Designing an Artificial Intelligence for a Complex Game

Background and Motivation:

There has been a recent surge in studies looking at developing learning AIs for games, both digital and physical. Some good examples of this are AlphaGo[1] for the board game Go, as well as Deep Blue[2] for Chess; an example of a digital game AI is OpenAI Five[3] for the video game Dota 2. These AI are developed with different pursuits in mind, but most look into training AI for real world application, using game contexts as substitutes due to their complex states and decision making requirements.

Magic: The Gathering (MtG) is a strategic card game between two players, who take the roles of duelling wizards, using a variety of spells represented by cards to reduce the other's life total to zero. Unlike traditional card games, MtG uses custom made cards in place of traditional playing cards. The game also makes use of imperfect information, in a similar vein to Bridge, where the exact contents of an opponent's hand are usually unknown. Each player uses a custom deck of cards made up from a selection of thousands of cards, which makes it hard to guess a new opponent's cards until they are played or otherwise seen.

I find MtG an interesting game due to its variety in cards, and the variety of play-styles this wide selection provides, with many cards altering how the game is played significantly. This variety also makes it an interesting case for developing AI to play MtG, as a set strategy will not always provide similar results against opposing strategies. Another facet to MtG is due to its imperfect information, there is commonly no computable optimal move, as unknown cards might make a usually optimal move sub-optimal, and vice versa.

Recent studies have shown that MtG is Turing Complete[4]; this is not directly relevant to developing an AI for the game, but it does mean that MtG is more computationally complex[4] than Chess and Go, and speculated to be the most computationally complex game in literature[4]. The cards required to induce a Turing machine state in the game will likely not be used in this project, due to complexity issues, as well as being an unrealistic scenario. However, the ability to prove MtG is Turing Complete shows the robustness of the game's logic.

This is the inspiration for this project: Is it possible to write a learning AI for MtG? The game contains complex states, as well as decision making around unknown variables, leading to a challenging environment for traditional AI. There has been research into using Monte Carlo Tree Search for card selection in order to play magic[7], but in this example they severely limit the game's complexity in order to simplify the problem. I hope to be able to develop an AI that can play with fewer limits, although some complexity and card variety – though not as much as [7] – will still be sacrificed in order to keep development time within reason.

A lot of game AIs use various types of decision trees in combination with neural networks including AlphaGo, Deep Blue, and others[6][8]. This is the approach I will likely be taking, using tree search to determine what neural network is appropriate for the current game state. The most challenging aspect of this project would be defining appropriate inputs and structure for the neural network(s).

AI Relevance

This project aims to simulate a human player, a challenging task as discussed above. This involves making intelligent decisions based on a given game state, as well as learning from game to game, improving at the game over time.

Aims and Objectives:

Aim:

To provide an AI that can learn to play Magic: The Gathering such that it provides a fun, challenging, and valuable opponent for human players.

Objectives:

1. To produce an AI capable of playing a game of Magic: The Gathering
2. The AI should be able to assess the current game state, and evaluate possible moves based upon this assessment.
3. The AI should choose what it considers the most favourable move from this evaluation.
4. The AI should be able to "learn" from playing games, and become gradually better performing.

External Aspects:

Forge[5] is an open source project to provide a digital environment to play MtG. It currently features a basic AI with a function set to get game information. This AI is not that proficient, nor complex, and can be frustrating or dull to play against if a player is sufficiently skilled. An AI that is more challenging would be more engaging to advanced players, and is something I think worth developing for this reason, and this is what I've decided to do for my project.

I believe the complexity of adding my AI to the Forge client will be achievable, due to examining the source code; it appears that Forge has a robust AI implementation for issuing instructions as a player. These inputs could be taken from a neural network output, and fed into the Forge client.

The external aspect for this project would be providing a free product for the Forge community to use. This product would be a functional AI that can be played against for practice and leisure, and perform better than the current implemented AI. If time allows, given smooth development, community feedback will be sought on how effective the AI is at achieving its goal.

Work Plan:

The first task required is to investigate and research approaches to AI creation in the context of games, taking inspiration from successful, high-profile examples such as AlphaGo. This should take around two or three weeks, and provide a good platform to develop further. In parallel to this, the second task is to examine the Forge framework to ensure that the AI is designed such that it can be easily integrated into the Forge client. This should also take about two weeks, and once both these tasks are complete, development can progress.

The next task would be to design the structure of the AI. This would include determining the input and output variables, as well as choosing how much the AI will be supervised with regards to playing the game well, how much the AI has to learn on its own. This will mean that there will be some – but limited – supervised learning in order to speed up how quickly the AI progresses. This should take about a week or two, before starting the development of the AI itself.

The majority of time for the project will then be devoted to the development and training of the AI. To start, the AI will simply assess the game state and execute an action based on its observations. Once this is working appropriately, the AI will then be made able to learn from its games, playing games on loop much faster than a human player could. This will aim to teach the AI how to play better than its initial iteration.

The last task of the project would be to release the AI to the Forge community, and receive feedback to be able to assess the AI from both a development and an interactive perspective. I will not pre-emptively ask for feedback, only once the project is complete and the AI is in a playable state.

Due to the vast quantity of cards available to play MtG with, the AI will be initially limited to certain sets so that the basic principles of the game can be learned. As the AI shows signs of being more capable, the amount of sets, and thus complexity, will be increased – giving the AI more ways to improve.

Bibliography

[1] Silver, D., Huang, A., Maddison, C., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T. and Hassabis, D. (2016). Mastering the game of Go with deep Neural Networks and tree search. *Nature*, 529(7587), pp.484-489.

[2] Campbell, M., Hoane, A. and Hsu, F. (2002). Deep Blue. *Artificial Intelligence*, 134(1-2), pp.57-83.

[3] OpenAI. (2019). *OpenAI Five*. [online] Available at: https://openai.com/blog/openai-five/ [Accessed 9 Oct. 2019].

[4] Churchill, A., Biderman, S. and Herrick, A. (2019). Magic the Gathering is Turing Complete. [online] Available at: https://arxiv.org/abs/1904.09828 [Accessed 9 Oct. 2019].

[5] Slightlymagic.net. (2019). *Forge - Slightly Magic*. [online] Available at: https://www.slightlymagic.net/wiki/Forge [Accessed 9 Oct. 2019].

[6] Barratt, J. and Pan, C. (2017). Playing Go without Game Tree Search Using Convolutional Neural Networks. [online] Available at: http://cs231n.stanford.edu/reports/2017/pdfs/603.pdf [Accessed 16 Oct. 2019].

[7] Ward, C. and Cowling, P. (2009). Monte Carlo Search Applied to Card Selection in Magic: The Gathering. [online] Available at: https://web.archive.org/web/20160528074031/http://scim.brad.ac.uk/staff/pdf/picowlin/CIG2009.pdf [Accessed 16 Oct. 2019].

[8] Silver, D. and Huang, A. (2012). Mastering the Game of Go with Deep Neural Networks and Tree Search. [online] Available at: http://www0.cs.ucl.ac.uk/staff/d.silver/web/Publications_files/unformatted_final_mastering_go.pdf [Accessed 16 Oct. 2019].