Monte Carlo Reinforcement Machine Learning Algorithm

The Monte Carlo algorithm runs simulations on a task to train its policy. Once this policy is well trained the algorithm can complete the task.

Training

While training, the algorithm completes several simulations on its environment. For each state, the policy has a value for the expected reward given for each move. During each simulation, the algorithm is run on its environment. After a simulation, the algorithm updates its policy based on the score it received during the simulation. This changes the policy for each of the moves taken during the simulation.

Policy Update

For each training simulation, the policy is updated. This process takes the penalty or reward from the last state and updates the last move. The policy for the move from the second to last state to the last state is updated by subtracting the reward or penalty from the policy for that move then multiplying that value by a weight and adding it to the current policy. The weight limits the amount that one simulation can affect the policy. After this, the expected reward is updated to reflect the moves taken past the previous state then the policy is updated for the move from the previous state to the next state. This continues until all moves taken have been updated based on the expected reward from that point.

Moving

While training the algorithm may move in a random direction or based on the policy. Random movements can give the algorithm a wider variety of experiences during the simulations. While testing the algorithm, movement should be based on policy. For each state the algorithm is in it should look for the move with the highest policy value and select it.

Testing

While testing, the algorithm uses its trained policy to complete the task. This involves selecting the move with the highest expected reward at each state. When the algorithm has been trained properly it will perform better than an untrained model showing that it learned from the simulations.