaviors into each human actions can make the prediction simple and easy to be evaluated.

In addition to the human behavioral aspects in actions, existing researches and studies also show that predicting such behavior can be used in assessing the intelligence of a student through such games for career guidance that uses cognitive models that analyze intelligence, patience, perseverance, speed of solving problems etc. (Jammalamadaka & Vudatha, 2019) These human behavioral prediction models help illustrate and demonstrate the perspectives on the human and emotional aspect in addition to the control that players have throughout the game.

The assessment of student behavior for career guidance resulted in an expert system called Tic-Tac-Toe Game Playing Career Guidance System that is useful to assess the psychological factors of the student through Tic-Tac-Toe game play, and build the cognitive model of the student and predict the appropriate career for the student based on the students performance. (Jammalamadaka & Vudatha, 2019) This is what similar models we will create in this project: building a neural network model based on tic tac toe game plays with the computer, recording their game play and being trained based on their previous actions to predict their next actions.

The wisdom of tic tac toe behind the control of the game provided a perspective on the human/emotional aspect behind the control that players have throughout the game, including two of the major plays in the tic tac toe: forking and blocking. From the game theory and psychological lessons, the tic tac toe can be categorized into the following:

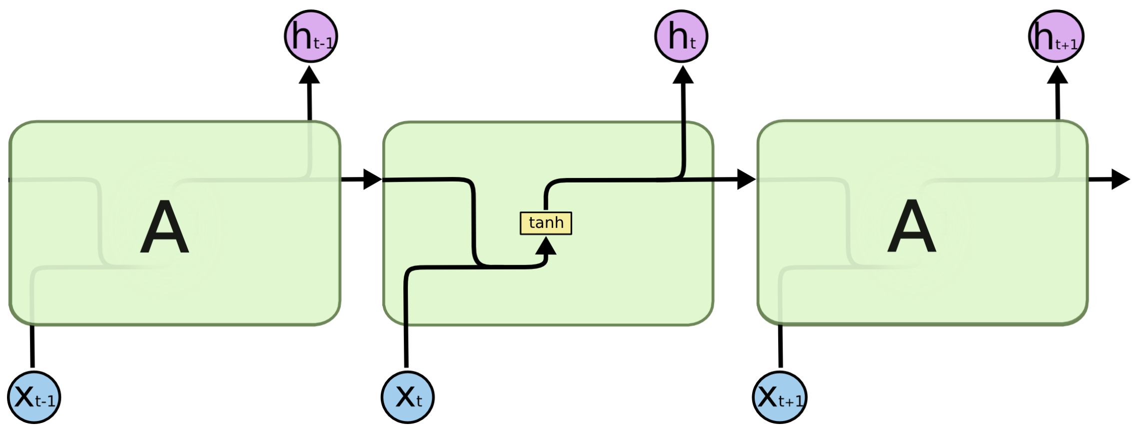
1. **Futile**: no-win scenario (as seen in WarGames)
2. **Zero-sum game**: First player always has advantage, second player must defend or lose
3. **Nash equilibrium**: each player prevents the other from winning
4. **3 Outcomes through Zermelo's Theorem**: any finite, two-person game of perfect information and alternating moves, either a) one of the sides can force a win, or b) both can force a draw

When using the game theory into the cognition and human behavior of tic tac toe into the assessment of student behavior for career guidance, one can encourage logical thinking, looking ahead and making predictions, as well as develop spatial reasoning and problem solving skills.

In the experiment of flexible strategy use in young children's game play in tic tac toe, the children's playing behavior has demonstrated a very interesting scenario: when the children attempted to play aggressively to win, they tended to fail to block the opponents and hence lose the game. On the other hand, when the children attempt to block to prevent the opponent from winning the game, they usually end up in a draw. The goal-based strategies in young children demonstrated some degree of human behavior that focus on exploring the game, finding the way to beat it, and analyzing the previous, current, and next moves. (Crowley & Siegler, 1993)

Approach

We chose the long short-term memory recurrent network for the implementation, where it can learn order dependence in sequence prediction problems, and connect past action sequences to predict the next action. One strength of LSTM networks is that they can remember information for long periods of time, which is suitable for the game like tic tac toe where the network needs to remember lots of opponent's actions to predict the next move, and hence predicting the human behavior.

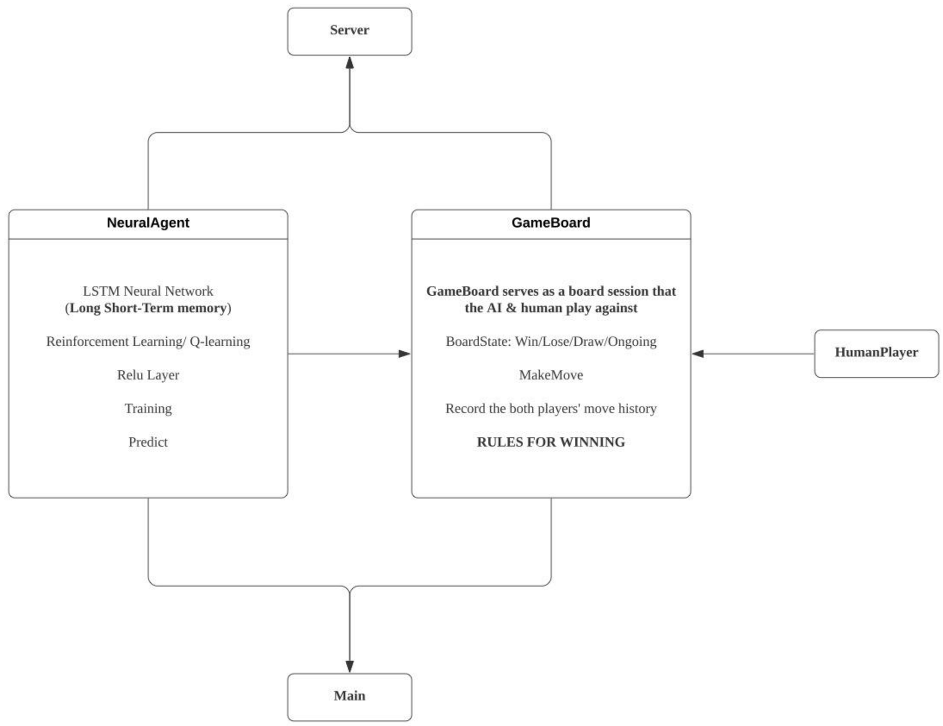


The repeating module in a standard LSTM layer

In addition to the neural network, we also needed a game board to record the moves of both players, which can then decide the outcome of the game by the winning rules: 3-in-a-row in horizontal, vertical, and diagonal. The approach here is that the LSTM neural network is able to fetch full information of the opponent's move sequences and predict the opponent's next move and behavior, which is desirable for predicting human behavior.

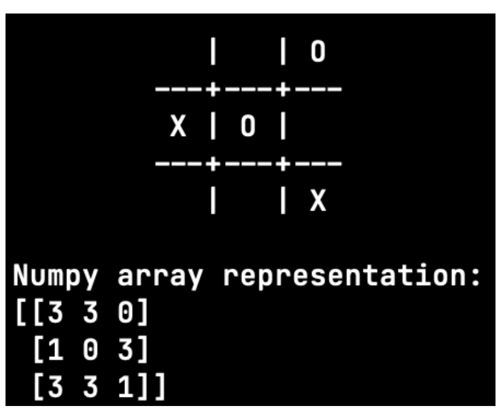
LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is their default behavior. All recurrent neural networks like LSTM have the form of a chain of repeating modules of neural network.

An UML class diagram of our approach is shown below:



The UML diagram showing the dependencies between classes in our approach

The class diagram shown above consisted of two major classes: ***NeuralAgent***, and ***GameBoard***. Both classes are responsible for predicting the human behavior in a tic tac toe game. The first class is ***GameBoard***, where it can record the game progress and store them as numpy arrays. Here is an example showing how ***GameBoard*** represents the game.



Numpy array and visual representation of the tic tac toe

As shown in the example above, the project uses a 2-dimensional numpy array to represent a tic tac toe game board with both rows and columns. The number representation in the numpy array shows the current states of the slot in the game board, which is specified as the following:

0: The slot where the “O” is
1: The slot where the “X” is
3: The slot that is still open

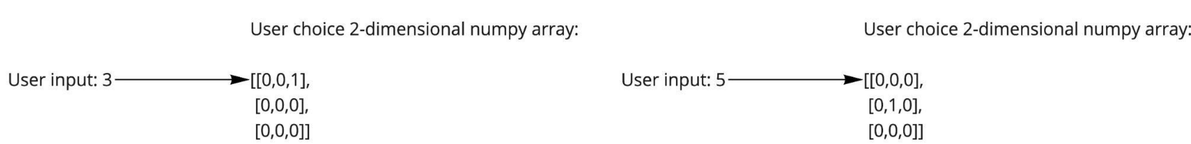
When the user chooses an open slot, it will input an integer between 1 to 9, specified as the following figure.

```
np.array([[1, 2, 3],
          [4, 5, 6],
          [7, 8, 9]])
```

The number representation of slots' location for user input

As the above figure suggests, an input of 5 will indicate the middle slot, an input of 1 will indicate the slot in the top-left, and an input of 8 will indicate the slot right below the middle slot (i.e. middle-bottom).

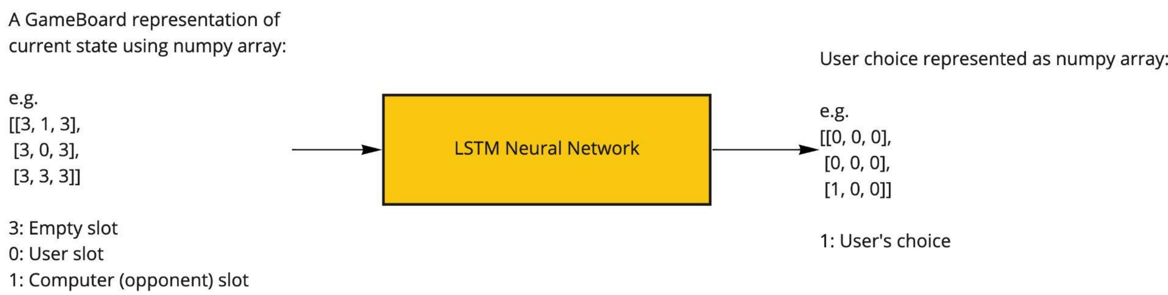
When the user selects a number for input, the game board will record the user’s choice and add that into the board. The board will also convert the user choice into a 2-dimensional array for neural network training. This 2-dimensional array consisted of only two kinds of numbers: 0 and 1, where 1 is the user’s choice. The array represents the slot location where the user has chosen. Here is an example.



The conversion from user input number to a 2-dimensional numpy array

As shown, the user choice number is converted to a 2-dimensional numpy array reflecting that choice. This 2-dimensional array acts as a user output array for the long short-term neural network training.

The long short-term neural network will have the following input and output prediction:



An input output (training datasets) diagram for the LSTM neural network

As shown in the figure above, the input array is a current state of the tic tac toe game board and the output will be the user choice represented as a 2-dimensional array. As both input and output are represented as numpy arrays, we are able to pass the numpy arrays from the game board to the LSTM neural network for both training and prediction.

The LSTM neural network consisted of the following layers.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 3, 9)              468
_________________________________________________________________
batch_normalization (BatchNo (None, 3, 9)              36
_________________________________________________________________
dense (Dense)                (None, 3, 3)              30
=================================================================
Total params: 534
Trainable params: 516
Non-trainable params: 18
_________________________________________________________________
```

As shown, we included one long short-term memory layer with a batch normalization to avoid overfitting. We also have a dense layer as output. The deep learning model is very simple since we only train the tic tac toe with a limited amount of data as we play to train.

Due to the fact that there are no training datasets for this project, we did the training by playing the game manually. As the game comes to an end, the program offers a few options to the human player. The player can respond to these options by typing “y” for yes and “n” for no. The first option is to train the neural agent, which essentially takes each state of the game we just previously played, inputs them into the neural agent, and do its predictions during the game with what actually happened. The LSTM model will train each time when a tic tac toe game is played and the user chooses “y” for training. The training has a total of 500 epochs to train a set of inputs and outputs. The training only takes about 6-7 seconds for each tic tac toe game play. However, in order to train for every possible set, the user needs to train the agent a numerous amount of times before the predictions given can be at a satisfactory level, which will be discussed in detail in the section below.

Results

The results of the training are not as outstanding as we expected, but it is enough as a satisfactory answer for this project. Since we did the training by manually playing the game, we will measure the accuracy by playing the game 10 times.

Training Entries	Training Accuracy
1	0.4166666567325592
2	0.8888888955116272
3	0.7777777910232544
4	0.4444444477558136
5	0.6666666865348816
6	0.3333333432674408
7	0.7777777910232544
8	0.75
9	0.5555555820465088
10	0.7777777910232544

Average	0.6388888985

A table showing the training accuracy of playing 10 tic tac toe games manually

As shown in the table above, the accuracy is fluctuating through multiple games. Due to the fact that there are too many scenarios that a neural network can predict, the accuracy of neural network training will continue to be fluctuating when doing training by playing games manually. It will take a very long time to manually play the game and train the results to a satisfactory level.

The prediction of human behavior will be shown in the following example:

```
         O | X | X
        ---+---+---
           |   |
        ---+---+---
           | O | X

Numpy array representation:
[[0 1 1]
 [3 3 3]
 [3 0 1]]
Prediction:
[['Closed Slot' 'Closed Slot' 'Closed Slot']
 ['0.000000857' '0.000002405' '0.252934635']
 ['0.000007810' 'Closed Slot' 'Closed Slot']]
```

The program output shown above is predicting the playing behavior of “O”. It shows a 2-dimensional array showing which slot is more likely to be played by the human player. The number is scaling from 0 to 1, where 0 is not likely and 1 is most likely. In this scenario, the highest score will be the slot at the middle-right because the human player needs to prevent the “X” from reaching 3-in-a-row in that column.

During the observation of results we found that the model provides inaccurate predictions on the user choice slots. These observations are usually in conflict with the basic common rules in tic tac toe. This may be a result of lack of training with a probability of overfitting issue in LSTM neural network. For instance, the following illustrates a tic tac toe game play where the human player “O” and computer player “X” are playing against each other. In this scenario, the human player “O” will be the next turn.

```
    O |   |
   ---+---+---
    O |   | X
   ---+---+---
      | X | X

Numpy array representation:
[[0 3 3]
 [0 3 1]
 [3 1 1]]
Prediction:
[['Closed Slot' '0.000000026' '0.003204137']
 ['Closed Slot' '0.000000000' 'Closed Slot']
 ['0.000006608' 'Closed Slot' 'Closed Slot']]
```

As shown here, the model predicts the highest probability of the human player (The "O" turn) making a move at the top right corner of the board. However, since this is the user "O" turn, it can just simply choose the bottom left corner to win the game. This is one example that shows there are conflicts between the prediction and the common sense of the basic tic tac toe game, which is a major downside of this solution. One of the reasons that the prediction is off can be lack of training as well the training method, where training based on a single game session is not the best way to learn human behavior efficiently. The other possible reason is the lack of tic tac toe rules, where we ignored the impact of tic tac toe rules on human behavior, which also caused some inaccuracies. Although this can be solved by incorporating min-max algorithms, it is also possible that with enough training, the neural agent will be able to identify those rules and be able to predict it. Because of that, it is possible that with sufficient training considering all possible scenarios, the probability of error predictions described above can be minimized. This is one of the ongoing issues that persists in the neural model.

Utopia

In a perfect world, the development and implementation of this project idea would have an impact infinitely larger than anything the group would have ever intended. One utopic scenario we can theorize about is that of developing a neural agent whose abilities to learn human behavior. Similar to Dale Purves' article discussing how the development of AI designed to win complex board games can help us better understand the function of the human brain (Purves, 2019), our project would result in the creation of a neural agent that skyrockets our understanding of human brain function/behavior. Our project could unveil new developments in how the brain works to make decisions, as well as how to model such in a virtual environment.

Additionally, the agent that we developed could prove to be adept at predicting human behavior, in general. The agent could emulate human behavior flawlessly, and find itself being used in various human behavior related scenarios. It could be implemented in mental health facilities, parsing info of different patients, and learning to predict any outbursts or actions that would harm others. The agent could be deployed at crime scenes and run a multitude of simulations of different de-escalation techniques in order to determine the perfect way to end the situation with minimal casualties. It could even be used on a massive scale to predict how populations would react to the passing of different laws and controversial political decisions in order to minimize public rioting and backlash.

Dystopia

We explore 3 different dystopia scenarios in question. The first involves using the Tic Tac Toe for unethical purposes - as we mentioned earlier, teachers have been using these games to determine career paths for children. In a dystopian world, people may use the game in more serious situations (for instance, determining college admissions or job applications) and continue to make inaccurate judgements of someone based on their performance in Tic Tac Toe.

What does it mean if you're interacting with a computer rather than a person during a game, where you were used to having some form of human connection? Multiplayer games like Tic Tac Toe are generally played with human interaction, and with the rise of computer automated games, social media, and ubiquitous computing, this raises serious concerns about how we understand each other as humans. We risk losing our ability to empathize with others as we constantly interact with a screen rather than a person.

Our final and worst dystopian scenario (which may as well be a Black Mirror episode) involves instrumental convergence: the hypothesis that sufficiently intelligent agents may pursue unlimited goals if their ultimate goal is unconstrained. In other words, a super AI with an unbounded, but seemingly harmless goal, could potentially act in destructive ways. In a dystopia, a superintelligent Tic-Tac-Toe machine may disregard all living beings on Earth in order to create a powerful computer with immense computational power to make perfect predictions.

Conclusion

The general approach to the tic tac toe human behavior prediction is appropriate as we integrate the prediction of human player move and the deep learning model. The LSTM neural model is able to achieve the goal of researching and executing a project that highlights the intersection between AI and Human Behavior through the medium of Tic Tac Toe. However, the implementation and execution did face a variety of challenges and issues the agent struggled to parse any of the key strategy components of the game from its limited experience of playing against members of the team. In addition to not learning from human behavior tendencies as much as we would have hoped, we also did not manage to get to our goal of creating a visual representation of the gameboard outside of the terminal, which was unfortunate.

The approach that is not satisfactory is the integration of the basic rules of tic tac toe when creating a deep learning network. So far the training datasets only consisted of a current game board state and the human player input. We can improve the predictions by adding additional predicting modules integrating those game rules besides the one that the LSTM network provides.

Although we made it possible for neural networks to have basic predictions of human behavior, improving such models by playing games manually could be time consuming and would require lots of users' efforts to play the game against the computer and require retraining per game, which is already described in detail in the "**Results**" section. This is one of the major downsides adopted in this project. We definitely underestimated how much training a neural net needs, which is what went wrong with the neural network approach in predicting human behavior.

In the future, we would like to implement a different approach to predict human behavior, or improve the LSTM network for training and prediction accuracies. One of the ideas we can do is the heuristic search tree to implement a min-max algorithm. We may also use a min-max algorithm to generate training data for the LSTM model and use Q-learning for prediction and response to human player's actions. We also needed to assess which strategy works and can be adopted to ensure its appropriateness.

References

1. Almeida, A., & Azkune, G. (2018). Predicting human behaviour with recurrent neural networks. *Applied Sciences*, 8(2), 305. doi:10.3390/app8020305

2. Bernardino, I., & Grieve, N. (2020, May 13). The wisdom of tic tac toe. Retrieved May 16, 2021, from https://medium.com/age-of-awareness/the-wisdom-of-tic-tac-toe-18f2355905b3

3. Bhalla, H. (2020, March 25). Using AI and Non-euclidean games to play tic-tac-toe with humans. Retrieved May 16, 2021, from https://medium.com/@hargun.bhalla1/using-ai-and-non-euclidean-games-to-play-tic-tac-toe-with-humans-af9ed4c8c0c

4. Crews, T. R., & Boone, R. S. (2017). Learning about machine learning through tic-tac-toe competition scenarios. *Journal of Computing Sciences in Colleges*, 33(2), 205-212.

5. Crowley, K., & Siegler, R. S. (1993). Flexible strategy use in young children's tic-tac-toe. *Cognitive Science*, 17(4), 531-561. doi:10.1207/s15516709cog1704_3

6. Lengacher, D. (2019, March 06). How to build LSTM neural networks in Keras. Retrieved May 16, 2021, from https://medium.com/@dclengacher/keras-lstm-recurrent-neural-networks-c1f5febde03d

7. Geoffreyyip/numpy-tictactoe. Retrieved May 16, 2021, from https://github.com/geoffreyyip/numpy-tictactoe

8. Jammalamadaka, S. K., & Vudatha, C. P. (2019). Assessing the intelligence of a student through tic-tac-toe game for career guidance. *International Journal of Electrical and Computer Engineering (IJECE)*, 9(6), 5545. doi:10.11591/ijece.v9i6.pp5545-5551

9. Long-short term memory neural network model. (2018, June 23). Retrieved May 16, 2021, from https://www.kaggle.com/kokonut/long-short-term-memory-neural-network-model

10. Sherstinsky, A. (2020). Fundamentals of recurrent neural Network (RNN) and Long short-term Memory (LSTM) network. *Physica D: Nonlinear Phenomena, 404*, 132306. doi:10.1016/j.physd.2019.132306