

Machine Learning and Mixed Strategy Games
Edmund Lam, Avi Patni

1 Introduction

A Mixed Strategy Game within this paper refers to a subset of strategic games within game theory. A game is defined to have P players and M moves per player. Each player can play any one of their M moves every round, and scores for each player are calculated by accessing the payoff matrix with the results of all their opponents. The payoff matrix is represented by a P -dimensional hypercube matrix with side length M . Players can be conceptualized as being arranged in a circle, each player is the first in their perspective, which increments for player to their right. Thus, player P_i 's' ($1 \leq i \leq P$) score is calculated by accessing the payoff matrix with the coordinates $(P_i, P_{i+1}, \dots, P_P, P_1, P_2, \dots, P_{i-1})$. Figure 1 shows an example of how payoffs could be calculated for each player. Players 1 and 2 make moves P_1 and P_2 , respectively. Thus, the players will receive scores $M_{<1,2>} = 3$ and $M_{<2,1>} = 3$, respectively.

Figure 1: Example 2×2 payoff matrix with sample moves and resulting scores.

Payoff Matrix	(P_1, P_2)	Payoffs
$M = \begin{bmatrix} 2 & 1 \\ 3 & 0 \end{bmatrix}$	$(1, 2)$	$(1, 3)$

An artificial Neural Network (NN) is a computing system loosely inspired by the biological neural networks found within animal brains. A neural network is composed of a number of interconnected groups of nodes, such that each node can transmit signals to other nodes within the Neural Network. This can be most simply represented by layers, each with a number of nodes, where two adjacent layers of nodes can be represented as a complete bipartite graph. Each connection between two nodes represents a weight $w_{<i,j>}$

Given layers x (length X) and y (length Y), node x_i ($1 \leq i \leq X$) will be connected to node y_j ($1 \leq j \leq Y$) with a weight $w_{<i,j>}$. This represents the weight held by x_i on the resulting value of y_j , which is calculated by summing each node x_i by the weight of its connection, $w_{<i,j>}$, which can be represented using the following formula, which will be run for every node y_j in layer y :

$$y_j = \sum_{i=1}^X x_i \cdot w_{<i,j>}$$

A set of weights w can thus be represented as an $X \times Y$ matrix, where $w_{<i,j>}$ still represents the weight between x_i and y_j . w can thus be multiplied by the first layer x ($1 \times X$) to calculate the next layer, y ($1 \times Y$). Furthermore, an added vector b of dimensions $1 \times Y$ is added, which will be added to y after it is calculated. This allows the vector y to be skewed, and results in the simple formula $y = x \cdot w + b$, where w represents the matrix of weights between x and

y. For neural networks with more than two layers, this is done sequentially to calculate each layer in the network. The first and last layers are vectors that determine the input information, and the output, respectively.

Our project utilizes artificial neural networks and natural selection to optimize strategies for mixed strategy games and maximize individual returns. By providing neural networks solely with information on the moves of each player for the last N rounds, and using natural selection based on the resultant score after a number of rounds with different players, we can observe the strategies emerging in the best neural networks, and test how they react to certain situations. We were particularly keen on observing more complicated payoff matrices, such as when $P > 2$, $M > 2$, as well as games similar to the Prisoner's Dilemma, where nash equilibrium exists, but is not necessarily optimal in the long term.

2 Methods and Procedure

3 Results

4 Discussion