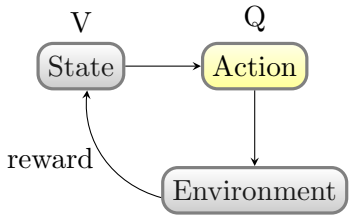
Report for 4YP Project:
Gaming AI

Stephen Lilico

February 29, 2016

Contents



1 Literature Review

This section details the mathematical framework and previous research on which this project was based.

1.1 Reinforcement Learning

Reinforcement learning is

Reinforcement learning is

"a technique where an agent attempts to maximise it's reward by repeated interactions with a complex uncertain environment."[1]

A MDP is a discrete time stochastic control process.

- Value function
- Q function
  - On Policy
  - Off Policy

1.1.1 Temporal Difference Methods

- Temporal Difference methods
  - TD(0)

$$V(s_n) = V(s_n) + \alpha(\gamma V(s_{n+1}) + r - V(s_n))$$

  - TD(λ)

$$R_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty} \lambda^{n-1} R_t^{(n)}$$

$$V(s_n) = V(s_n) + \alpha(R_t^\lambda - V(s_n))$$

1.1.2 Policy Gradient Methods

REINFORCE Updates the parameters of the function approximation directly

$$\nabla_\theta J = \frac{1}{M}\sum_{i=1}^{M}\sum_{t=1}^{T} \nabla_\theta \log \pi(a_t^i | s_{1:t}^i; \theta)(R^i - b_t)$$

1.2 Deep Q-Learning

1.3 Recurrent Models of Visual Attention

1.4 Deterministic Policy Gradient

1.5 Asynchronous Methods for Deep Reinforcement Learning

[1] taken from [**Sutton:1998:IRL:551283**]

2 The Initial Project

2.1 Context

Producing an AI that can play complex games well is a very difficult task. In field, TDgammon (a backgammon playing agent that uses reinforcement learning) made a significant stir when it managed to achieve world class competency, and more recently the paper from Google deep mind that uses the same reinforcement learning structure that plays ATARI games using only receives the screen as input triggered further interest in the field. And the results are not limited to game playing; the results from the deep mind paper underpin recent work that uses end to end training to produce significant performance improvements in visuomotor control, and more generally some robots have to function in adversarial environments.

2.2 Initial Aims

3 Intial Implimentations

3.1 Maze solving agents

4 Magic: the Gathering

The particular game type this project was chosen for was collectible card games, in particular starting with Magic: the Gathering (MtG). These games provide a number of interesting and difficult challenges for AI: uncertain information, stochastic results, variable action spaces, along with additional opportunity for further depth should deck building also be considered. They are also turn based and don't require a physics engine, so they can run through many iterations of play quickly. Of these, MtG was chosen as it represents both a significant breadth of possibilities and different interactions without excessive card complexity or specificity and it has a more approachable learning curve than most. Hearthstone was considered, but a suitable emulator was hard to find due to the copyright issues.

4.1 Aims

This project aims to produce an architecture that can learn to play at or around human competency trading card games. Ideally this would involve use of novel techniques that could find other use as well.

4.2 Goals

- Produce an architecture that can learn to play MtG to an acceptable level of competency with a given deck
- Produce an architecture, possibly using the above architecture, that can produce and improve decks of it's own and learn to adapt to changes in metagame.
- generalise this architecture to be able to learn and play other card games

5 Completed Work

5.1 Initial Setup

A suitable open source emulation environment for playing MtG was identified, and modifications were made to it to allow the learned AI to be used in it and trained agains the extant rules based AI. Neural Networks libraries for java that use the GPU were installed and unit tested. An overview of AI techniques and the particulars of standard reinforcement learning algorithms were read. The exact state size was considered, and some additional restrictions were made on the type of cards that would be used within MtG's 15,000 card pool to reduce the initial complexity.

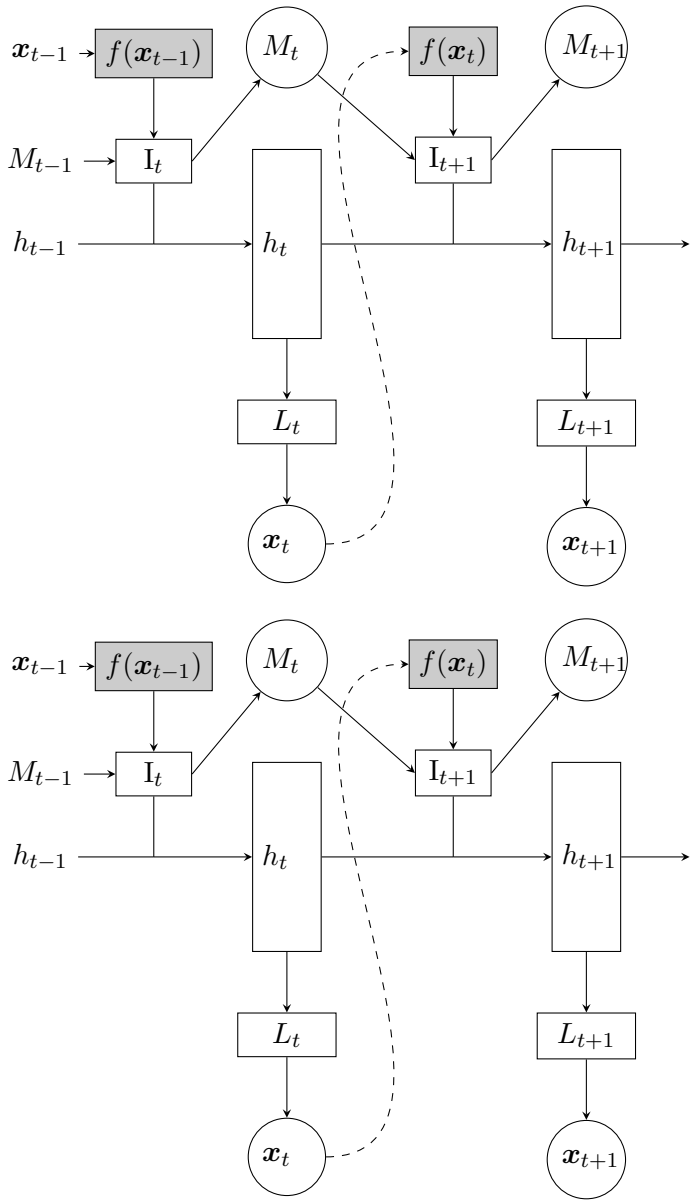
5.2 Preliminary Development

As an initial test of the neural network architecture with reinforcement learning, a simple maze enviroment was made. Currently the agent is not converging to a solution, and so the details of the implimentation are being looked at.

6 Initial Changes

To get a firmer grip on reinforcement learning techniques, the maze agent will be tested with a number of tabular methods as well as the current function approximation basede technique.Then the structure for the preliminary reinforcement learning agent to play MtG will be made. Initially it will only choose what card to play, then other play decisions. It's likely a naïeve approach won't be sufficient, so a number of possible improvements are on the boards, with more to be considered:

7 Final Assessment of Feasibility



8 Function Optimization

9 References