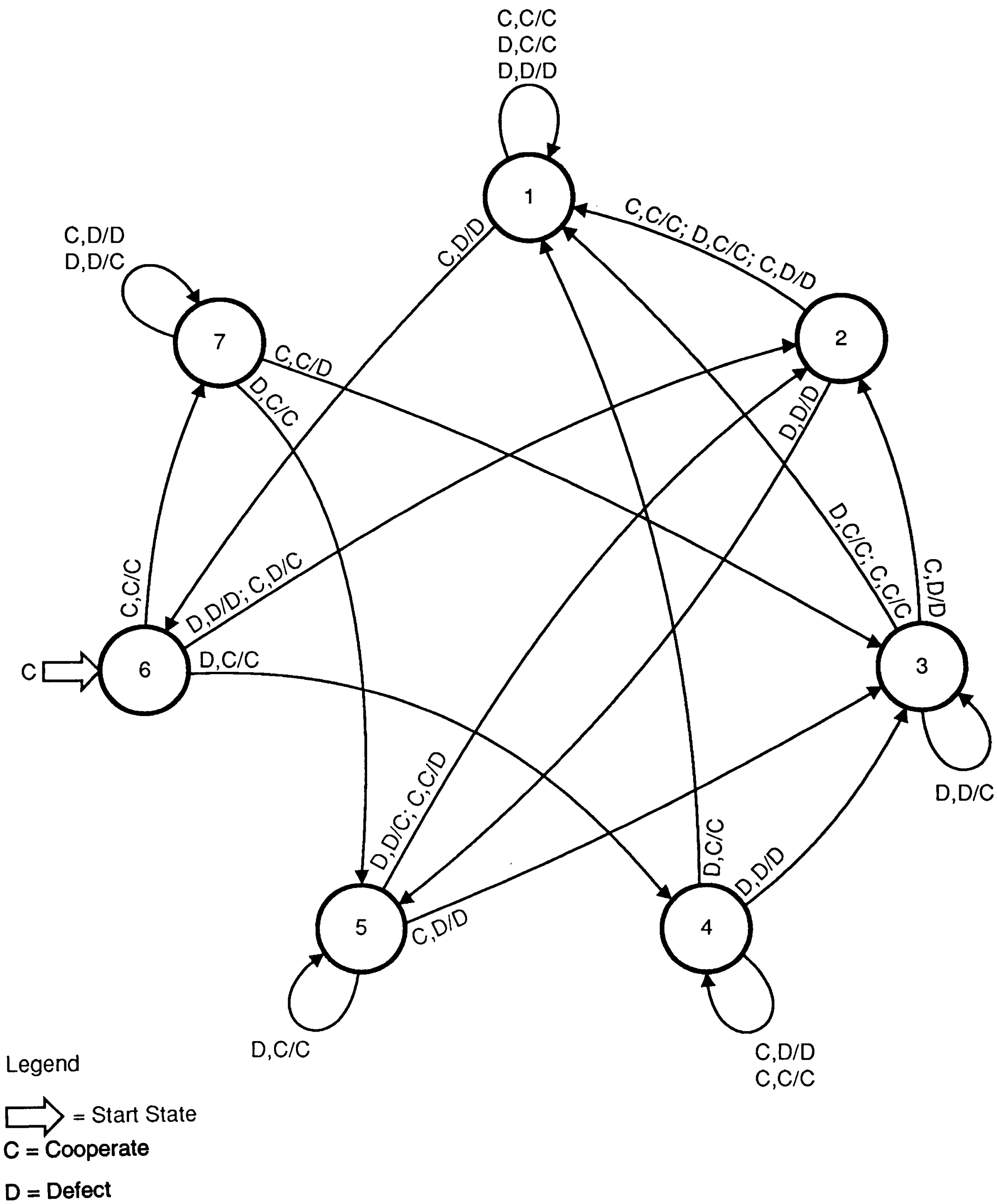

Figure 1. A Mealy machine that plays the iterated prisoner's dilemma. The inputs at each state are a pair of moves that corresponds to the player's and opponent's previous moves. Each input pair has a corresponding output move (cooperate or defect) and a next-state transition. The figure is taken from [27].

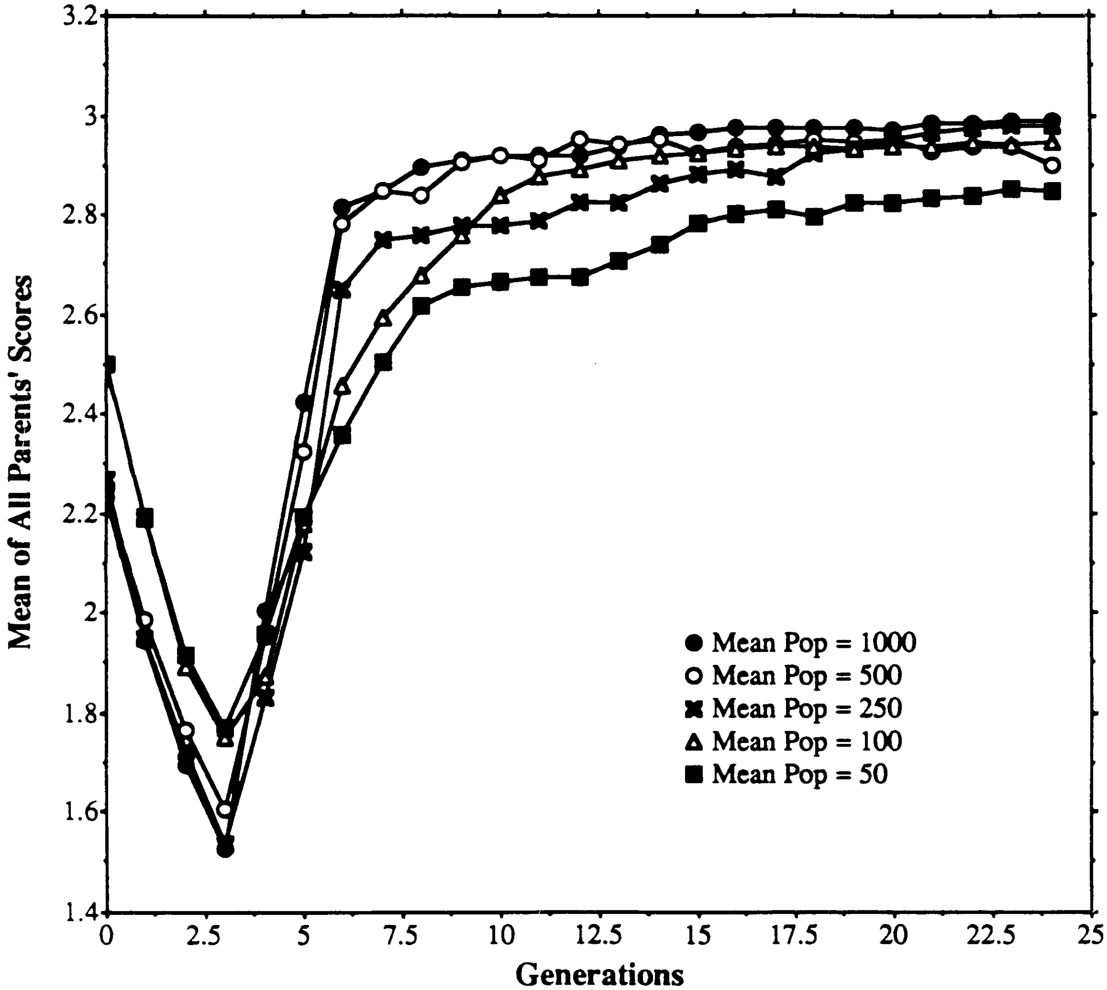

Figure 2. The mean of all parents' scores at each generation using populations of size 50 to 1000. The general trends are similar regardless of population size. The initial tendency is to converge toward mutual defection, however mutual cooperation arises before the 10th generation and appears fairly stable. The figure is taken from [19].

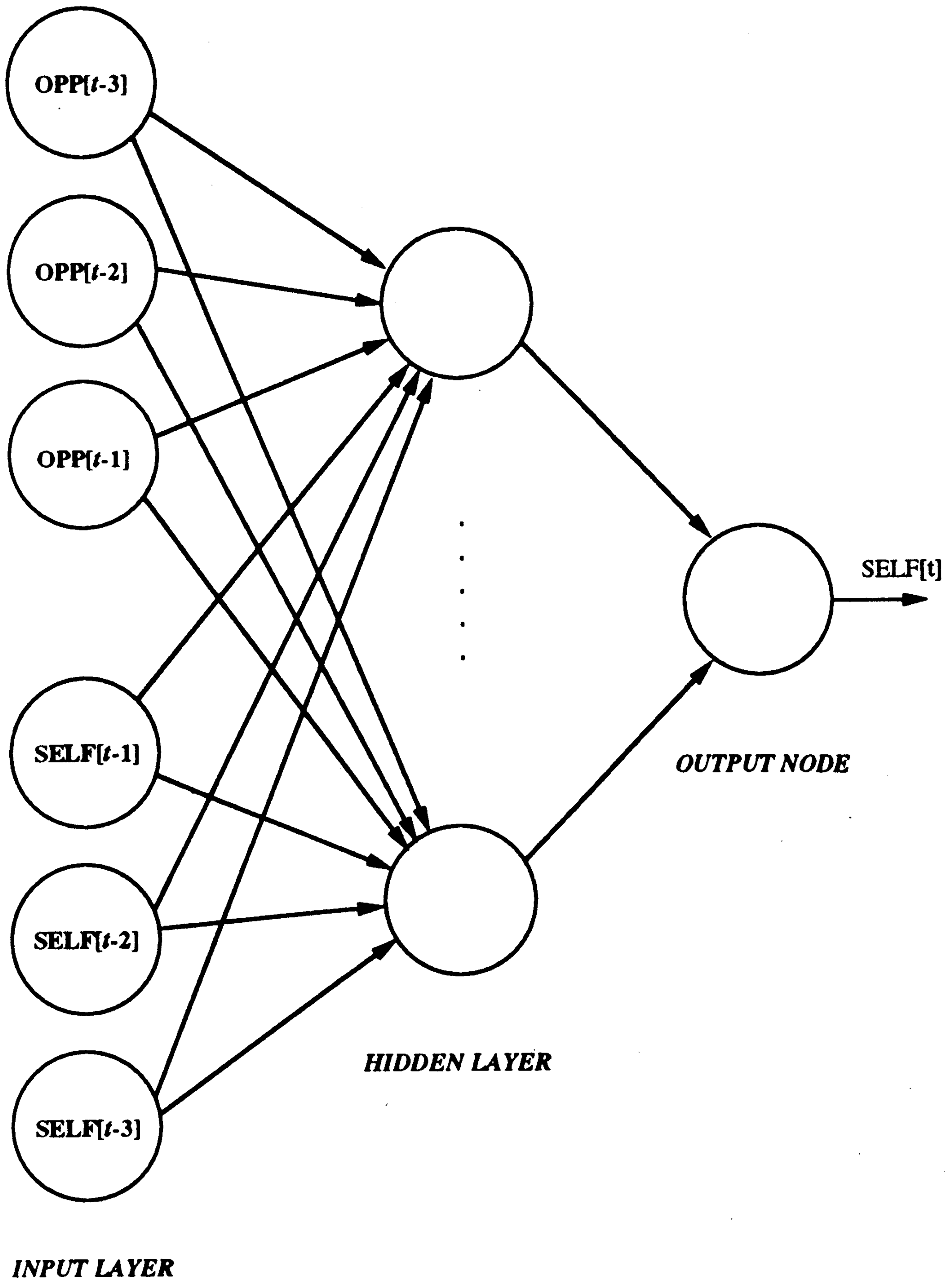

Figure 3. The neural network used in [26] for playing a continuous version of the iterated prisoner's dilemma. The inputs to the network comprise the most recent three moves from each player. The output is a value between –1 and 1, where –1 corresponds to complete defection and +1 corresponds to complete cooperation.

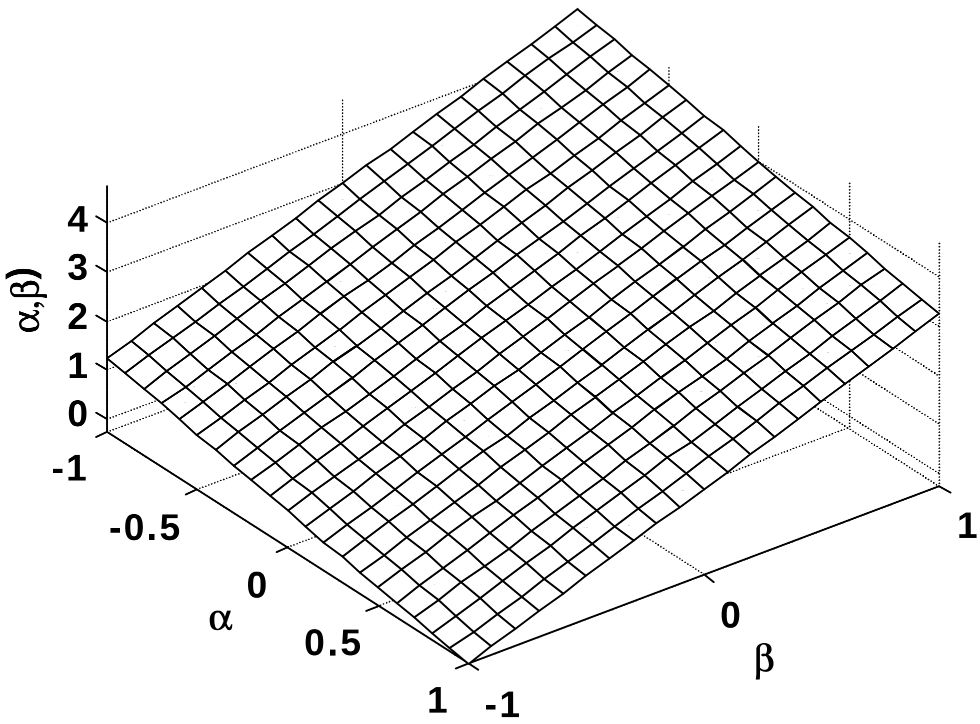

Figure 4. The planar approximation to Axelrod's payoff function [12] that was used in [26].

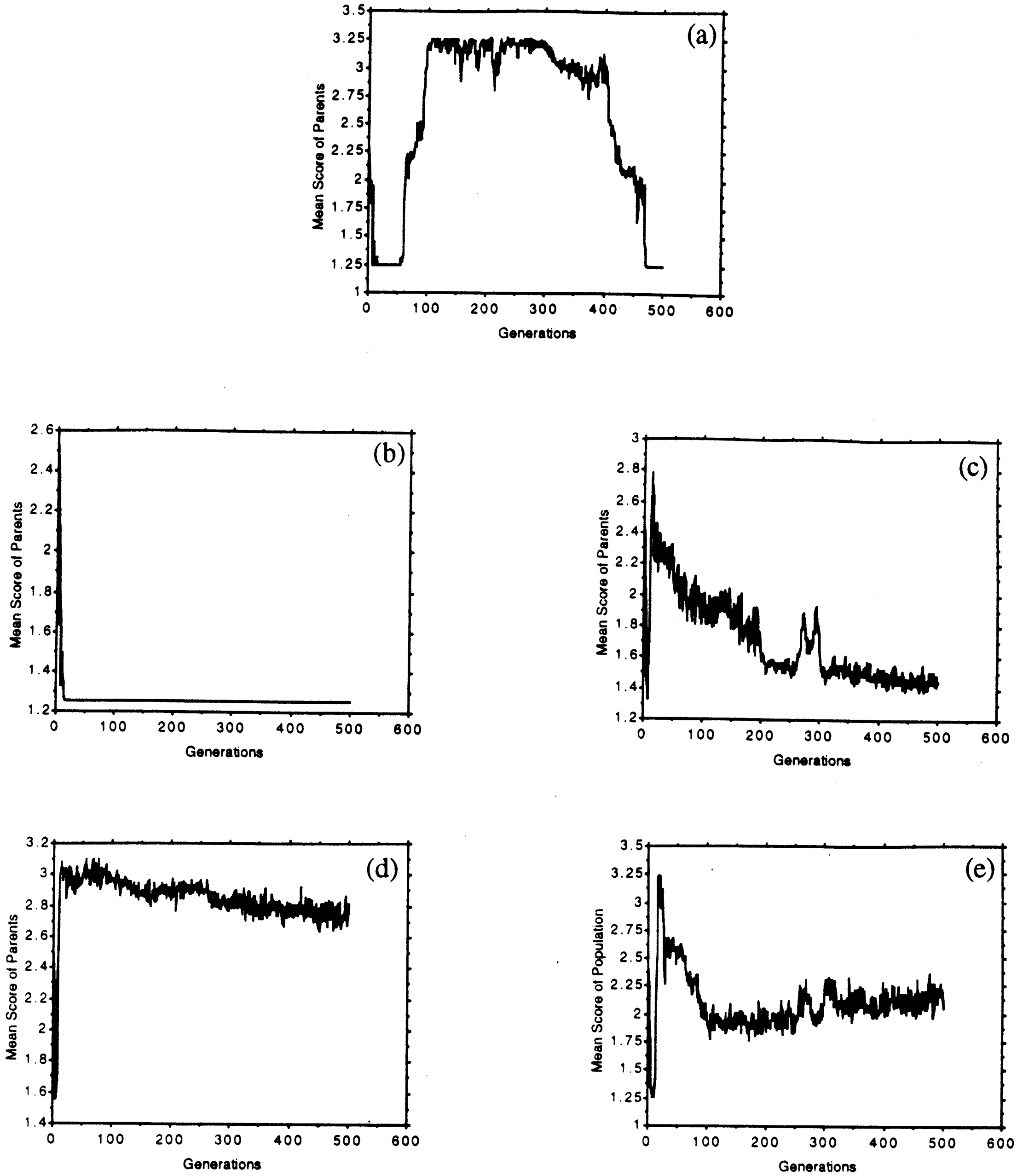

Figure 5. Examples of various behaviors generated based on the conditions of the experiments. (a) Oscillatory behavior (10 parents, trial #4 with 6-2-1 networks), (b) complete defection (30 parents, trial #10 with 6-20-1 networks), (c) decreasing payoffs leading to further defection (30 parents, trial #9 with 6-2-1 networks), (d) general cooperation with decreasing payoffs (50 parents, trial #4 with 6-20-1 networks), (e) an increasing trend in mean payoff (30 parents, trial #2 with 6-2-1 networks). The figure is taken from [26].

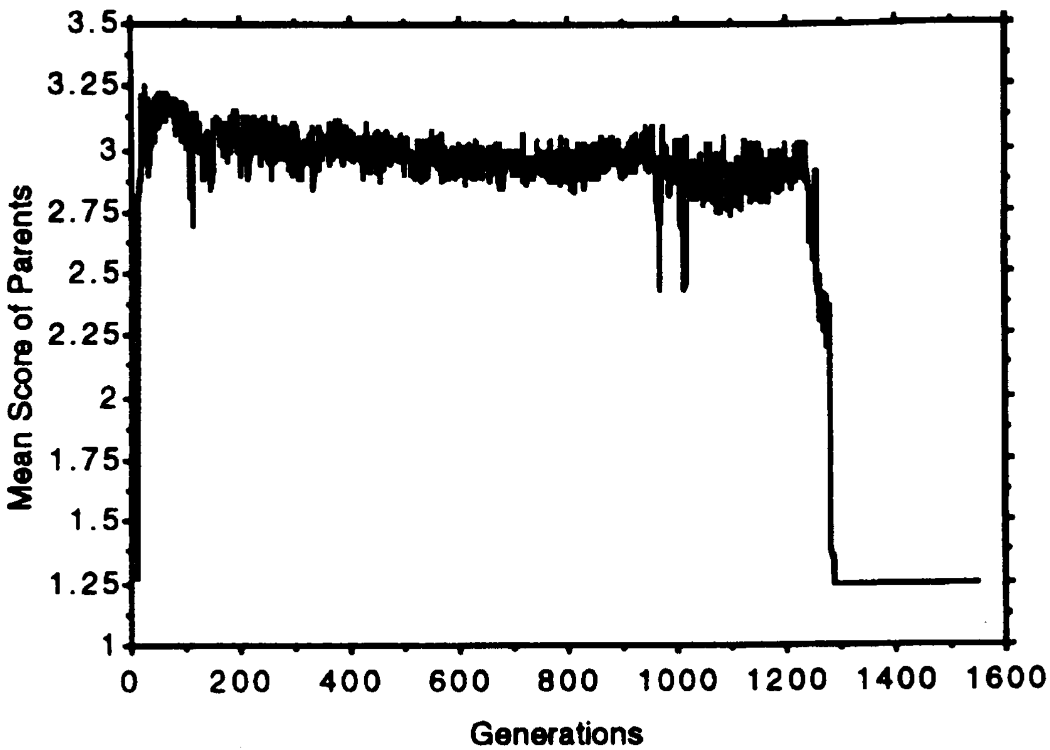

Figure 6. The result of trial #10, with 20 parents using 6-20-1 networks iterated over 1500 generations. The population's behavior changes rapidly toward complete defection after the 1200th generation (from [26]).

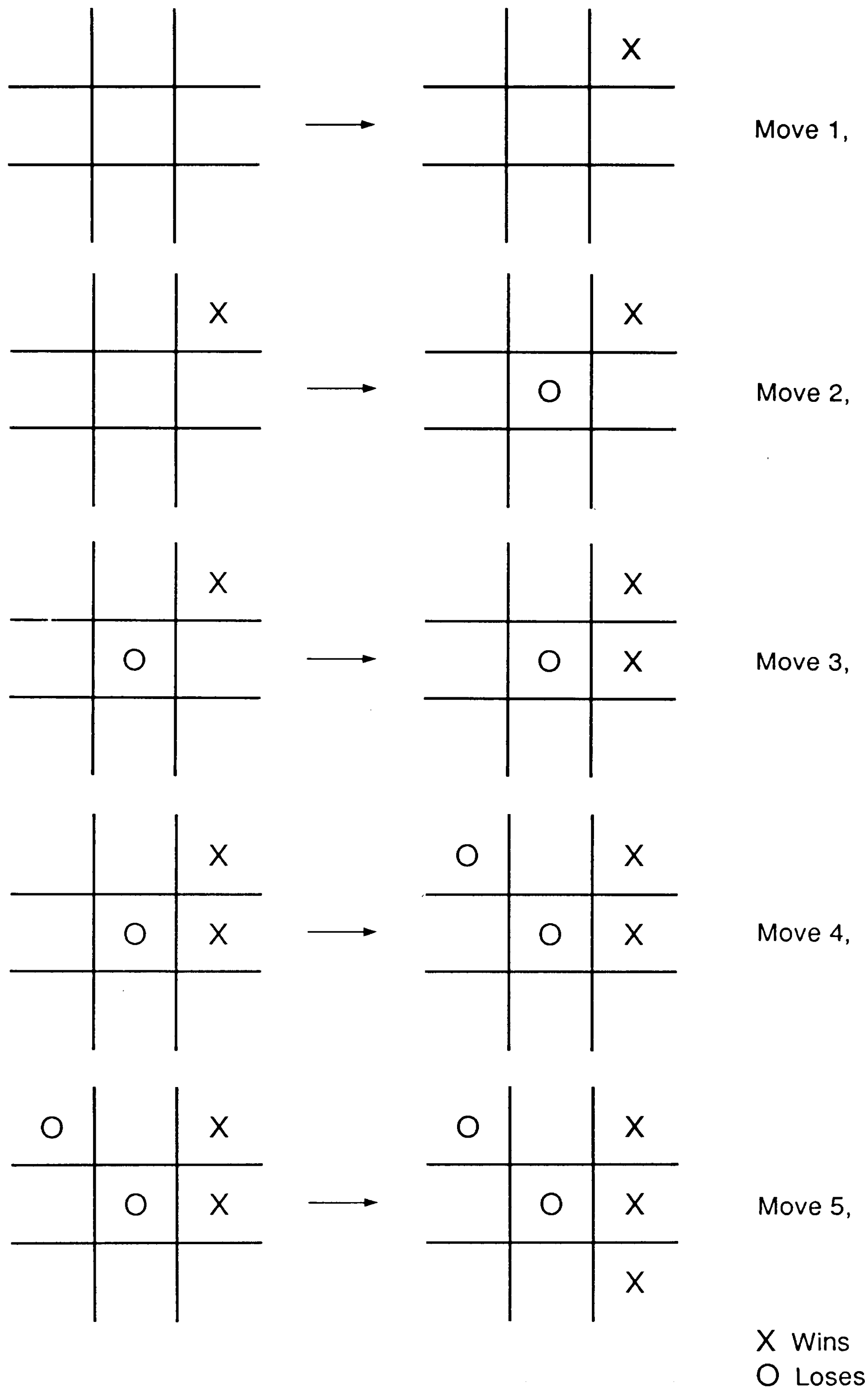

Figure 7. A possible game of tic-tac-toe. Players alternate moves in a 3×3 grid. The first player to place three markers in a row wins (from [27]).

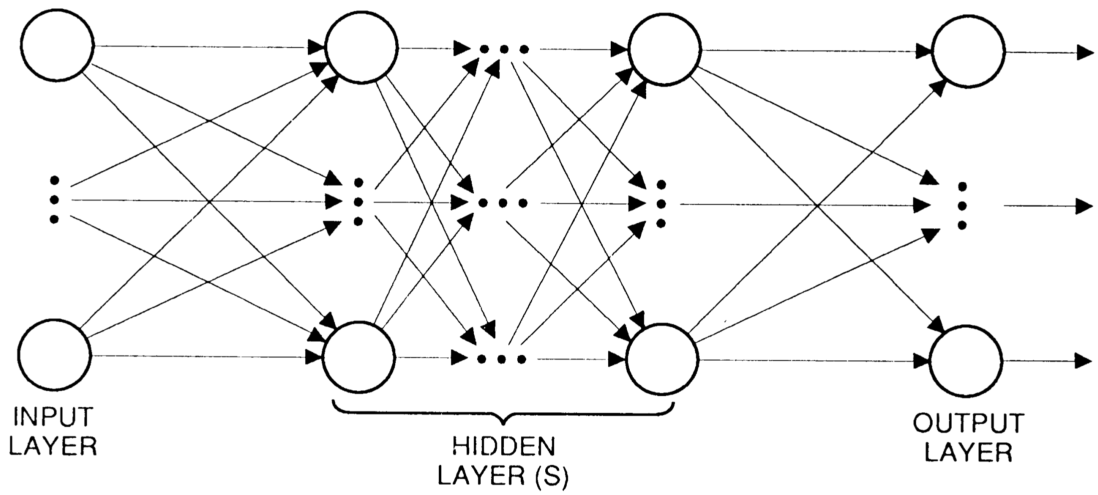

Figure 8. A multilayer feedforward perceptron. The neural network used in the tic-tac-toe experiments comprised a single hidden layer of up to 10 nodes. There were nine inputs and outputs which corresponding to the positions of the board (from [27]).

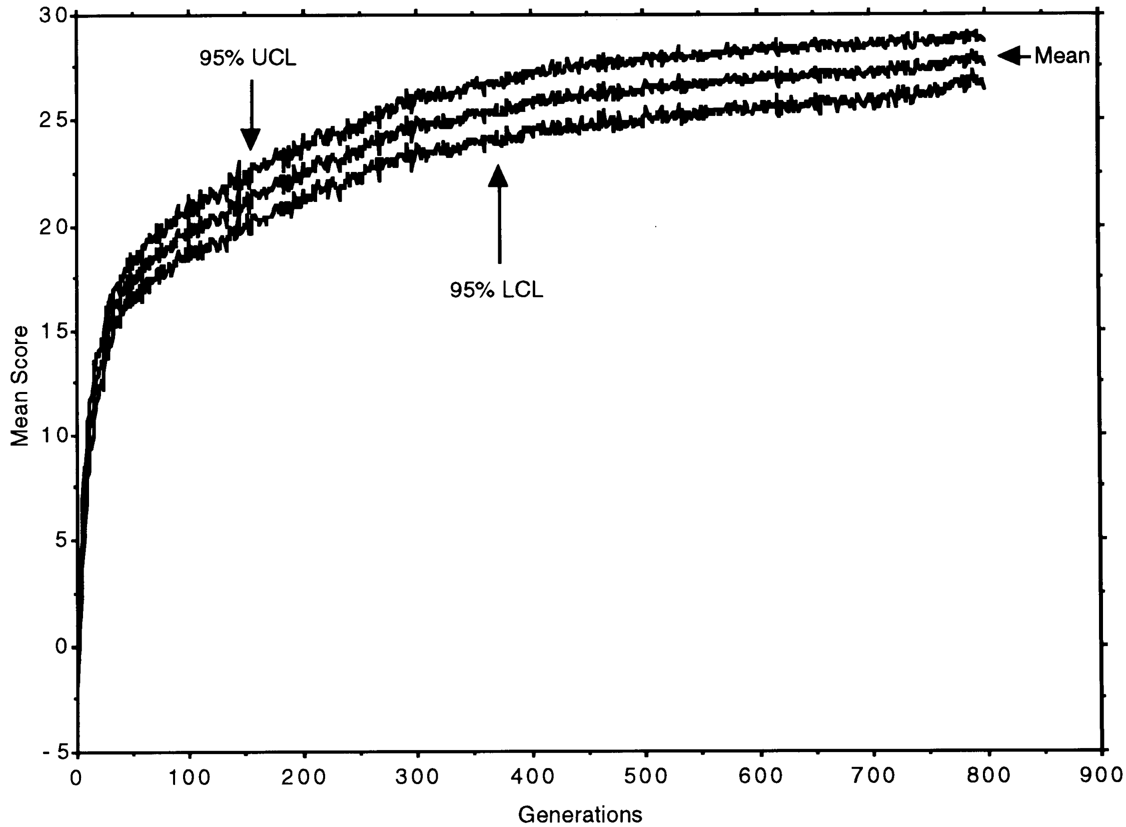

Figure 9.The mean score of the best neural network in the population as a function of the number of generations averaged across 30 trials using payoffs of +1, –1, and 0 for winning, losing, and playing to a draw, respectively. The 95 percent upper and lower confidence limits (UCL and LCL) on the mean were calculated using a *t*-distribution (from [27]).

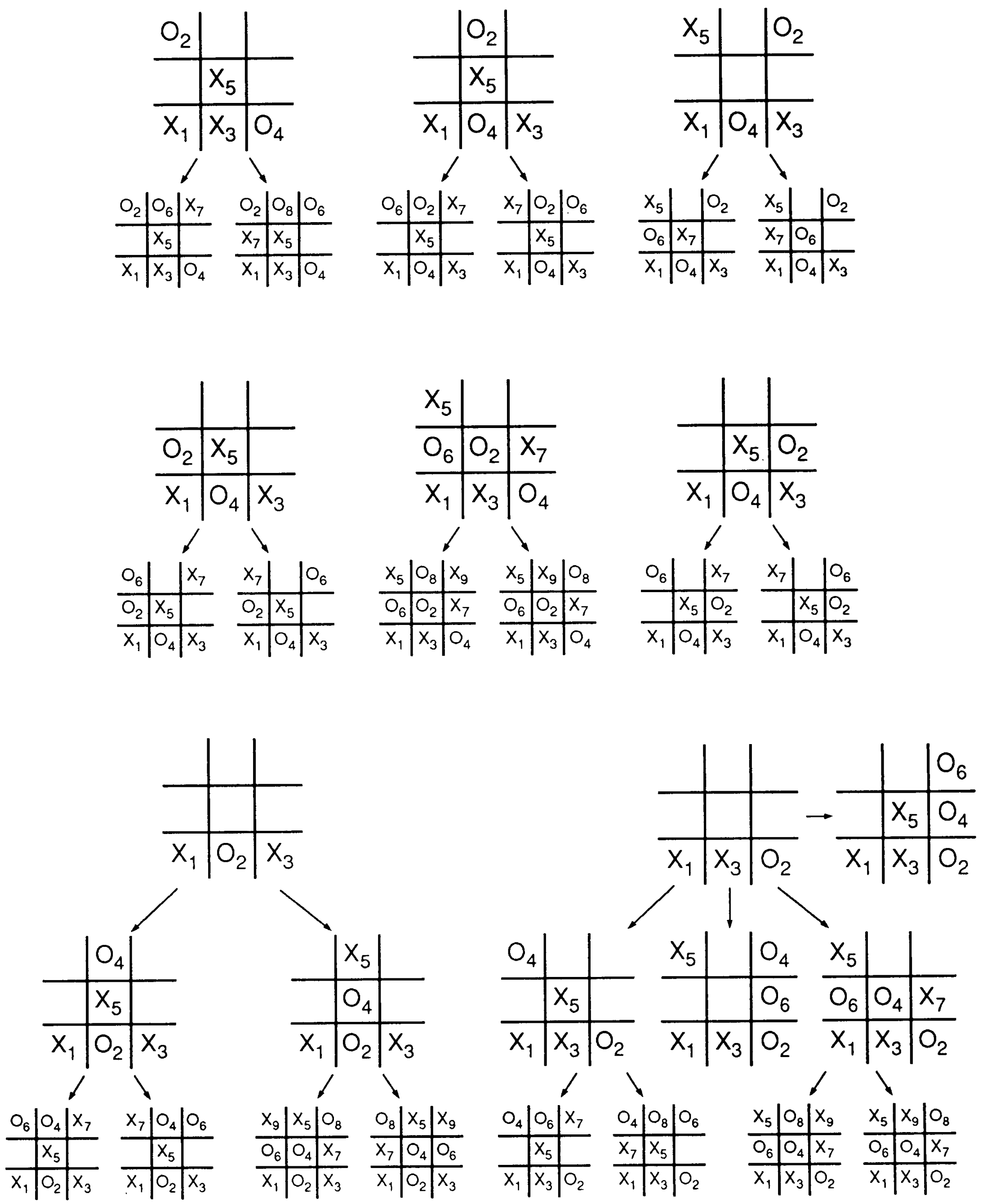

Figure 10. The tree of possible games when playing the best-evolved network from trial 2 with payoffs of +1, –1, 0 for winning, losing, and playing to a draw, against the rule base without the 10 percent chance for making purely random moves. Each of the eight main boards corresponds to the eight possible next moves after the neural network moves first (in the lower-left hand corner). Branches indicate more than one possible move for the rule base, in which case all possibilities are depicted. The subscripts indicate the order of play (from [27]).

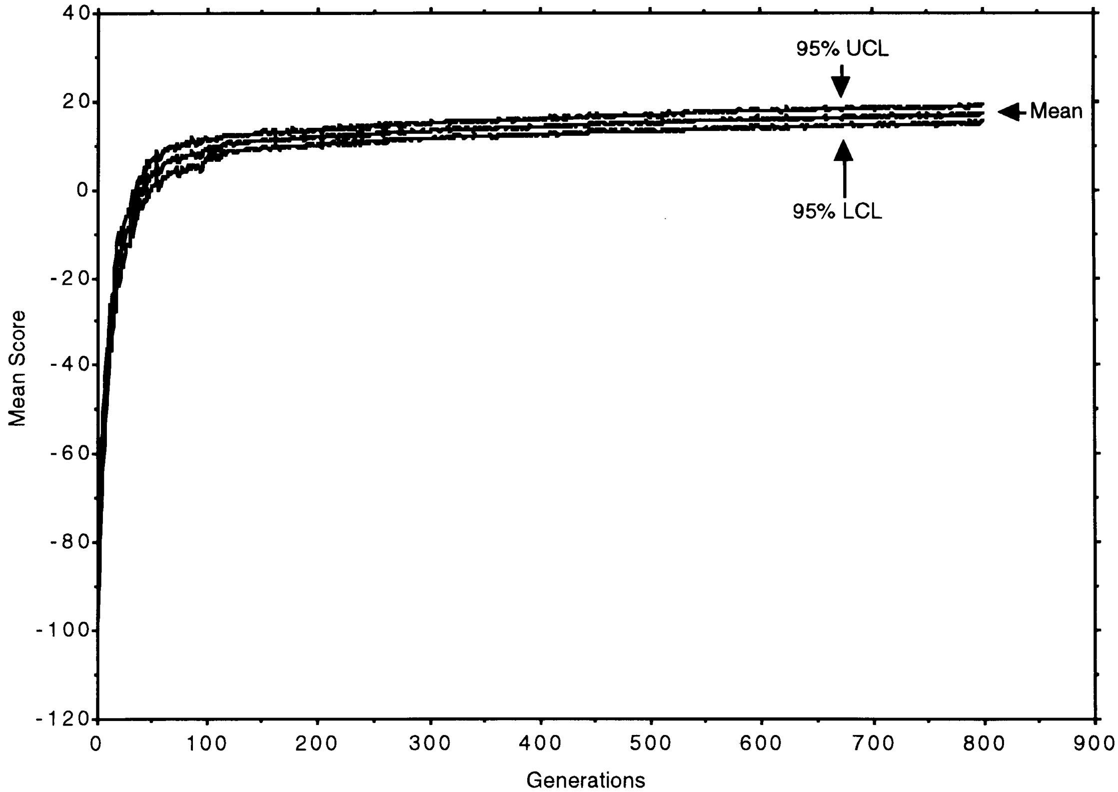

Figure 11. The mean score of the best neural network in the population as a function of the number of generations averaged across 30 trials using payoffs of +1, –10, and 0 for winning, losing, and playing to a draw, respectively. The 95 percent upper and lower confidence limits (UCL and LCL) on the mean were calculated using a *t*-distribution (from [27]).

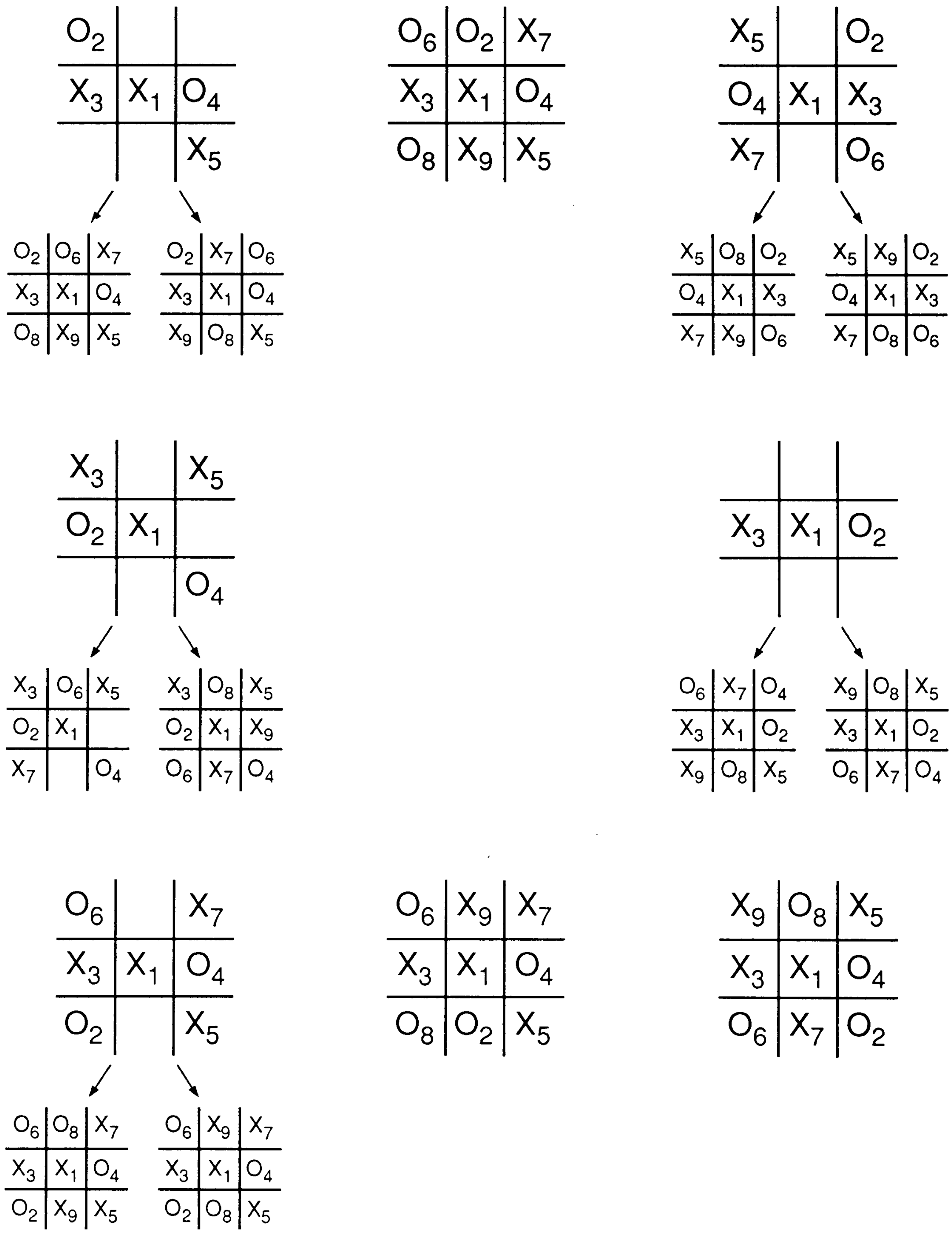

Figure 12. The tree of possible games when playing the best-evolved network from trial 1 with payoffs of +1, –10, 0 for winning, losing, and playing to a draw, against the rule base without the 10 percent chance for making purely random moves. The format follows Figure 10 (from [27]). The best-evolved network never forces a win, but also never loses. This reflects the increased penalty for losing.

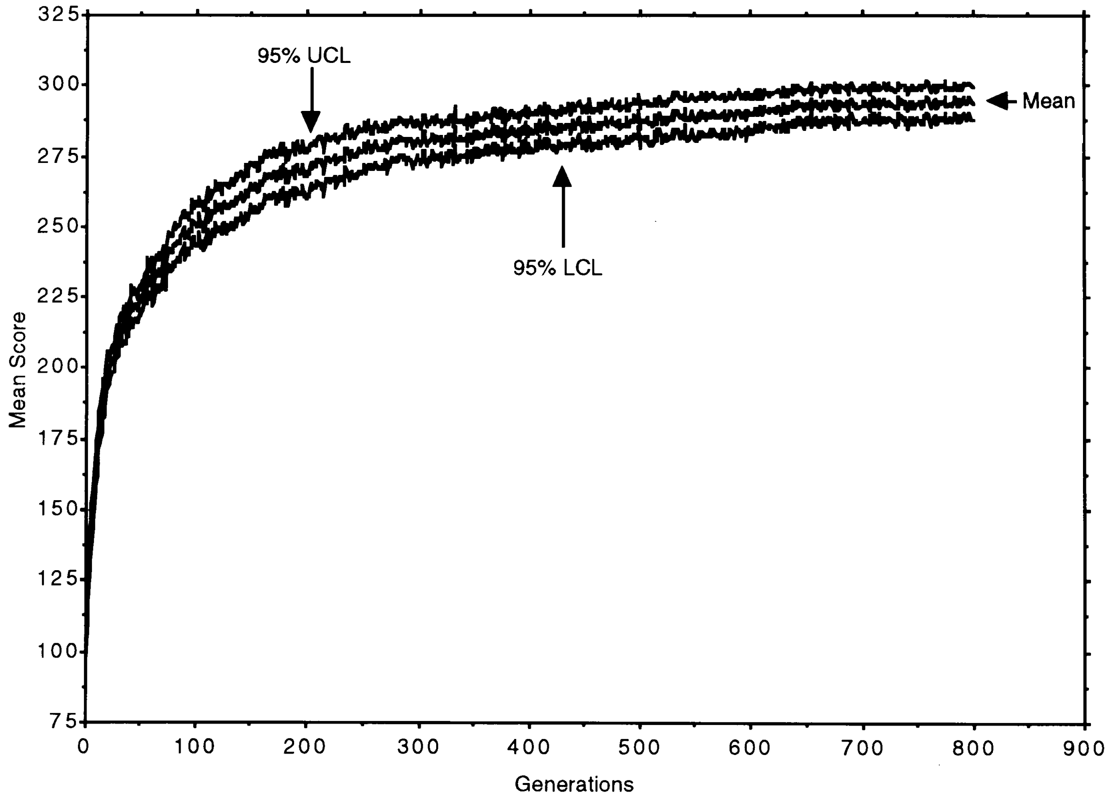

Figure 13. The mean score of the best neural network in the population as a function of the number of generations averaged across 30 trials using payoffs of +10, –1, and 0 for winning, losing, and playing to a draw, respectively. The 95 percent upper and lower confidence limits (UCL and LCL) on the mean were calculated using a t-distribution (from [27]).

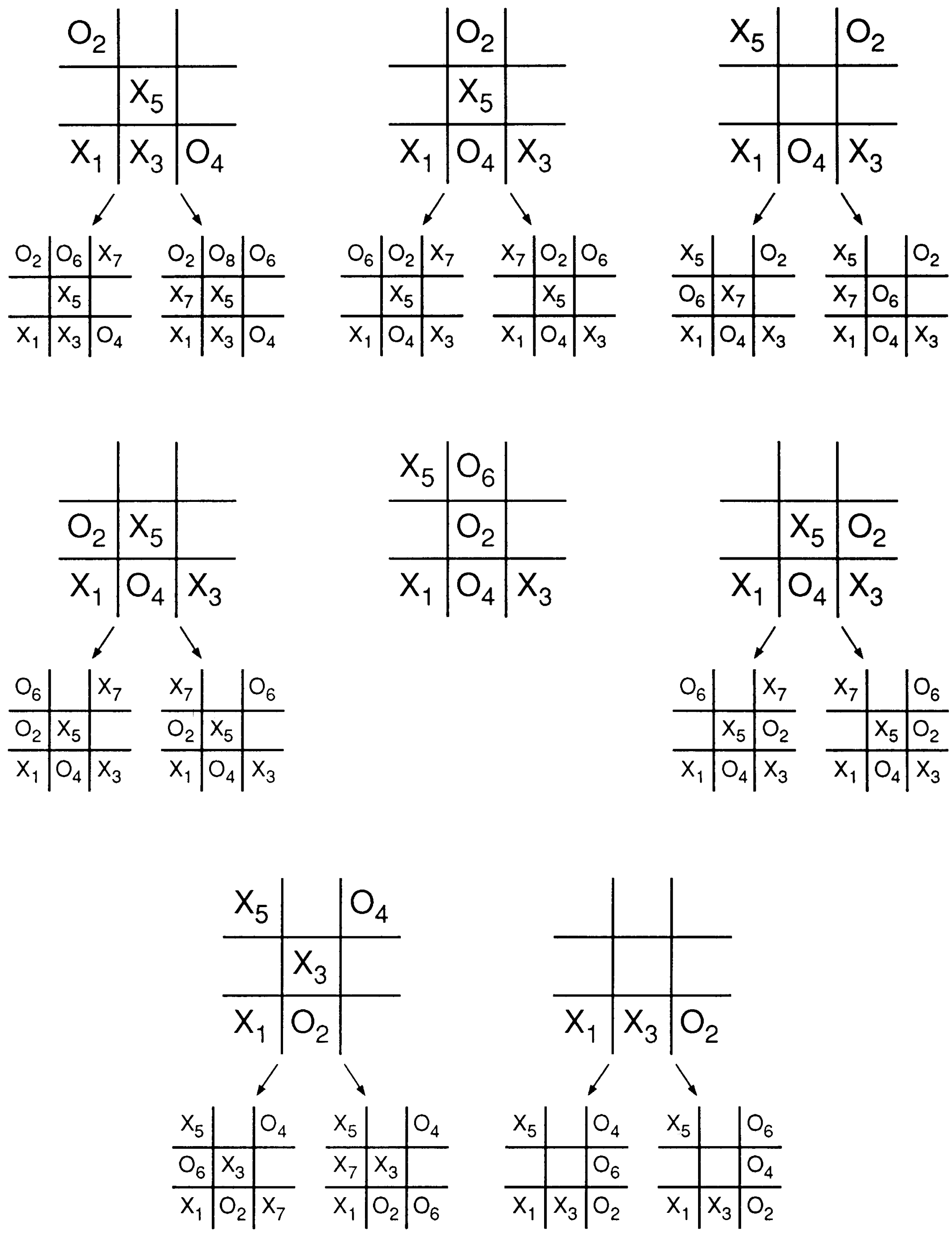

Figure 14. The tree of possible games when playing the best-evolved network from trial 1 with payoffs of +10, –1, 0 for winning, losing, and playing to a draw, against the rule base without the 10 percent chance for making purely random moves. The game tree has many of the same characteristics observed in Figure 10, indicating little difference in resulting behavior when increasing the reward for winning as opposed to increasing the penalty for losing.

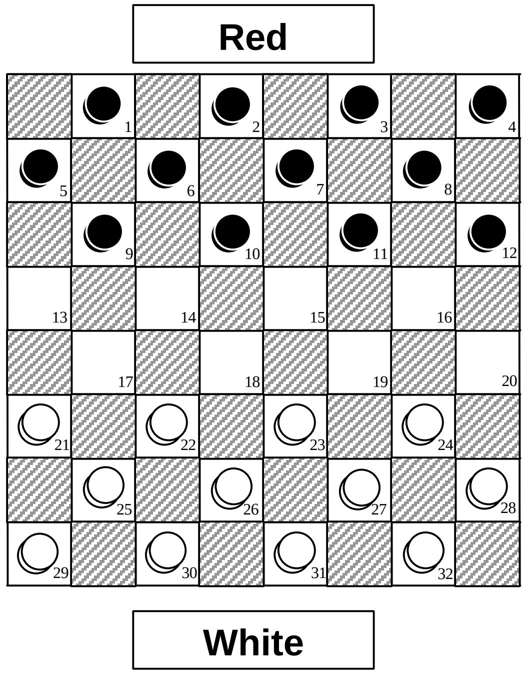

Figure 15. The opening board in a checkers game. Red moves first.

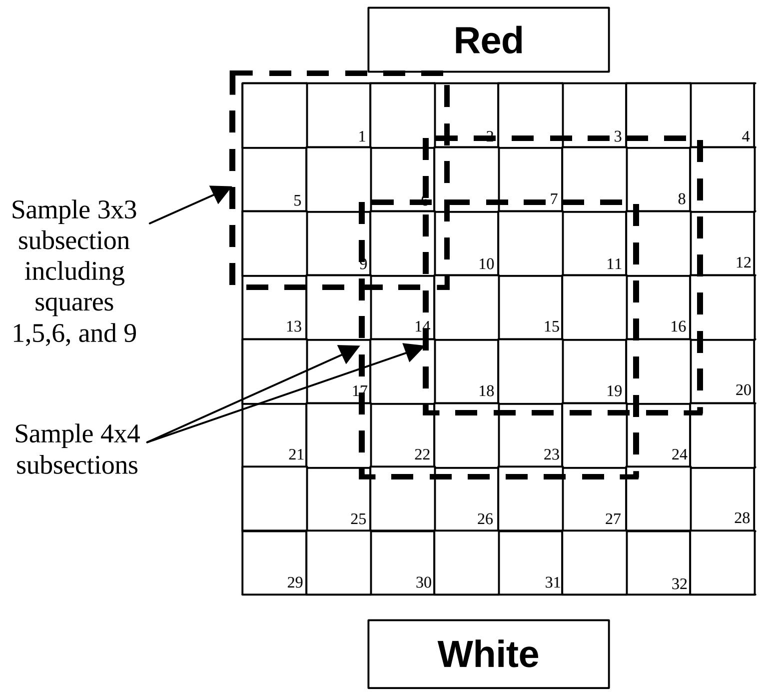

Figure 16. The first hidden layer of the neural networks assigned one node to each possible 3×3, 4×4, 5×5, 6×6, 7×7, and 8×8 subset of the entire board. In the last case, this corresponded to the entire board. In this manner, the neural network was able to invent features based on the spatial characteristic of checkers on the board. Subsequent processing in the second and third hidden layer then operated on the features that were evolved in the first layer.

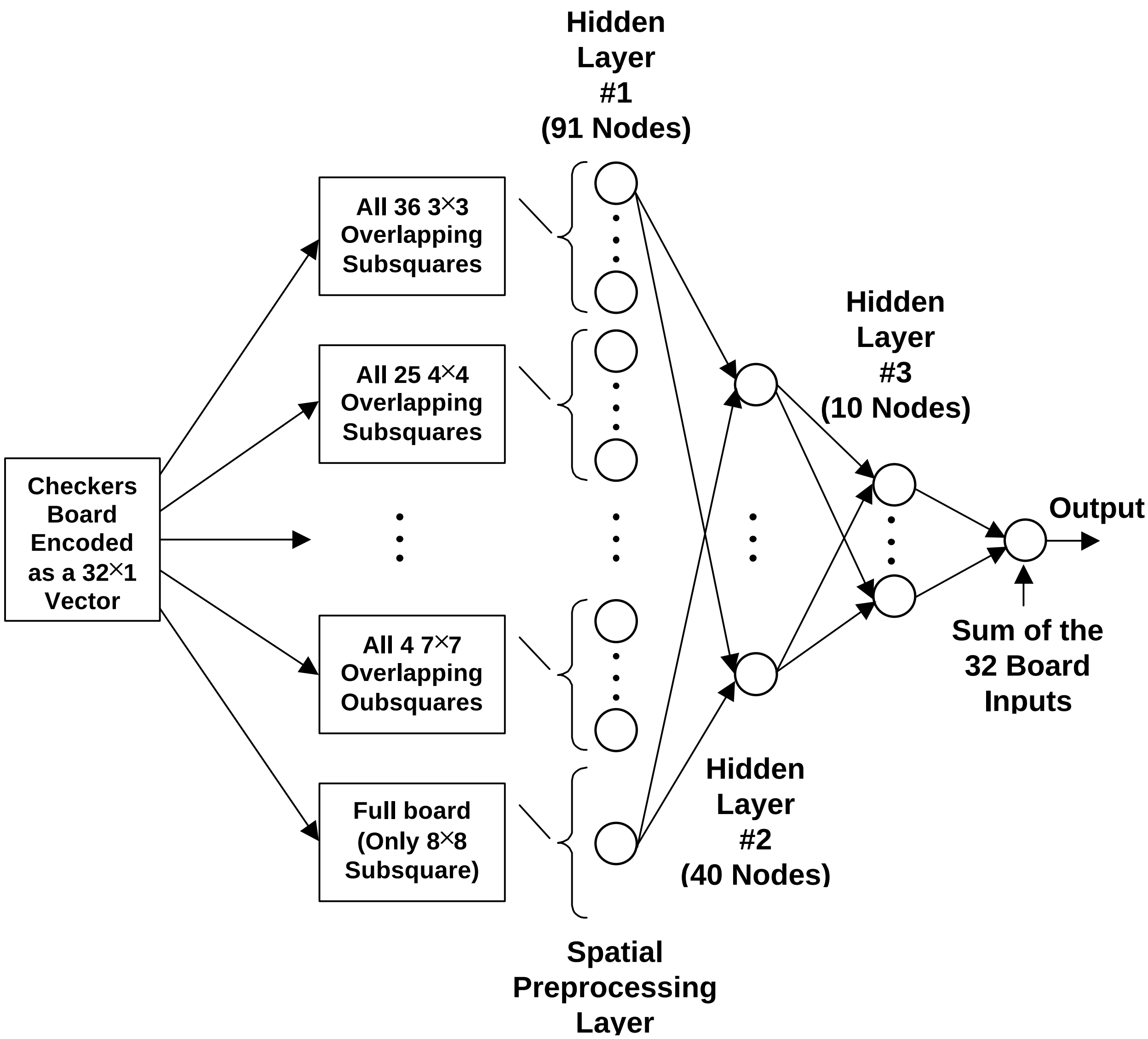

Figure 17. The complete "spatial" neural network used to represent strategies. The network served as the board evaluation function. Given any board pattern as a vector of 32 inputs, the first hidden layer (spatial preprocessing) assigned a node to each possible square subset of the board. The outputs from these nodes were then passed through two additional hidden layers of 40 and 10 nodes, respectively. The final output node was scaled between [−1,1] and included the sum of the input vector as an additional input. All of the weights and bias terms of the network were evolved, as well as the value used to describe kings.

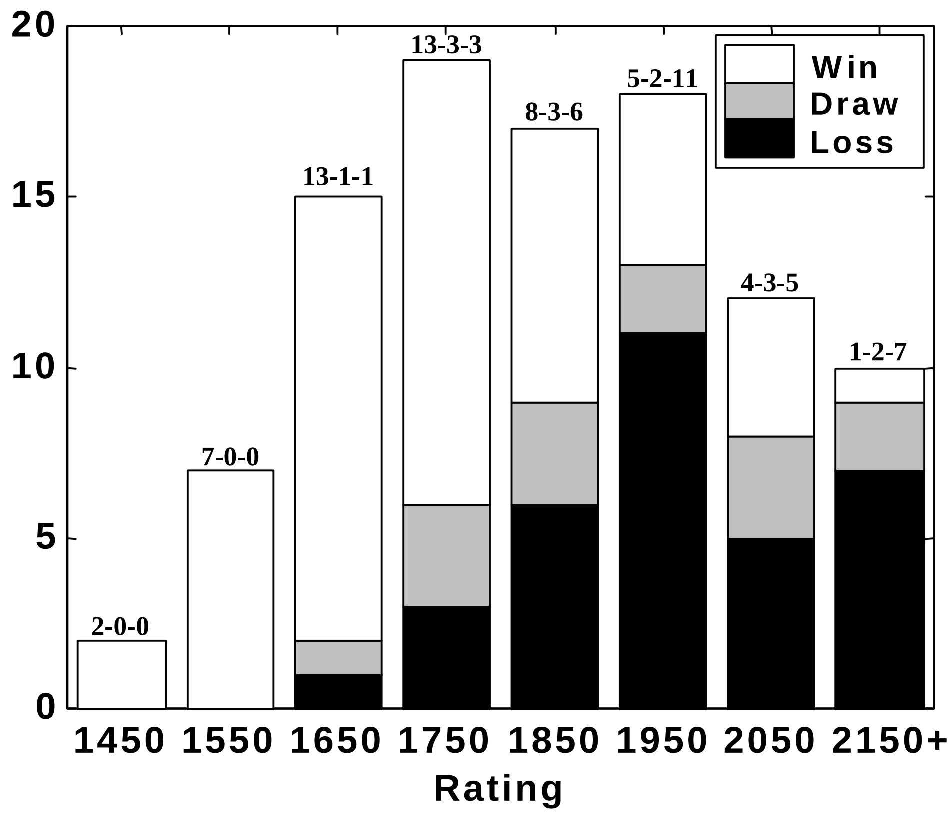

Figure 18. The performance of the best evolved neural network after 230 generations, played over 100 games against human opponents on an internet checkers site. The histogram indicates the rating of the opponent and the associated performance against opponents with that rating. Ratings are binned in intervals of 100 units (i.e., 1650 corresponds to opponents who were rated between 1600 and 1700). The numbers above each bar indicate the number of wins, draws, and losses, respectively. Note that the evolved network generally defeated opponents who were rated less than 1800, and had a majority of wins against those who were rated between 1800-1900.

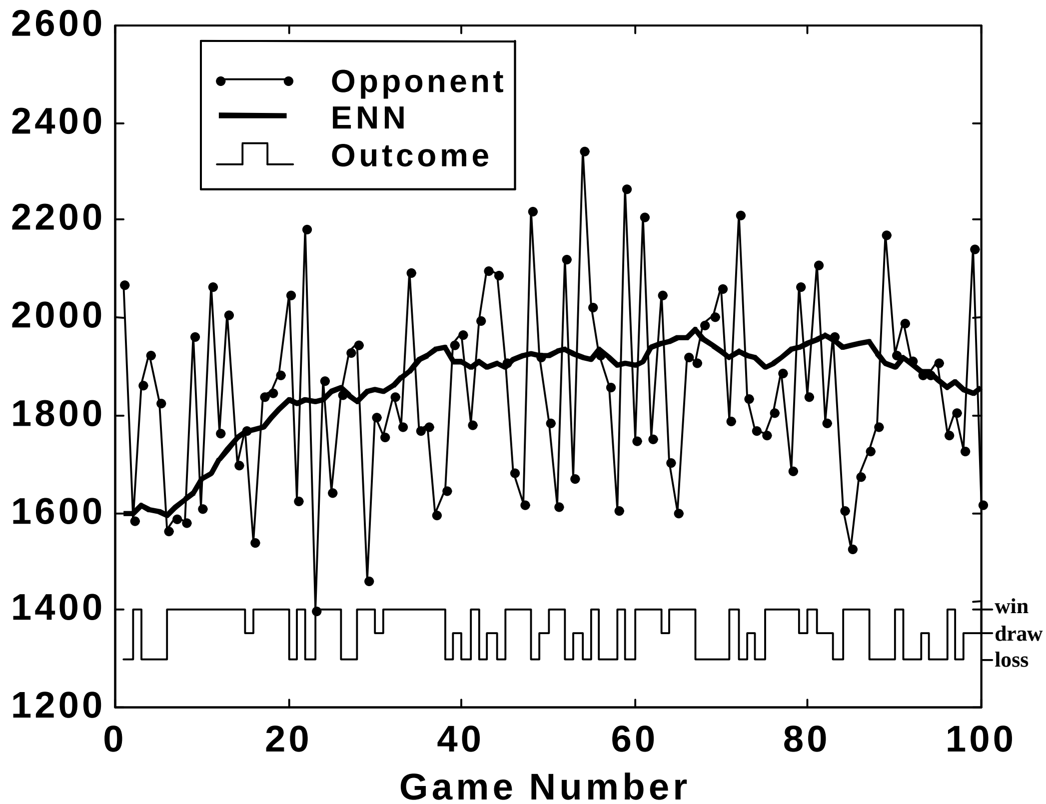

Figure 19. The sequential rating of the best evolved neural network (NN) over the 100 games played against human opponents. The graph indicates both the neural network's rating and the corresponding rating of the opponent on each game, along with the result (win, draw, or loss). The final rating score depends on the order in which opponents are played. Since no learning was performed by the neural network during these 100 games the order of play is irrelevant, thus the true rating for the network can be estimated by taking random samples of orderings and calculating the mean and standard deviation of the final rating (see Figure 20).

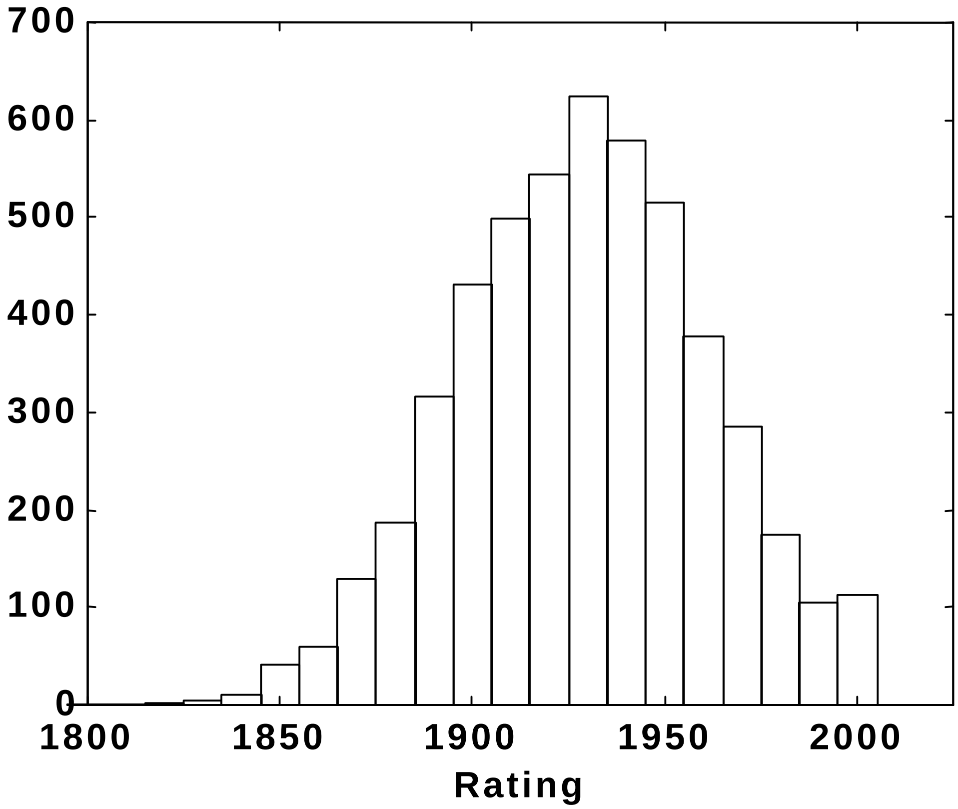

Figure 20. The rating of the best evolved network after 230 generations computed over 5000 random permutations of the 100 games played against human opponents on the internet checkers site (www.zone.com). The mean rating was 1929.0 with a standard deviation of 32.75. This places it as an above-average Class A player.

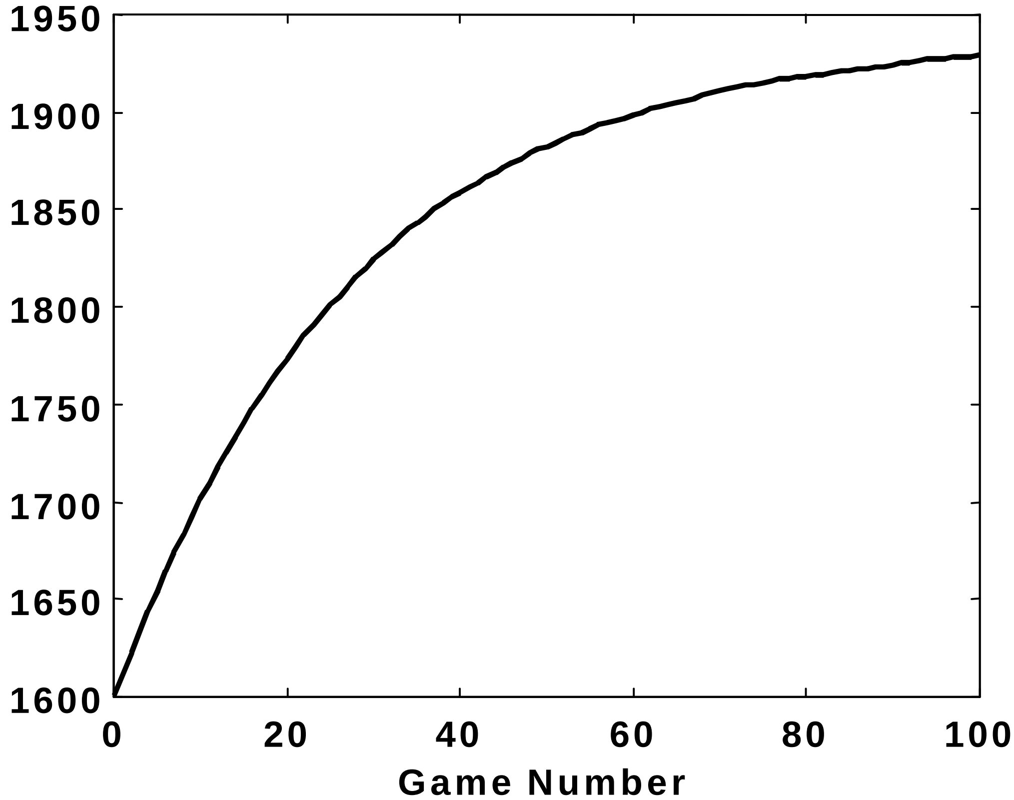

Figure 21. The mean sequential rating of the best evolved neural network (NN) over 5000 random permutations of the 100 games played against human opponents. The mean rating starts at 1600 (the standard starting rating at the website) and steadily climbs to a rating of above 1850 by game 40**.** As the number of games reaches 100, the mean rating curve begins to saturate and reaches a value of 1930.0.

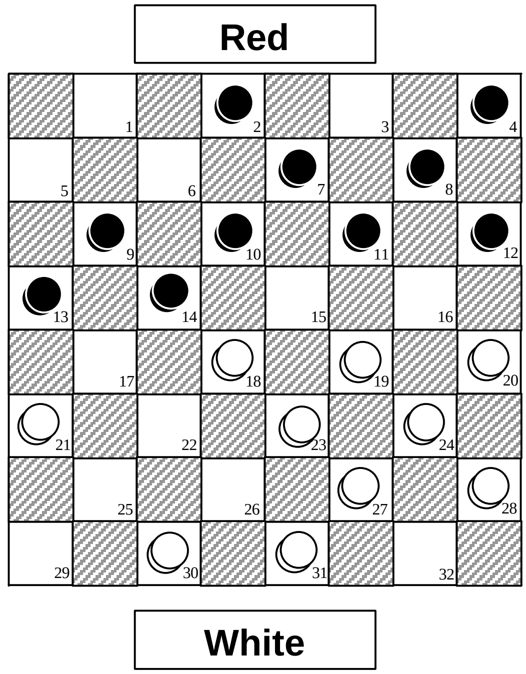

Figure A1. The game reaches this position on move number 11 as White (Human) walks into a trap set by Red (NN) on move number 8 (Table A1). Red (NN) picks 11-16 and the game proceeds `11.W:20-11(f); 12.R:8-15-22`. White loses a piece and never recovers.

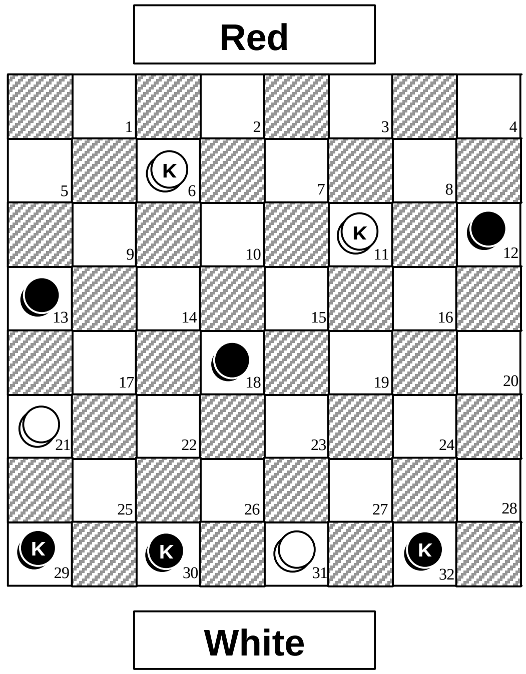

Figure A2. An endgame position at move 32 after Red (NN) chooses to move `R:14-18`. White (Human) is behind by a piece and a king. White's pieces on 13 and 31 are pinned and he foresees a three-king (Red) vs. two-king (White) endgame. White resigns.

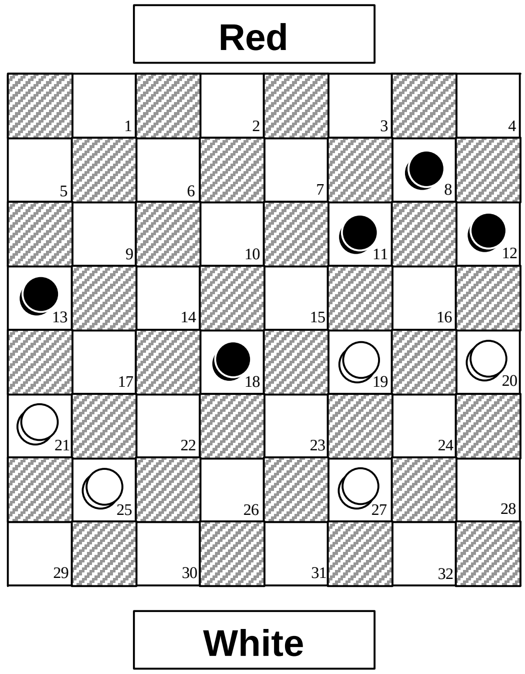

Figure A3. The board position after move 19 (`19.R:14-18 19.W:29-25`) on a game with the best neural network (W) playing a human expert (rated 2025 and playing R). Every legal move for Red will result in the loss of one or more pieces. Red chooses 11-15 and is forced to give up a piece. Red never recovered from this deficit.

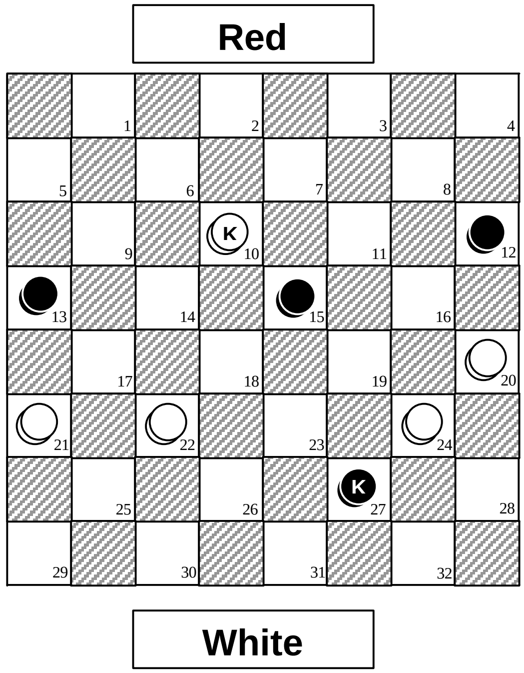

Figure A4. The final board position, with Red (Human) to move. Red's mobility is very restricted with his pieces on 12 and 13 being pinned. Further, Red is down by a piece and is about to lose the piece on 15 as he cannot protect it. Red (Human) forfeits the game.