Reinforcement Learning, Looking for New Backgammon Strategies

Fatema Alkhanaizi

A Dissertation presented for the degree of
Bachelor of Computer Science with placement

Supervised by Rob D. Powell
Department of Computer Science
University of Durham
United Kingdom

April 2019

Chapter 1

Background

The domain of complex board games such as Go, chess, checkers, Othello, and backgammon has been widely regarded as an ideal testing ground for exploring a variety of concepts and approaches in artificial intelligence and machine learning. TD-Gammon of Tesauro [Tes92; Tes02] had demonstrated the impressive ability of machine learning techniques to learn to play games. TD-Gammon used reinforcement learning techniques with a Neural Network (NN) that trains itself to be an evaluation function for the game of backgammon, by playing against itself and learning from the outcome [Tes02]. eXtreme Gammon [Gam17], Snowie [Egg17], and GNUBG [Sil06] are some of the strongest backgammon programs that use Neural Networks; eXtreme Gammon is currently the supreme software [Dep12]. Different variants of backgammon[PR11; PR12], training techniques, learning methods and neural network architectures have been the focus of some researches that followed the work of Tesauro. This project will focus on studying the effect of including hybrid of Backgammon strategies to the learning network and comparing it to including the strategies separately; the introduction of those strategies will be part of the NN architecture. Many studies do not include the doubling cube in their analyses. As an extension to this project, further analyses with doubling cube will be added.

Chapter 2

Research Question

1. How does combing game strategies for backgammon affect learning?
2. Which combination of strategies results in the best performance after a fixed number of games?
3. Does using hybrid strategies improve learning outcomes?
4. would the best strategy from the previous questions be better than a basic player with a bigger hidden layer in its NN?
5. How would adding doubling cube to the setup of the previous question affect the outcome?

Chapter 3

Project Objectives

- minimum objectives
    1. create a basic backgammon player to be used as a bench player
    2. create a player that uses the Priming Game
    3. create a player that uses the Back Game
    4. create a player that combines the Priming and Back Games
    5. evaluate the performance of the players with strategies against the basic player
- intermediate objectives
    1. create a basic player with a bigger hidden network
    2. evaluate the performance of the best player with a strategy against the new basic player
- advanced objectives
    1. include doubling cube to the players
    2. perform a rollout for one of the games
    3. test the effect of including other Backgammon Game strategies

Chapter 4

Objectives Discussion

- minimum objectives

    1. Design and build a neural network architecture that will incorporated the race game strategy and a general neural network for all other positions. Define the raw and expert input features and output features; the input expert features will be based on the strategies of each neural network. Let the network train against itself (self-learning) for 100,000 games then run 500 test games of the current network against its previous weights; this will help make sure that the network is improving over time. Continue training the network and running tests until the network has been trained for 1,000,000 games.
    2. Build upon the basic player design and include another neural network for the Priming game and include the appropriate expert input features. Run the same training and testing cycle as the basic player. This player will be referred to as player-P.
    3. Build upon the basic player design and include another neural network for the Back game and include the appropriate expert input features. Run the same training and testing cycle as the

basic player. This player will be referred to as player-B.

4. Build upon the basic player design and include another neural network for each the Priming game and the Back game and include the appropriate expert input features for each. Run the same training and testing cycle as the basic player. This player will be referred to as player-PB.
5. Run 1000 test games for each of the 3 players against the basic player and record the results. Create a graph for the results. The player with the highest equity will be the best player.

- intermediate objectives

1. Increase the size of the hidden network for the basic player. Retrain the basic player with the updated NN architecture.
2. Run 1000 test games for the best player from the previous evaluation against the new basic player and record the results. Create a graph for the results.

- advanced objectives

1. Redesign all of the previous players to include doubling cube in their evaluation. Retrain all the players and test them. Evaluations of the performance will be similar to the previous objectives.
2. Perform a rollout for one of the games to check how doubling cube is being handled by each player.
3. Include another player that has different or more strategies and test it against the best player from the previous objectives. This player will follow the same training strategy as the previous players.

Chapter 5

Project Milestones

References

[Dep12] Michael Depreli. *Bot Comparison Final Table*. http://www.extremegammon.com/studies.aspx\#D2012, accessed: 20/09/2018. 2012.

[Egg17] Olivier Egger. *Backgammon Snowie*. http://www.bgsnowie.com/about/what-is-backgammon-snowie.dhtml,accessed: 28/09/2018. 2017.

[Gam17] GameSite. *eXtreme Gammon*. http://www.extremegammon.com/default.aspx,accessed: 26/09/2018. 2017.

[PR11] Nikolaos Papahristou and Ioannis Refanidis. "Training Neural Networks to Play Backgammon Variants Using Reinforcement Learning". In: *Applications of Evolutionary Computation - EvoApplications 2011: EvoCOMPLEX, EvoGAMES, EvoIASP, EvoINTELLIGENCE, EvoNUM, and EvoSTOC, Torino, Italy, April 27-29, 2011, Proceedings, Part I*. Ed. by Cecilia Di Chio et al. Vol. 6624. Lecture Notes in Computer Science. Springer, 2011, pp. 113–122. ISBN: 978-3-642-20524-8. DOI: 10.1007/978-3-642-20525-5_12. URL: https://doi.org/10.1007/978-3-642-20525-5%5C_12.

[PR12] Nikolaos Papahristou and Ioannis Refanidis. "On the Design and Training of Bots to Play Backgammon Variants". In: *Artificial Intelligence Applications and Innovations - 8th IFIP WG 12.5*

International Conference, AIAI 2012, Halkidiki, Greece, September 27-30, 2012, Proceedings, Part I. Ed. by Lazaros S. Iliadis, Ilias Maglogiannis, and Harris Papadopoulos. Vol. 381. IFIP Advances in Information and Communication Technology. Springer, 2012, pp. 78–87. ISBN: 978-3-642-33408-5. DOI: 10.1007/978-3-642-33409-2_9. URL: https://doi.org/10.1007/978-3-642-33409-2%5C_9.

[Sil06] Albert Silver. *All About GNU*. http://www.bkgm.com/gnu/AllAboutGNU.html,accessed: 24/09/2018. 2006.

[Tes02] Gerald Tesauro. "Programming backgammon using self-teaching neural nets". In: *Artif. Intell.* 134.1-2 (2002), pp. 181–199. DOI: 10.1016/S0004-3702(01)00110-2. URL: https://doi.org/10.1016/S0004-3702(01)00110-2.

[Tes92] Gerald Tesauro. "Temporal Difference Learning of Backgammon Strategy". In: *Proceedings of the Ninth International Workshop on Machine Learning (ML 1992), Aberdeen, Scotland, UK, July 1-3, 1992*. Ed. by Derek H. Sleeman and Peter Edwards. Morgan Kaufmann, 1992, pp. 451–457. ISBN: 1-55860-247-X.