Chess with Thinking Pawns

Abstract

What makes a rational actor rational? Most quantitative theories of behavior, whether human or animal, simply assume this question away and imagine a world in which decision-making operates consistently, calculatedly, and quickly. Specifically, the economic tool of game theory uses a systematic calculation of the payoffs of each potential decision to figure out what strategy is most efficient. Real-world actors are not so easily categorized and calculated due to imperfection, emotion, and spontaneity. This paper seeks to explore the processes by which intelligence is developed and how different methods for creating intelligence can lead to very different results, many of which may not fall in line with traditional economic predictions. I simulate a simple two-player peace-war game in which each competitor may choose to attack or make peace with the other, with different payoffs for each action. Each player is trained with a different genetic algorithm modelled after natural selection. I find that, while game theory is moderately good at predicting the behavior of most evolved intelligences, it definitively fails to explain the emergence of strategies other than the Nash-equilibrium strategy. Furthermore, the overall best strategy (a tit-for-tat player, which only attacks when attacked) is not selected for by game theory's payoff metric. In fact, those players evolved by a technique which selected for the least optimal strategies according to game theory were far more likely to develop the tit-for-tat strategy and scored higher on average. To delve deeper into this question, I hope to simulate more complex games and integrate sophisticated techniques such as deep learning into the algorithms I have trained in the future.

Rationale

All of us are decisionmakers, and all of our decisions implicate each other. The advent of modern technologies of communication, production, and consumption have brought into sharp relief the relevance of economic laws to our daily lives. Workplaces, schools, and social spaces are increasingly defined by scores, competition, and rules; test-score-based pay raises have transformed the teaching profession into a game of statistics, and data-driven parenting techniques have become ubiquitous (Levitt & Dubner, 2011). Our cumulative pursuit of profit is destroying our planet because each has more to gain from exploiting resources than from conservation (Newkirk, 2016). Economics is inextricably embedded in the policymaking process, which determines our taxes, healthcare, and prisons. Life becomes a game, and so we are left to wonder how best to play. The task at hand is to build the tools to find our answers.

One of the most descriptive tools we have in life is the study of life itself: biology. Evolutionary psychology can provide powerful insight into the nature and origin of human behavioral traits (Burke, 2014). Analyses of precisely why these traits are successful in economically scarce situations governed by asymmetrical incentives is the wheelhouse of game theory, which makes it especially useful in understanding the workings of human societies because they are often extremely asymmetrical and feature constant competition (Gibbons, 1997). These models alone, however, fail to capture probably the most unique aspect of human societies: the human mind. The quest to imitate human intelligence on a computer drove scientists to develop a technique known as deep-learning – a simulation of a single brain function built using increasingly complex “abstractions” which build up simple concepts into more descriptive heuristic evaluators (Goodfellow, Bengio, & Courville, 2016).

Only recently have researchers begun to explore how well game theory can predict the behavior of evolved intelligence. While the efficiency of genetic algorithms developed with game-theory in mind far outperforms previous evolutionary algorithms in speed of training and accuracy, they still require explicit programming of meta-parameters, such as the mechanisms for imitation, belief, and communication (Yang, 2017). The research does not use a deep-learning framework as “genetic material,” which means while current techniques can deliver interesting results by observing the interactions of a large quantity of simple actors (such as an overall model of a market’s equilibrium price), they miss out of observing more direct interactions of complex actors (Protopapas, Kosmatopoulos, & Battaglia, 2009). Those kinds of nonlinear interactions are necessary to accurately simulate how intelligent actors interact in real life due to intrinsic variations and fluctuations in how we think, the fact that nobody is perfectly rational, and contextual psychological effects such as paranoia or social awkwardness (Osoba & Davis, 2018). Likewise, although complex machine thinking has indeed been achieved through neuro-evolution of deep networks (e.g., David & Greental, 2017), these methods are unable to comprehensively simulate a system of actors; they may be able to model and gradually evolve a population of neural networks, but the fitness systems used are not rooted in game theory and thus are closer to test scores than actual natural selection.

Despite these limitations, an intersection between these two fields has the potential to greatly enhance existing studies of game theory by more accurately modelling intelligent decision-makers. Specifically, applying neuro-evolution to deep-learning can greatly speed up selection of hyper-parameters and make the task of tuning the network much easier, such as the number of neurons or the expression used to represent a model's accuracy (Miikkulainen et al., 2017). Particularly in the context of games-playing, deep learning might outperform other methods because it can easily learn to recognize features in a variety of contexts, instead of relying on a more calcified initial input layer to generate results. For example, recognizing that a particular set of pixels constitutes a particular shape, regardless of its position on the screen, might allow a deep network playing Tetris to succeed where a Bayesian network based solely on discrete probabilities might fail. Deep learning, unlike other methods, makes better use of massive data sets because it does not require a scientist to intervene and label training data, but rather develops its own organic understandings (Mahapatra, 2018). Interestingly, certain pure genetic algorithms which don't even incorporate neural networks in their architecture have been able to outperform deep learning networks in myriad Atari games (Dar, 2018). As it stands, these genetic algorithms rely on purely random generation of code, which means it is impractical to apply them to more complicated, nonlinear games like Dota 2 or Fortnite, which have a multitude of players and combat features. This means deep learning, which could simplify such complex features, might be combined with genetic algorithms which can efficiently process those simplified features.

Given the potential of an intersectional study using genetic algorithms, game theory, and deep learning, I propose a simulated game of competing neural networks, iteratively evolved through artificial selection, to shed light on economic theory when the actors are irrational and developing. I hypothesize that game-theory based predictions of the networks' behavior will be mostly accurate, but certain rogue machine intelligences will think of successful strategies that game theory cannot explain.

Instrumentation and Variables

Tools

I use the Haskell programming language for this experiment (Jones, 2003). As a purely "functional" language, its inputs and outputs are fully predictable. This is opposed to a state-based or "imperative" language, such as Java, where internal randomness or errors in allocation of memory might crash a program mid-operation. Instead of risking memory loss in the middle of a crucial program execution, Haskell's compiler catches errors before the program runs, almost guaranteeing safety. Haskell's terse syntax and support for rapid composition of functions means relatively complex programs can be written very efficiently. This is important for the present project due to the multilayered nature of the algorithms involved. For example, being able to "link" functions together from input to output makes training a typical neural network much easier, because one can calculate the initial base "gradient" (essentially a small tweak to the network which makes it incrementally more accurate) with a single function and then pass the result to other functions which apply that gradient to the neural network proper, which is then passed along to another function which verifies if the new network has indeed improved. Especially in this context, where I layer deep learning, genetic algorithms, and game theory, being able to port functions together is vital, in a very literal sense, to explore interdisciplinary ideas.

In addition to the programming language, I will rely on a few software tools to help me develop my project. I use Atom and Emacs as my text editors because I am familiar with both and they have excellent inbuilt support for Haskell and enable fast editing using convenient keyboard shortcuts. For debugging and compilation, I rely on the standardized Haskell platform toolkit, including the Glasgow Haskell Compiler and specifications (Himmelstrup, 2006). Finally, version control is handled by git, a tool which helps keep track of changes and updates to one's projects.

Experimental Procedures

I have created a linear algebra library which handles matrix transposes, multiplication, and addition (Appendix A). Matrix operations are necessary to calculate a neural network's final output. I also programmed critical number-manipulation algorithms, such as the equations to calculate a neural network's accuracy and its "activation functions," which help make a network more flexible or stable in what it can represent by mapping its internal calculations to functions with unique shapes. Using matrix operations, I defined the architecture for the neural networks in Haskell (Appendix B). Each neural network is composed of "layers," with each layer being a collection of weights (a matrix) and biases (a vector). The value of the output of each layer is equal to the sum of the biases with the product of the input and weights (weights * input + biases). The neural networks I used for this test had two layers of dimensions (2,3) and (3, 1), respectively.

I used a process of artificial genetic evolution to improve each neural network. The weights and biases of each layer are the "genes" of each algorithm. Varying these quantities, one can dramatically alter what a neural network does. Thus, randomizing these values and testing whether the resulting network has high fitness can yield an iteratively better "gene" pool through selection. In order to simulate real-world unpredictability, I created three different selection processes to determine which neural networks can pass their "genes" on to the next generation. Regardless of the process used, the selected genes are then cloned such that the new gene pool is the same size as the original and then mutated by randomly modifying their weights and biases. In the first process, the genetic codes which achieved the highest fitness will be passed on to later generations. In the second process, the genetic codes which achieved the lowest fitness will survive. In the third process, an even mix of high and low performers will pass on their genes.

The game that I designed is much like the prisoner's dilemma, in that each player has a choice to defect (attack another player) or collaborate (remain peaceful). A player who chooses to defect (attack) another player who is not attacking back will gain ten points, while the peaceful victim will gain only one. However, if two players attack each other, both will gain only three points. A player who is peaceful will gain six points at nobody's expense. Each iteration of the game only has two players who are pitted against each other for ten turns.

Statistical Procedures

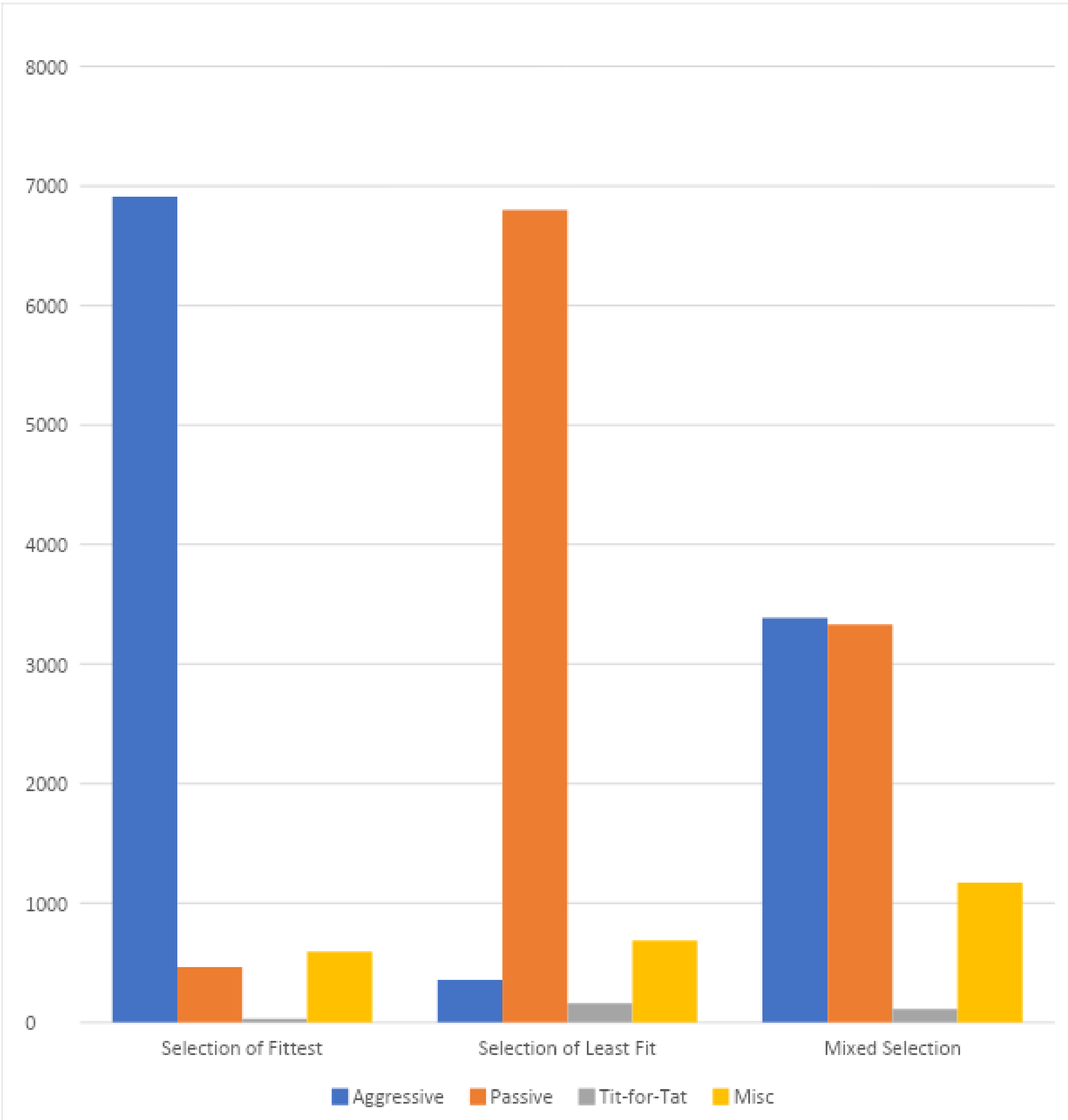
I simulated three different gene pools, each corresponding to one of the three selection processes I mentioned above. Each pool began with 500 sub-pools of neural networks, each consisting of 16 players. These sub-pools were insulated genetic environments, with each neural network within a sub-pool interacting only with other members of its sub-pool. Each sub-pool went through 15 cycles of a selection process, and the outputs were accumulated into a final pool of 8000 evolved networks. For each final pool, I tested the finalized neural networks to determine which of four strategies they had chosen: aggressive, passive, tit-for-tat, or miscellaneous. Aggressive networks attack the other player unconditionally. Passive networks are always cooperative. Tit-for-tat players will only attack if attacked. Miscellaneous strategies are all those which do not fall into the above three categories. Using this categorical data, I conducted a chi-squared test with three degrees of freedom to determine the goodness-of-fit between the observed distribution of strategies and the predicted distribution according to game theory. For selection process 1 (select most fit), game theory predicts 100% of players will be aggressive, because defection is the dominant strategy in the payoff matrix (Miller, 1985). For selection process 2 (select least fit), game theory predicts 100% of players will be passive. For the mixed selection process, game theory predicts 50% of players will be passive and the rest will be aggressive. I also gathered the final scores of all the evolved networks and found their means, medians, variances, and standard deviations. Finally, I calculated the percentage of sub-pools within each larger pool which contained at least one tit-for-tat player.

Results

In figure 1, each row represents a gathered statistic and each column represents a pool evolved with one of the three processes I mentioned above. Give the conditions of the game, and the fact that each game lasted ten turns, the minimum score one player could get was 10, and the maximum was 100. Figure 2 visually represents the distribution of strategies in each pool.

	Selecting most fit	Selecting least fit	Selecting mixed fitness
Number of aggressive players	6911	360	3387
Number of passive players	463	6798	3329
Number of tit-for-tat players	32	158	115
Number of miscellaneous players	594	684	1169
Goodness-of-fit	Null hypothesis rejected. P-value < 0.00001	Null hypothesis rejected. P-value < 0.00001	Null hypothesis rejected. P-value < 0.00001
Mean score	33.385	59.201	49.6656
Median score	30	60	37
Variance of scores	278.7525	202.7132	1068.6086
Standard deviation of scores	16.6959	14.2377	32.6896
Percent sub-pools with at least one tit-for-tat player	2.3%	7.4%	6.5%

(Figure 1: Table of statistics for each gene pool)



(Figure 2: Distribution of strategies)

Discussion

Analysis

The data seems to suggest that game theory, while moderately good at predicting how most actors will behave in aggregate, fails to explain or predict the development of alternative strategies. Based solely on a game theory analysis, the dominant strategy (the decision which yields more points regardless of the opponent's decision) should be the always defect: an aggressive strategy. Indeed, for the gene pool trained with a "survival-of-the-fittest" technique, the vast majority of players (6911 out of 8000) ended up in the Nash equilibrium of simultaneous defection. According to the data, the evolutionary process, on face, tends to favor uncompromising and decisive, instead of dynamic and responsive forms of intelligence. Similarly, the gene pool trained with a "survival-of-the-least-fit" process contained mostly passive players (6798 out of 8000), and the mixed pool was roughly split half-and-half between passive and aggressive players. Despite these initial appearances, the chi-squared tests I conducted all successfully rejected the null hypothesis that game theory predictions were a good fit for the observed data. This may lead us to believe that, while most players created through evolutionary techniques behave in a mathematically optimal way, there is a substantial and undeniable segment of the population which adopts counter-intuitive or unconventional strategies for decision-making.

Interestingly, large-scale analyses and tournaments involving the peace-war game indicate that tit-for-tat players, not aggressive ones, are the most successful overall (Miller, 1985). Given that genetic algorithms are often applied to discover non-linear, innovative solutions to complicated problems, we might expect the evolved gene pools to be composed largely of tit-for-tat players (Whitley, 1994). However, this is clearly not the case, as the highest proportion of tit-for-tat players in any of the three gene pools is still less than 2% of the population. One potential explanation is that the evolutionary process penalizes strategic experimentation early on by pushing the most aggressive players to the top at the expense of players who hesitate to attack. For example, a single game between a tit-for-tat player and an aggressive one will end up in favor of the aggressive player because they attacked first, gaining nine points relative to their initially peaceful opponent, and all subsequent turns consist of both players attacking each other and gaining net zero points. This would also explain why the selection processes which allowed lower-performing players to advance to the next generation also developed the greatest number of sub-pools containing tit-for-tat strategies (7.4% and 6.5% compared to 2.3% in the "survival-of-the-fittest" pool). If my hypothesis is correct, the data I collected might indicate that imperfections and variance in training techniques are not a source of inefficiency but rather innovations which increase efficiency in unexpected ways.

Furthermore, the training process that resulted in the greatest diversity of strategies (selection of the least fit) also ended up with the lowest variance in final scores, despite having the highest mean score of all the gene pools. It can be inferred that most of the players made the same decision (to cooperate) even if they used different decision-making heuristics. A tit-for-tat player will behave exactly like a passive player if its opponent is passive, as will a miscellaneous player which only attacks when both itself and its opponent choose to attack. This seems to be a reverse Nash-equilibrium in which each player refuses to attack, even if they would benefit immensely from defection, because gaining points actually decreases their chance of survival.

This can be analogized to the reverse of a tragedy-of-the-commons situation: each actor makes an unselfish choice which benefits the whole collective.

The fact that the “negative natural selection” resulted in the greatest number of optimal strategies and the best overall scores is incredibly counter-intuitive and may cast a troubling light on our competition-based society. Intellectual grading systems such as the college admissions process or the distribution of scholarships create a “battle royale” of highly motivated individuals to determine who gets access to financial and educational resources. These resources, like the fitness scores in a genetic algorithm, determine who gets to pass their “genes” on, both literally and figuratively: Wealthier and more knowledgeable people have a much greater social and intellectual influence over the communities they live in, and can afford to give their children many more opportunities than those from less privileged families. Perhaps these competitions have a similar effect to a “survival-of-the-fittest” selection process, in that they weed out intellectual diversity and unique, unprecedented skills early on because the kids who can’t initially score as high on tests or write as eloquently go on to receive less recognition and less access to resources. Furthermore, the results of these competitive processes might have a negative impact on society. Grades, money, and fame all tend to select for “aggressive” players who quickly seize on opportunities and do not hesitate to supersede their slower or less decisive peers. The end result, however, is a Nash-equilibrium of constant “defection:” Everyone chooses to compete all the time, resulting in stress and the punishment of those who fall behind. Disabled people with meaningful insight on life might be silenced because they are determined “unfit” by economic or academic rubrics. That being said, the analogies I have drawn are all products of conjecture based on limited data and should not be taken as fact.

Limitations

Because the scope of the game and the training techniques I am testing is very limited, I cannot confidently generalize my results and conclusions to most large-scale or complex applications. Furthermore, much of the variation of strategies I observed could be due to the inherent randomness of artificial selection and mutation. The dimensions of the networks I tested are also extremely small compared to sophisticated neural networks which attempt to model the behavior of complex animals or even people, which means they are probably not descriptive of real-life evolved intelligence which has had billions of years to develop. The game in question is also very discrete, with each player only having two options and each payoff being fixed. Such a predictable situation is rare if not nonexistent in the real world.

Next Steps

To answer my initial question about game theory in more depth, I want to introduce more methods for training, including genetic algorithms which incorporate “cross-over” – the equivalent of sexual reproduction – and more complex neural networks. I might also develop a more complicated game which better models a real-world scenario, such as a simulated multiplayer peace-war game with continuous instead of discrete decision options (such as being able to devote differing amounts of troops to any particular war-front). Finally, I would allow each gene pool to evolve hundreds or even thousands of times in order to achieve a more realistic degree of evolution. In order to reach a broader audience, I will create a graphical representation of my project in the JavaScript programming language, which is commonly used for animations on the web. In particular, the p5.js library has built-in tools for drawing dynamic figures and shaped. I will represent the game played by the AI’s as two nations on a map which can either choose to attack one another with missiles or extend a handshake (to be represented by nation-sized hands).

Works Cited

Burke, D. (2014). Why isn't everyone an evolutionary psychologist? *Frontiers in Psychology*, *5*, 910. https://doi.org/10.3389/fpsyg.2014.00910

Dar, P. (2018). Evolutionary Algorithm - The Surprising and Incredibly Useful Alternative to Neural Networks. Retrieved from https://www.analyticsvidhya.com/blog/2018/07/evolutionary-algorithm-perfect-useful-alternative-neural-network/

David, E., & Greental, I. (2017). Genetic Algorithms for Evolving Deep Neural Networks. *CoRR*, *abs/1711.0*. Retrieved from http://arxiv.org/abs/1711.07655

Gibbons, R. (1997). *An Introduction to Applicable Game Theory* (Technical Working Paper Series). https://doi.org/10.3386/t0199

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Himmelstrup, D. (2006). Interactive debugging with GHCi. In *Proceedings of the 2006 ACM SIGPLAN workshop on Haskell* (p. 107).

Jones, S. P. (2003). *Haskell 98 language and libraries: the revised report*. Cambridge University Press.

Levitt, S. D., & Dubner, S. J. (2011). *Freakonomics: A Rogue Economist Explores the Hidden Side of Everything*. HarperCollins. Retrieved from https://books.google.com/books?id=wNPnl5zYA-cC

Mahapatra, S. (2018). Towards Data Science - Why Deep Learning over Traditional Machine Learning? Retrieved from https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063

Miikkulainen, R., Liang, J. Z., Meyerson, E., Rawal, A., Fink, D., Francon, O., … Hodjat, B. (2017). Evolving Deep Neural Networks. *CoRR*, *abs/1703.0*. Retrieved from http://arxiv.org/abs/1703.00548

Miller, N. R. (1985). Nice Strategies Finish First: A Review of "The Evolution of Cooperation." *Politics and the Life Sciences*, *4*(1), 86–91. Retrieved from http://www.jstor.org/stable/4235437

Newkirk, V. R. (2016). Is Climate Change a Prisoner's Dilemma or a Stag Hunt? Retrieved September 20, 2018, from https://www.theatlantic.com/notes/2016/04/climate-change-game-theory-models/479340/

Osoba, O., & Davis, P. (2018). *An Artificial Intelligence/Machine Learning Perspective on Social Simulation: New Data and New Challenges*. Santa Monica, CA. Retrieved from https://www.rand.org/pubs/working_papers/WR1213.html.

Protopapas, M. K., Kosmatopoulos, E. B., & Battaglia, F. P. (2009). Coevolutionary Genetic Algorithms for Establishing Nash Equilibrium in Symmetric Cournot Games. *CoRR*, *abs/0905.3*. Retrieved from http://arxiv.org/abs/0905.3640

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing*, *4*(2), 65–85.

Yang, G. (2017). Game Theory-Inspired Evolutionary Algorithm for Global Optimization. *Algorithms*, *10*(4).