Modeling Learning and Cooperation in Iterative Games

Aleksey Chernobelskiy, Vineet Dixit, Agostino Cala, Siddharth Pandya, and Hector Javier Rosas
University of Arizona, Tucson, Arizona 85721 USA

In this paper, we describe the general framework of neural networks and how such frameworks can be adapted to model human game play and the learning that takes place during iterative games. We introduce a method of pre-processing game matrices in an effort to produce cooperative strategies in games with non-cooperative dominant strategies, such as the Nash Equilibrium solution to the Prisoner's Dilemma. We find that the introduction of the pre-processed matrix increases the probability that the network plays a cooperative strategy significantly when compared to the network behavior without pre-processing.

I. INTRODUCTION

Mathematical modeling for human learning and interaction has been studied for many years [1]. One way of studying human learning is through repetitive games. These games are usually taken from classical game theory, where mathematical criteria can be used to determine whether a player's action is the best response, given the player's beliefs about the payoffs in the game [2]. With the advent of computer learning, many studies have modeled the human learning from repetitive playing of the same game using neural networks ([3] and references there in). However, recently there has been interest in players of a game cooperating with other players to play a mathematically inefficient strategy (i.e. a strategy that is not a Nash equilibrium) [4]. In this paper we study a well known model with regret [3] and add a sympathy factor to encourage cooperation. A mathematical model of human learning and cooperation could bring insights to how humans interact in small groups. This ranges from better prediction of hypothetical games between humans to an improved understanding of behavioral finance.

Economic models often assume that when agents such as human players, are faced with decisions, they always act in their own best interest. Game theory takes this assumption a bit further, and attempts to analyze the outcomes of games played between players with limited or no information. To explain this further while motivating our research, we consider the well-known Prisoner's Dilemma [5].

The Prisoner's Dilemma poses the scenario as follows: Two men are arrested for a crime, but the police do not have strong enough evidence for a conviction. Immediately after the arrest, the individuals are put into separate rooms and are given the options to speak, or to remain silent. The police officer explains to each individual that if his partner betrays him while the individual decides to stay silent, the betrayer will go free, and the individual choosing to stay silent will serve a one year sentence. If both players remain silent, they will only be kept in jail for one month on a minor charge. If both players betray each other they will be kept for three months. To represent the outcomes for each player, we assign numerical values for the utility each person receives based on their allotted jail time represented in the payoff matrix in Table I. Thus, higher numbers in the table correspond to shorter sentence periods. For example, no jail time is represented by a 10 in the table and a jail time of one year is represented by a 2.

By observing the outcomes, we see that the betray action is strictly dominant for both players [5]. In other words, given any action of the other player, the other player would unanimously choose to betray the other. Thus, the cell with the (5, 5) payoffs is named the Nash Equilibrium [2]. Now suppose that the presented game is played iteratively with the same conditions imposed on each iteration.

A player is in Nash equilibrium, in the most general statement of the concept, when it is making the best decision it can make, taking into account the choices of the other people in the group, who are playing the game. It is important to note that the Nash equilibrium does not ensure the maximum payoffs for any individual player in the group. By making alliances, or targeting individual (or subsets of) players, certain players can maximize their payoffs.

Numerous articles and studies were used to construct the theoretical foundation, and implementation of the model described in this paper. Some of the most relevant ones are included in [6] and [3]. In [7], experimental data was gathered for different games played iteratively. Other works, including an additional piece by Erev and Roth, were instrumental in the execution of the model [6] which introduced reinforcement learning. This served as a point for comparison to more traditional models that did not use regret (e.g. [8]). Marchiori and Warglien, in [3], add to

Action	Player A remains silent	Player A confesses
Player B remains silent	(7,7)	(2,10)
Player B confesses	(10,2)	(5,5)

TABLE I: Payoff matrix for Prisoner's Dilemma.

previous methods of modeling human learning, by introducing regret factor into the decision making process, which is determined by the payoff discrepancy (if one exists) between the two players when they make their decisions. The factor of regret is incorporated due to the belief that it plays a role in a person's decision making. After choosing an action and experiencing its payoffs, a person would theoretically experience some sort of regret, in the form of either a positive or negative regret. A positive regret would correspond to a player reinforcing his belief in the efficacy of his decision, when it results in an outcome that benefits him more than it benefits his opponent. A negative regret would correspond to a player reinforcing a different decision because the outcome of the current decision was not the maximum payoff.

In some games, strategies that are beneficial for both players but are not Nash equilibrium strategies exist. In the Prisoner's Dilemma example, both players staying silent adds up to a larger total (sum of player A and player B's payoffs) than any other outcome. Choosing such courses of action has been observed in experiments, namely the tit for tat strategy [9] in which a player plays silent the first play, then chooses the opposing players previous choice for the next game.

Building on the model of [3], our goal is to realistically model human gameplay in a context of game theory in which cooperation of two or more players leads to a higher payoff for the cooperating group. To be clear, we are not interested in building a neural network that converges to optimal results the quickest. Instead, we hope to illuminate decision making mechanisms that may be at work when human agents pursue co-operative strategies that are not anticipated by analysis by means of the aforementioned game theoretical, or economic modeling philosophies.

In section II we introduce the neural network model used to obtain our results, and show its performance using the well known Prisoner's Dilemma game. We then introduce cooperation into our model in section III, which allows players to have sympathy [Eq. (7)] toward their opponents' payoffs. Detailed results and discussion can be found in section IV. We then close the paper in section V with conclusions and possible further research.

II. NEURAL NETWORK MODEL

The type of artificial neural network we employ to simulate human learning is a single-layer feed-forward network, also known as a simple perceptron. The simple perceptrons consists of a vector of input values ($\vec{x}$ containing n individual inputs x_j), a vector of output values ($\vec{y}$ containing m individual outputs y_i), and a m by n matrix of coupling weights w_{ij}. A weight matrix with no elements statically defined to be 0, indicates a fully connected network. Output neurons often function as threshold logic units (TLUs) [10], that evaluate an activation level, or local field [11, 12] l_i for an output node y_i by,

$$l_i = \sum_{j=1}^{n} w_{ij} x_j. \tag{1}$$

An activation threshold θ is then subtracted from the local field. The resulting quantity serves as the input to a transfer function, which can be the signum function, heaviside function, or a function with similar characteristics; the output of this function serves as the value of an output node of the network. In the most general case, each output node can have its own transfer function ($\mathcal{F}_i$) [13] and threshold (θ_i). So, generally, the value of an output node y_i is

$$y_i = \mathcal{F}_i \left(l_i - \theta_i \right). \tag{2}$$

Taking the example of the signum function, local fields exceeding the threshold, θ would result in output values of 1, and local fields whose values were below the threshold would result in output values of -1. The binary outputs of TLUs lend themselves well to classification tasks [13]. To this end, training algorithms such as the perceptron learning rule [13], the gradient descent based delta rule [10], and back propagation algorithms [13] are traditionally used to iteratively determine the values that will populate the neural coupling matrix w_{ij}. These training algorithms rely on training sets, which consist of input vectors ($\vec{\chi}^p$, the p^{th} input vector of the training set) which resemble the type of input the network will be classifying and a corresponding desired output vector ($\vec{\gamma}^p$, the desired output vector corresponding to $\vec{\chi}^p$)[13].

Although the network used in this paper to model learning resembles such perceptrons in terms of topology, it differs in function, as it is not used to classify inputs, but rather demonstrate the learning rate and trend a player exhibits when playing a game, under the constraints of a particular weight update formula, which hopefully captures the factors and motivations which humans take into account when making repeated decisions.

In our model, each player's behavior is governed by a simple perceptron. The inputs of the perceptron are populated with both the player's and the opponent's payoffs for every possible combinations of actions, as described in the payoff

matrix of the game. So, in a game with k actions, $n = 2k^2$, since there are k^2 actions for both the player and the opponent. Specifics of our model will be discussed in part A of this section.

The algorithm used for this model involves a turn based repetition calculation in which the initialized values are randomized and new generated values are based on previous values. Qualitatively, this signifies a player who has no previous experience in the game and relies on the experience gained from repetitive play to inform his decisions. To help explain and evaluate the algorithm the Prisoner's Dilemma example will be used in the model. The model can be broken down into 6 parts:

1. Randomization of Initial Inputs and Weights
2. Generation of Outputs
3. Decision from Stochastic Choice Rule
4. Make an Action
5. Check Action against Best Possible Action
6. Update Weights and Repeat Process

For part 1, the initialization of inputs and weights, the inputs are the payoffs of the game matrix while the weights are initially randomized as a uniform number between zero and one. Figure 1 provides a visual representation of the network architecture. The circles on the right represent inputs (or payoffs), and the circles on the left represent outputs. The w_{ij} values, which are associated with each line linking an input node to an output node, represent weighting factors; initially, these weighting factors are assigned random values. Given the initial values, the outputs can be calculated by

$$y_i = \tanh\left(\beta * \sum_{j=1}^{N} w_{ij} x_j\right). \tag{3}$$

Equation (3) takes the place of $\mathcal{F}_i$ in equation (2) for all output values i. The threshold θ_i is set to 0, and a scaling constant β (one of the scaling parameters that influence the learning rate) multiplies the local field l_i. The purpose of the sigmoidal transfer function tanh is to transform the outputs of the network into a simplified bounded value between -1 and 1. The function is monotonically increasing, and saturates for large input values, which prevents an overflow on the output values of the system, and thus reduces the runtime and computational cost of the algorithm. Furthermore, the sigmoidal transfer function's linear regions correspond to a susceptibility to positive or negative regret terms that disappears as the magnitude of the weights increase (accumulation of regret in a single direction); we feel that this property qualitatively resembles the decision making process in humans. However, many other sigmoidal transfer functions can be used. The values returned by the sigmoidal transfer function, the network outputs, can be viewed as the propensities (but not probabilities, since the sum of the outputs can exceed 1) to choose a certain action.

The decision process is based on our Stochastic Choice Rule, or deciding the action based on calculating uniform probabilities and a random choice. The output vector is normalized and probabilities are calculated using

$$P_i = \frac{e^{y_i}}{\sum_{i=1}^{N} e^{y_i}}. \tag{4}$$

The next step involves comparing the action chosen to the best possible choice, and is where regret comes into consideration. If the action chosen is the best possible action, the ex-post best response value ($t_i(a^{-k})$) takes on the value of +1; if the action chosen is not the best possible value the ex-post best response takes on a value of -1. In addition, the regret value is calculated. In this paper we compute the regret of a player as a function of the payoffs:

$$R_k(a^k, a^{-k}) = x_m^k(a^{-k}) - x^k(a^k, a^{-k}) \tag{5}$$

In equation (5) $x_m^k(a^{-k})$ is the maximum payoff that the player could experience given the opponent's action a^{-k}, and $x^k(a^k, a^{-k})$ is the payoff that the player experienced given his action a^k and his opponent's action a^{-k}.

Given the previous values, the weights can be changed for the succeeding steps. The change weight function is the most important part of the model's architecture, as it takes into account all the properties of both the input and output nodes. The equation for the change in weight is

$$\Delta w_{ij} = \lambda^2 * [t_i(a^{-k}) - y_i] * R^k(a_m^k, a^{-k}) * x_j. \tag{6}$$

In equation (6), λ is the second parameter (along with β) that determines the learning rate of the network, and $t_i(a^{-k})$ is the ex-post best response indicator. The term $[t_i(a^{-k}) - y_i]$ is characterized by [3] as "the distance from

i,j	1	2	3	4	5	6	7	8
1	0.4738	0.4756	0.9596	0.7977	0.9122	0.2933	0.5041	0.2830
2	0.3563	0.6710	0.0891	0.5908	0.1011	0.0516	0.7684	0.2254

TABLE II: Player A's initial weight matrix (w_{ij}).

i,j	1	2	3	4	5	6	7	8
1	0.3313	0.7374	0.3825	0.9653	0.1320	0.3830	0.2868	0.5352
2	0.4533	0.5099	0.9055	0.6283	0.6183	0.9912	0.7062	0.1932

TABLE III: Player B's initial weight matrix (w_{ij}).

ex-post best response," which is an indicator of the urgency of the change. Because the outputs y_i are mapped to the interval (-1,1) by equation (3), the values of the term $[t_i(a^{-k}) - y_i]$ will fall on the interval (-2,2). Consider that the i^{th} action has an output propensity y_i close to -1, which suggests that it will rarely by chosen by the network (per equation (4)); if it turns out that $t_i(a^{-k})$ is 1, indicating that the network has determined the i^{th} action to be the action with the highest payoff given the opponent's action, the term $[t_i(a^{-k}) - y_i]$ will be close to 2, which will work to increase the weighting factors to the output node, and thus increase the propensity of playing the i^{th} action. This indicates the network's recognition of the fact that given the ex-post best response, and the current state of the output node y_i, much change is needed, as the i^{th} action, that was deemed by the ex-post vector to be the best action, is currently unlikely to occur. If however, assuming the same output propensity y_i close to -1, the network determines that the i^{th} action is not the best action, resulting in $t_i(a^{-k})$ being equal to -1, the term $[t_i(a^{-k}) - y_i]$ will be close to 0, resulting in attenuated change to the weights associated with that output node; this indicates the network's recognition of the fact that given the ex-post best response, and the current state of the output node y_i, much change is not needed, as the i^{th} action is currently unlikely to occur.

The payoff associated with w_{ij}, x_j, is multiplied in the weight change equation to make the coupling strengths change in proportion to the payoff. If the payoffs are very large in magnitude and a sigmoidal transfer function $\mathcal{F}_i$ is being used, a scaling factor can be applied to the local field l_i to prevent premature saturation. The input x_j multiplies the preceding terms to incorporate what [3] refer to as "input saliency". The inclusion of this term scales the Δw_{ij} term proportional to the payoff, which means that changes with regards to propensities of actions involved in the very lucrative payoff scenarios will occur rapidly in ths system. Equation (6) can be intuitively thought of as: "adjustment = learning rate $*$ distance from ex-post best response $*$ regret $*$ input saliency" [3].

A. Prisoner's Dilemma Example

In this section, we demonstrate how we, following the exampled in [3], adapt the Prisoner's Dilemma game to the single perceptron neural network architecture. Self interested players, whose beliefs about the game as can be seen in the game matrix presented in Table I, will confess as it results in a higher payoff regardless of their opponent's decision. This can be seen by observing a player's best outcome given another player's action. To find this Nash Equilibrium, first suppose that Player B remains silent. In this situation, Player A will always choose to confess. Now suppose that Player B confesses. Once again, in this situation Player A chooses to confess. This suggests that Player A will always choose to confess. But now, since Player B knows that Player A will always choose to confess, Player B will confess as well since that has a higher payoff.

As a neural network, the Prisoner's Dilemma governed by the payoff matrix in Table I, is illustrated in Figure 1 where the payoffs are represented in the left circles for both players, w_{ij} represent the randomized weights, and y_i is the output to Equation (3). We then use randomly chosen numbers using uniform distribution between 0 and 1 (MATLAB "rand" function) to fill players' weight matrices. The weights for each player are presented in Table (II) and Table (III). These values affect the propensities to play a given action, which is discussed in detail below. Now, using Equation (3), we get the resulting Table (IV). Using Equation (4) we obtain the Table (V).

To determine a player's action, we take one sample for each player from the Uniform $[0, 1]$ distribution. Suppose that we picked 0.8976 for Player A and 0.1263 for Player B. Since 0.8976 is an element of the interval [0.5040,1], Player A decides to confess, which results in an ex post evaluation of +1, as it was the correct decision, given the advantage of confessing explained earlier in this section. Since 0.1263 is an element of the interval [0,0.4960], Player

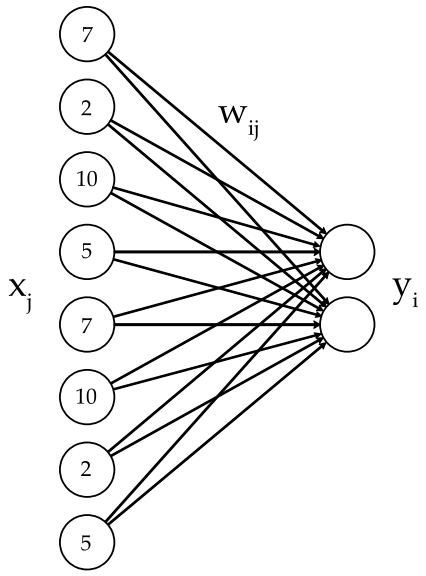

FIG. 1: Neural network architecture for the learning model.

	Player A	Player B
Propensity to remain silent	0.9917	0.9671
Propensity to confess	0.9755	0.9973

TABLE IV: Propensitites to play (outputs) generated using Equation (3).

B chooses to stay silent, which results in an ex post evaluation of -1 since remaining silent was not the player's best decision, given his understanding of the game. We use equation (6) in order to compute the new weight ($w_{i,j}$) for Player A (Table VI) and Player B (Table VII). By adding the weight change (δw_{ij}) matrices for players A and B respectively, represented by Tables VI and VII (the Δw_{ij} tables) with the original weight matrices for players A and B respectively, Table II and Table III (the current weights) respectively, we obtain the new weights for Player A (Table VIII) and Player B (Table IX). Using these new weights, another iteration of the algorithm is run.

III. INTRODUCTION OF COOPERATION

In attempt to create a network that would act with the objective of maximizing not only its own success, but the success of its 'opponent' (or friend, now), we implemented a procedure to modify the static game matrices to reflect the mindset of a player with such a philosophy. The advantage of such an approach, if successful, is that the weight adjustment formula can remain unchanged, and the pre-processing needs to occur only once, and will not add to the computational cost or runtime. Our implementation of pre-processing is intended to reward players for actions which have a minimal discrepancy of rewards between players. To simulate a sympathetic player, a new parameter, 'sympathy,' is introduced

$$P_A(i,j)_f = P_A(i,j)_0 * e^{-sympathy*|P_A(i,j)_0 - P_B(i,j)_0|}, \tag{7}$$

	Player A	Player B
P(Remaining Silent)	0.5040	0.4924
P(Confessing)	0.4960	0.5076

TABLE V: Probabilities of playing an action computed using Equation (4).

i,j	1	2	3	4	5	6	7	8
1	-0.0837	-0.1195	-0.0239	-0.0598	-0.0837	-0.0239	-0.1195	-0.0598
2	0.0010	0.0015	0.0003	0.0007	0.0010	0.0003	0.0015	0.0007

TABLE VI: Player A's weight change (Δw_{ij}) matrix.

i,j	1	2	3	4	5	6	7	8
1	-0.0826	-0.0236	-0.1180	-0.0590	-0.0826	-0.1180	-0.0236	-0.0590
2	0.0001	0.0000	0.0002	0.0001	0.0001	0.0002	0.0000	0.0001

TABLE VII: Player B's weight change (Δw_{ij}) matrix.

where $P_A(i,j)_f$ is the modified/processed payoff for player A, when it chooses action i, and its opponent chooses action j. $P_A(i,j)_0$ is the un-modified/standard payoff for player A, when it chooses action i, and its opponent chooses action j. This parameter is responsible for the sensitivity of the player to the discrepancy between its payoff, and the payoff of its friend/opponent. Before any games are simulated, the payoff value associated with every element of a player's payoff matrix is weighted by an exponential that decays as the discrepancy between the player's payoff and its opponent's payoff increases. If there is no discrepancy between the players' payoffs for a given combination of actions, then the payoff values in the cell corresponding to that combination is unchanged. The algorithm does require that a given player have access to the other's payoff matrix (in addition to its own). Information about the opponent's payoff is essential in empowering the player to assess the payoff discrepancy that results from any given action; if he is a sympathetic player, an assessment of the payoff discrepancy will change the player's beliefs about the game, and consequently his best response.

Given two sympathetic players (figures 3 and 5) with adequately high sympathy values, if by chance, player A chooses to remain silent consistently, player B's beliefs about the game outcomes will lead him to consider remaining silent the best option. Recall that the initial game behavior is determined by random weights connecting the payoffs inputs with the propensity outputs, and behavior trends (if they exist) emerge only after repeated play. Player B's propensity to confess when player A chooses to remain silent consistently, is greatly reduced as a result of the application of equation (7) to the large payoff (of 10) associated with the decision to confess when A remains silent. The option to remain silent, as player A had done, appears more appealing to player B, as the payoff (of 7) associated with being jointly silent with player A has not changed as a result of the application of equation (7). Player B's sympathetic behavior, seeking a low payoff discrepancy, can be characterized as cooperative behavior. However, if by chance, player A consistently chooses to confess in the first few rounds of play, player B will continue seeking a discrepancy minimizing course of action, and choose to confess as well. This behavior is present in the case of the unsympathetic players (figure 2), but is no longer the only equilibrium, when both players are sympathetic. It is unlikely that player A will consistently choose to confess, because his propensity to pursue the high payoff associated with that action is reduced by equation (7) due to the action's large payoff discrepancy.

If, however a sympathetic player B is playing with an unsympathetic player A, who pursues the high payoff associated with confessing, then he (player B) will exhibit remnants of the behavior from the unmodified Prisoner's Dilemma, but will not be committed to that behavior as if it was the optimal choice (figure 4). The sympathetic player B, when pitted against such an unsympathetic player, believes a better outcome lies (influenced by equation (7)) in the event that both players remain silent, as he is not blinded by the large payoff (of 10) that locks the unsympathetic player A in his tendency to confess. Player B's behavior can be considered risky, or naive, when pitted against an unsympathetic opponent, as player B will occasionally have a higher propensity to remain silent in the hopes of attaining an outcome that an uninvolved observer to the game will recognize is not probable given the unsympathetic opponent's beliefs.

i,j	1	2	3	4	5	6	7	8
1	0.3902	0.3561	0.9357	0.7380	0.8285	0.2694	0.3846	0.2232
2	0.3573	0.6725	0.0894	0.5915	0.1022	0.0519	0.7698	0.2261

TABLE VIII: Player A's weight matrix (w_{ij}) after one iteration of the game.

i,j	1	2	3	4	5	6	7	8
1	0.2487	0.7138	0.2645	0.9062	0.0494	0.2650	0.2632	0.4762
2	0.4534	0.5099	0.9056	0.6283	0.6184	0.9914	0.7062	0.1933

TABLE IX: Player B's weight matrix (w_{ij}) after one iteration of the game.

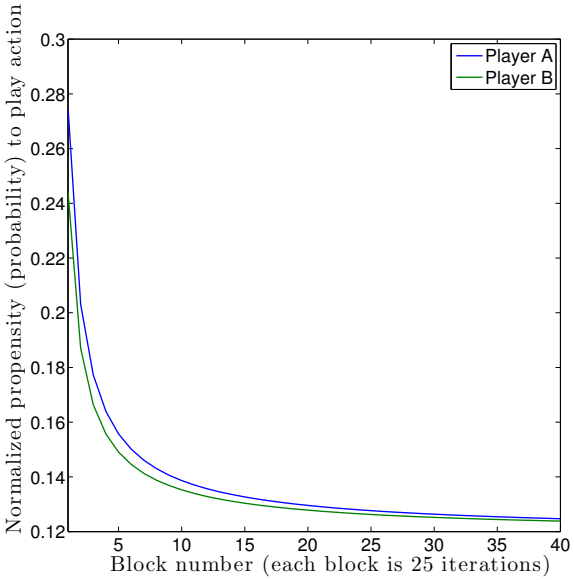

FIG. 2: Plot of two players playing the Prisoner's dilemma game ($\beta = 0.1$ and $\lambda = 0.1$), governed by the payoff matrix in Table I. The cumulative average probability to remain silent is shown. The sympathy parameters for Player A and Player B are set to 0.

IV. RESULTS AND DISCUSSION

We studied the effects of pre-processing most extensively with the Prisoner's Dilemma (whose payoff matrix is shown in table I). Recall, that with an unmodified game matrix, the equilibrium of action occurred when the players settled to a state in which their propensity to play action 1 was low; the players found it in their best interest to confess their crime (action 2), as is shown in figure 2. When we implemented pre-processing (using equation (7)), with a 'sympathy' parameter of 0.5 (figure 3), we obtained graphs that indicated that the players did not reach a stable equilibrium immediately; they were more open to choosing actions that previously had a prohibitive payoff discrepancy from the choice they had deemed it best to settle on (confessing their crime).

Note: The plots denoted as non-cumulative propensities to play an action are representations of the evolution of the network over successive iterations. These plots capture the state of the output nodes of neural network at each iteration; values plotted on the graphs are normalized propensities, the values of the output nodes converted into probabilities by the stochastic choice rule (equation (4)).The plots denoted as cumulative average propensities represent the average of the network outputs processed by the stochastic choice rule (equation (4)), until that iteration. The plot represents the values of a cumulative moving average.

Because the choices are made randomly (and, without a seed), trends in output graphs differed. However, we observed that many sets of trials resulted in both players tending to remain silent (as opposed to the previous equilibrium), because the payoff when they confessed had been reduced, in the case when the other player remained silent (see figure 3). Furthermore, we observed that the players' actions, as was the case in the un-modified simulation of this game, tracked each other. However, this tracking did not force an equilibrium. Without pre-processing of the input matrix, an equilibrium was forced when a player had a high propensity to confess (a low propensity to remain silent); the opposing player was forced to confess, as it was his best recourse because remaining silent, had a payoff discrepancy of 8.

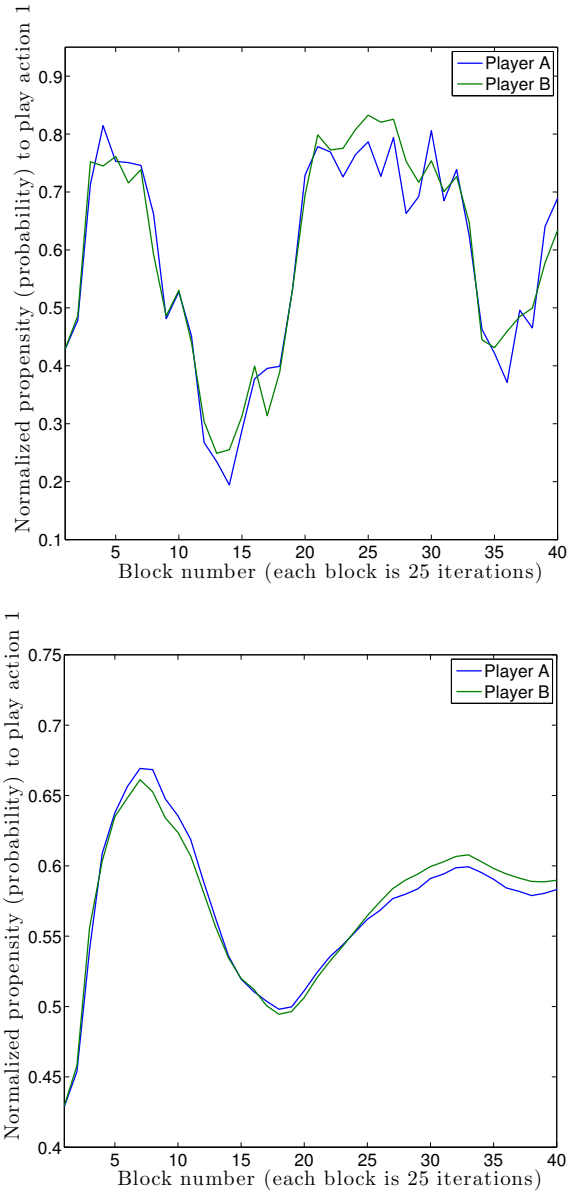

FIG. 3: Plot of two players playing the Prisoner's dilemma game ($\beta = 0.1$ and $\lambda = 0.1$), where both players have the same sympathy parameter (0.5). The non-cumulative probability to remain silent is shown above, and the cumulative average probability to remain silent is shown below.

It appears that the pre-processing on the matrices modifies the response in the network, and promotes the search for courses of action in which there is less payoff discrepancy between players. Recall, action 1 is the action to remain silent. There are two periods ($N < 250$ (corresponding to blocks below 10), and $N > 500$ (corresponding to blocks above 20)) in figure 3 when the two players tend toward a secondary equilibrium, in which both players remain silent and receive the same payoff.

In the interest of modeling human learning, we determined that the sympathy parameter could be used to profile agents, and predict how very sympathetic agents would interact with un-sympathetic agents. Figure 4 shows that when an unsympathetic agent (player a, in blue) is paired with a sympathetic agent agent (player b, in green), the latter expresses a relatively low propensity to remain silent compared to the pairing when both agents were sympathetic (figure 3); the sympathetic agent's strategy is punctuated by bursts of attempting to change its strategy, which are not tracked or mirrored by the unsympathetic agent, which in turn does not allow for a dramatic departure from the un-processed (figure 2) network behavior over time: the cumulative average propensity to remain silent is at most a third of what it was when both agents were sympathetic (figure 3), and not much higher than the network behavior

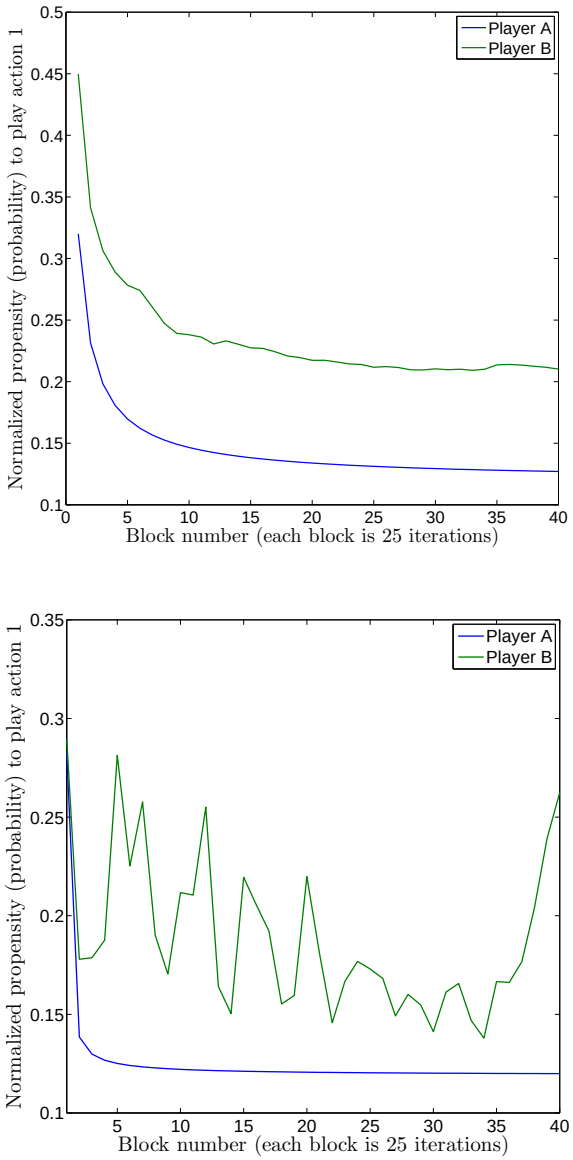

FIG. 4: Plot of two players playing the Prisoner's dilemma game ($\beta = 0.1$ and $\lambda = 0.1$), where Player A does not express sympathy ($a_{symp} = 0.01$) and Player B does express sympathy ($b_{symp} = 0.50$). The non-cumulative probability to remain silent is shown above, and the cumulative average probability to remain silent is shown below.

without the addition of cooperation (figure 2).

To investigate whether or not there was a critical value of sympathy that would serve as a threshold to reach a cooperative equilibrium solution (both players remaining silent) other than the original equilibrium (both players confessing), we ran multiple simulations with different sympathy values to better understand the relationship between output propensities and sympathy values, when the payoff matrices underwent pre-processing.

With all other parameters fixed, it was observed that the effects of adding sympathy saturated after a certain point. It appears that decreasing the payoffs for the situation in which one player remained silent while the other confessed was enough to modify the equilibrium present when sympathy was absent from the game (or negligible, as was the case

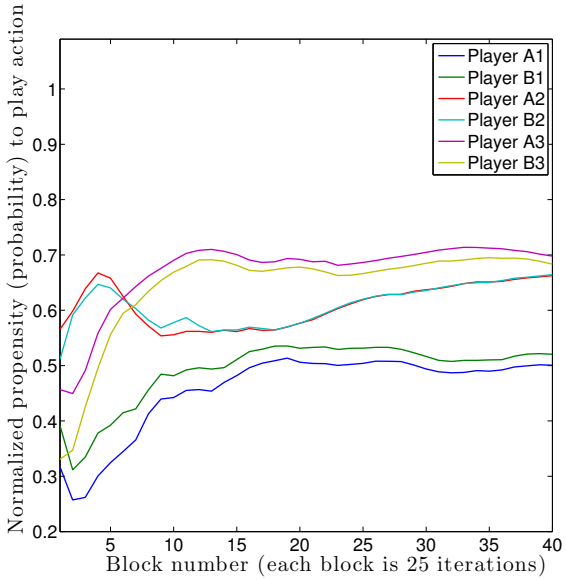

FIG. 5: Plot of two players playing the Prisoner's dilemma game ($\beta = 0.1$ and $\lambda = 0.1$), where the players sharing the same sympathy parameter. The first pair of players (A1/B1) both have a sympathy parameter of 0.1. The second pair of players (A2/B2) both have a sympathy parameter of 0.4. The third pair of players (A3/B3) both have a sympathy parameter of 0.7. The cumulative average probability to remain silent is shown.

when sympathy = 0.01). As we increased the sympathy parameter beyond 1, the outputs of average propensities had similar shapes and trends, as the exponential function we used had effectively decreased the payoffs for the actions in which players chose different actions (the (10,2), or (2,10) choice) to 0 for both players (since e^{-8} is less than one-thousandth).

First, we simulated the behavior of two (identically) sympathetic agents playing the Prisoner's Dilemma game with different sympathy values. In figure 5, we see that the cumulative average propensity to remain silent does not increase linearly with the sympathy parameter, but saturates, as the negative exponential that makes action combinations with payoff discrepancies unattractive saturates (approaching 0). We see that with a sympathy value of 0.1, the sympathetic network's propensity to remain silent is almost five times greater, than the unsympathetic network (figure 2).

Finally, the pairing of two agents with a discrepancy in sympathy parameters was explored; figure 6 shows that a small discrepancy between the sympathies of the players can still yield favorable results (see trial 2 in figure 6), in terms of cooperative behavior. Such results, were contingent on the least sympathetic agent's sympathy parameter to be above a threshold of 0.05; otherwise, the results were similar to the pairing detailed in figure 4. The sweep of sympathy values with non-identical sympathetic agents also shows that the starting propensity to play a given action (determined by the randomly generated weights at the time of network instantiation) may have an effect on the final outcome of the game.

As mentioned earlier, in this section, the random nature of the choices means that the conclusions drawn from the graphs do not represent absolute, or even consistent trends in the sympathetic network's behavior. The figures shown here are, a representative sample of the dominant trends that we observed after simulating the networks multiple times. A more extensive numerical study is needed to describe possible trends. This includes sampling many more nodes of the parameter space than was feasible for this study.

V. CONCLUSIONS AND OUTLOOK

In this paper, we have built a mechanism that effectively engenders sympathy in actors, on an existing neural network model. This resulted in cooperation in iteratively played games, whereas before, the model showed convergence to selfish strategies that merely maximized a single player's payoffs. An advantage of the pre-processing approach to

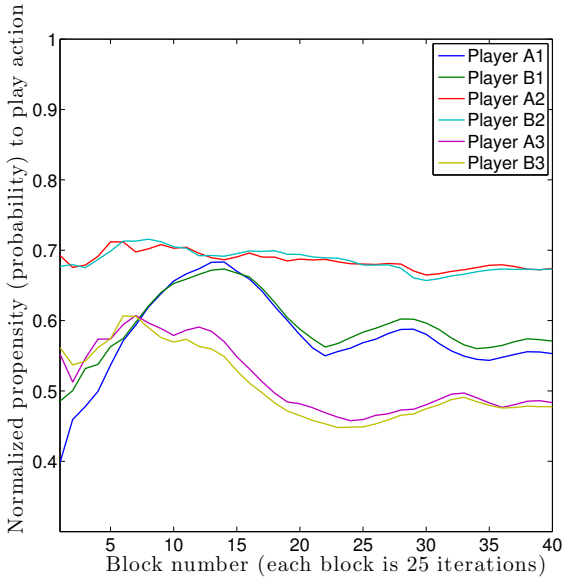

FIG. 6: Plot of two players playing the Prisoner's dilemma game ($\beta = 0.1$ and $\lambda = 0.1$), where the players are simulated with non-identical sympathy parameters. Players A1, A2, and A3 have a sympathy parameter of 0.1. Player B1 has a sympathy parameter of 0.1; Player B2 has a sympathy parameter of 0.4; Player B3 has a sympathy parameter of 0.7. The cumulative average probability to remain silent is shown.

modeling cooperation is that, given normalized payoff matrices, the network can simulate the behavior of agents with different levels of sympathy and selfishness in games with variable payoffs as well.

The pairing of sympathetic and selfish agents produces unintuitive results (see figure 6) after multiple iterations of a game, and can be used to model the interactions of actors with sympathetic and selfish tendencies, based on empirically collected data, by mapping a parameter vector (whose elements are the learning rate parameters β and λ and the sympathy parameter) to each agent through a sweep of the parameter space. Given empirical data of two agents playing an iterative game, the model's parameters can be swept in the usable ranges, and the parameters which produce the smallest mean-squared deviation (or equivalent curve-fitting metric) can be used by the model as character profiles, which can then be used to simulate the behavior of those agents in different games.

While our model was created to make a smart sympathetic plaer, the unintuitive behavior of the various pairing (figure 5) casts doubt upon the notion that our model has created a realistic model of a sympathetic human player. In section 3, we detailed the logic behind our expectations for the network's behavior. In some trials, this expected behavior emerges. As we mention in section 4, we observed variegated trends in the cumulative network tendencies when performing sweeps of the sympathy parameter unilaterally (akin to figure 6) and bilaterally (akin to figure 5); while it may be illuminating to characterize these trends and compute their relative frequency of occuring, such an undertaking is beyond the scope of this project.

Acknowledgments

This project was mentored by Scott Hottovy, whose help is acknowledged with great appreciation. Support from a University of Arizona TRIF (Technology Research Initiative Fund) grant to J. Lega is also gratefully acknowledged.

[1] A. Newell, J. Shaw, H. Simon, Psychological Review (1958)
[2] D. Fudenberg, J. Tirole, *Game theory* (MIT Press, Cambridge, MA, 1991)
[3] D. Marchiori, M. Warglien, Science (2008)
[4] W. Press, F. Dyson, PNAS (2012)
[5] R. Gibbons, vol. 54 (FT Prentice Hall, 1992). URL `http://www.amazon.com/dp/0745011594`
[6] I. Erev, A.E. Roth, The American Economic Review **88**(4), pp. 848 (1998). URL `http://www.jstor.org/stable/117009`

[7] D. Malcolm, B. Lieberman, Psychon. Sci. (1965)
[8] T.H. Ho, C.F. Camerer, J.K. Chong, J. Econom. Theory **133**(1), 177 (2007). DOI 10.1016/j.jet.2005.12.008. URL http://dx.doi.org/10.1016/j.jet.2005.12.008
[9] R. Axelrod, W. Hamilton, Science (1981)
[10] A. Blais, D. Mertz, An introduction to neural networks pattern learning with the back-propagation algorithm. Tech. rep., IBM (2001)
[11] *Neural Networks: A Comprehensive Foundation* (New Jersey: Prentice Hall, 1999)
[12] R. Grzeszczuk, D. Terzopoulos, G. Hinton, SIGGRAPH (1998)
[13] B. Krse, B. Krose, P. van der Smagt, P. Smagt. An introduction to neural networks (1996)