Reinforcement Learning Agent To Play Checkers

Hari Surayagari, Tom Peter
hari.surayagari@colostate.edu, tom.peter@colostate.edu
Colorado State University, Fort Collins, USA

Motivation

- Create a machine learning agent that learns to play checkers
- The game of checkers seems very simple at first glance but in fact, it does indeed have few rules However, little is known about winning strategies and complications arise quickly, as there are literally more possible board states and moves than suspected atoms in the universe
- Checkers is particularly well-suited to reinforcement learning because a board state is easily observed and an action is simply the next move

Research Problem

Problem Statement:

Developing a Reinforcement Learning agent to play Checkers

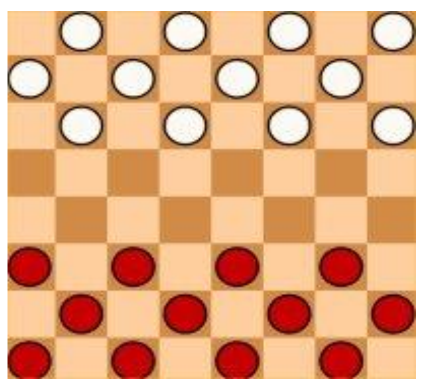
Hypothesis:

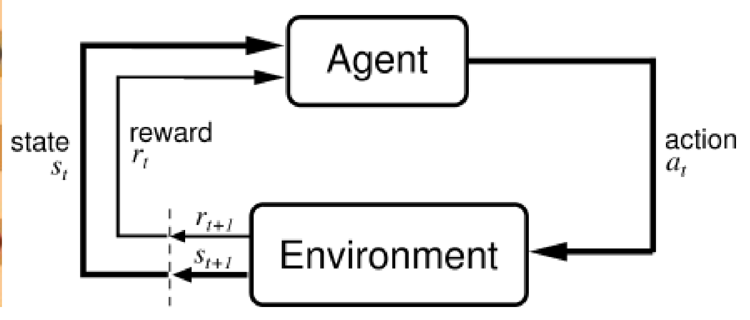
Develop a reinforcement learning agent that can learn how to play checkers by playing the game repeatedly by following different states and defining possible rewards for each state

Research outline:

- Define the various elements of Reinforcement learning such as state, action, reward, policy with respect to checkers game
- Program the main classes involved in the project i.e. Reinforcement Learning agent, random agent and the checkers game
- Study the impact of using reinforcement learning agent on checkers game

Background Work





- Develop a program for checker i.e. the board and the legality of the moves. The checkers board will be the observable state. The action will be the next move and the reward is 1 for a win, 0 for a draw and -1 for a loss

Methods

Standard Reinforcement Learning techniques have been applied in this project.

Temporal Difference learning (TD(0)) is used in this project. TD(0) requires a value of the state which in this case is the value of the board at a given point.

V(s) = (7 * x1) + (5 * x2) + (3 * x3)

Where, V(s) = Value of state

X1 = Number of pieces in the opposing third.

X2 = Number of pieces in the middle third

X3 = Number of pieces in the bottom third

This forces the pieces to move forward but if there is an opportunity for a piece to jump then it has no choice but to do so.

Q learning:

It is an off-policy algorithm of TD(0). If we have enough evidence to converge, then the algorithm does so.

$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma * Q(s`, a`) – Q(s,a)]$

Where, α = Learning Rate, set at 0.2

γ = Discount Factor, set at 1

r = Reward

In this project the agent is always black and the opponent is always white. The V(s) function above calculates the values separately for each side.

Algorithm:

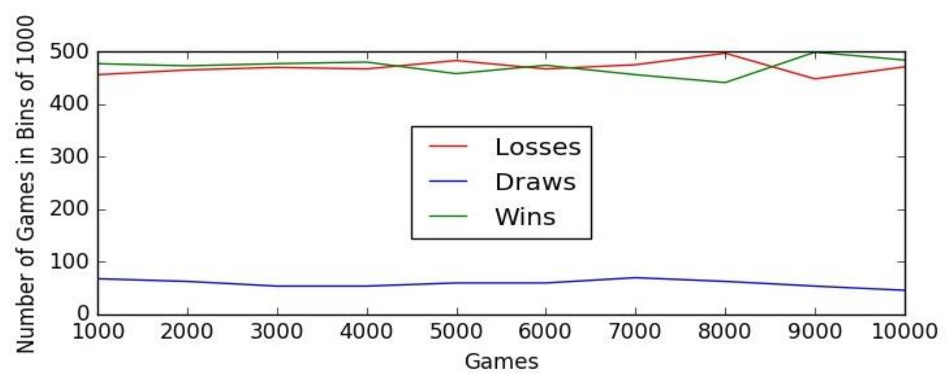
1. Check for legal moves
2. Check for moves in Q dictionary.
3. If move not found or its value is less than 0, run AI or select random move
4. Store the move and the board state prior to move in Q dictionary.
5. Update the Q dictionary
6. Check if game over, if not - repeat 1-5
7. If game over: reward = 1 if agent won, -1 for defeat and 0 for a draw.
8. Repeat until the max games is reached.

The game completion is decided by checking if all the pieces of any side are removed or if there are no legal moves for both sides.

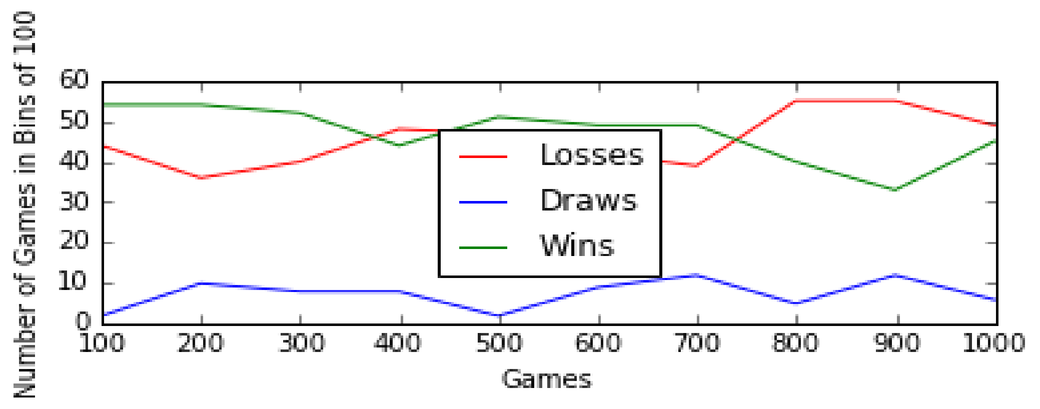
Results & Conclusion

Random Agent: Total games played : 10000

- Number of wins by black :4719
- Number of wins by white 4699
- Number of draws : 582 draws



- Total games played : 1000
- Number of wins by black :471
- Number of wins by white 455
- Number of draws : 74 draws



Conclusion : As per the results generated above, it is observed that the number of wins by black (learning agent) is slightly more than white which shows this checkers learning agent seems to have learned how to slightly tower over the given opponent. The involvement of AI to help the agent learn faster didn't particularly help as the computation time with AI was more than it was without it. Checkers has a search space of $5x10^{20}$, this number indicates the requirement of a large number of Q values for the algorithm to function properly. Given the time constraints and computation capabilities, generating enough Q values hasn't been feasible.

Future Work: The primary purpose of this project was to explore the numerous possibilities of reinforcement learning and understanding the complexities of checkers. Although the project didn't yield concrete proof of learning, we plan on maybe increasing the depth of search and hope to decrease the number of Q values required. This will hopefully indicate and clear learning by the agent.

References

- Ghori I, *"Reinforcement Learning in Board Games"*, Dept of Computer Science, University of Bristol, report CTR-04-004 2004
- Marco A.W, Jan Peter P, Henk Mannen, *"Learning to Play Board Games using Temporal Diference Methods",* Institute of Information and computing sciences, Utrecht University, Report UU-CS-2005-048
- Richard S Sutton and Andrew G Barto, *"Reinforcement Learning: An Introduction"* MIT Press 1998
- Samuel, Arthur L., *"Some Studies in Machine Learning Using the Game of Checkers"*, IBM Journal of Research Development, Vol.3. No. 3, 1959
- Schaeffer, Lake, Lu, Bryant, Treloar. Chinook.University of Alberta, Edmonton, Alberta, Canada http://www.cs.ualberta.ca/~chinook/, last visited 12/10/2015