Vanilla configuration

To begin our project, we settled on a basic set of parameters which came to be known as the vanilla configuration. Features of the vanilla configuration include gameplay on a reversi board with size 8x8. To generate the inputs given to neural networks, an array was created with each entry corresponding to a piece on the board. If the space is occupied by a friendly tile, the value of the entry is 1. For enemy tiles, we use -1 and empty tiles are assigned 0. The size of the population is 10x10 arranged in a hexagonal pattern. Each organism plays 6 neighbors, and organisms on the edge play organisms along the opposite edge. This creates a Taurus shaped population. Theopulation. Weights are generated at the start of an evolution by using a uniform distribution between -1 and 1. Mutation occurs with a 1% chance, and will generate a brand new weight, using the same uniform distribution, and reassign the old weight to the new weight. Each organism must choose another organism to crossover and produce a child organism. In order to select the parent from 6 surrounding neighbors, they are assigned probabilities based on their fitness scores calculated earlier. The lowest scoring neighbor is thrown out and his fitness score is subtracted from the remaining neighbors scores. This new value is directly proportional to the probability it will be selected as a parent.

Of these properties, several settings will be varied for each future configuration. se organisms will only be allowed to search one move ahead of the current board state, thus no minimax is implemented within their gameplay. In order to rate our organisms, each is assigned a fitness function. To calculate the value, we use a combination of wins, losses, ties, and the number of pieces controlled by a player at the end of a match. Wins are awarded 64 points, losses earn 0 points, ties are awarded 32 points, and the number of pieces controlled is added to this sum. Our initial configuration also includes settings for mutation and initialization of weights for the p