The University of Manchester
Department of Computer Science
Project Report 2020

Reinforcement learning for a learnable agent in classic arcade games

Author: Sean J Parker

Supervisor: Dr. Konstantin Korovin

Abstract

Reinforcement learning for a learnable agent
in classic arcade games

Author: Sean J Parker

The aim of the project is to investigate the performance of Gismos and to design and construct a super multi-functional Gismo.
The novel aspects of the new Gismo are described. The abstract should perhaps be about half a page long.
The results of testing, which show the abject failure of the Gismo, are presented.
In the conclusions proposals for rectifying the deficiences are outlined.

Supervisor: Dr. Konstantin Korovin

Acknowledgements

I would like to thank my parents, my school teachers, my friends, my wonderful supervisor, and all my wonderful fellow students for their unswerving support during my project. Without your help none of this would have been possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

My project aims to replicate some of the reinforcement learning algorithms that can be used to play classic Atari 2600 games. It also compares the results of different tests with these algorithms such as varying the hyperparameters of the network. By observing the effects on the trained agents[1] when the hyperparameters are changed we can deduce a set of optimal values such that the networks can play three different Atari 2600 games. Overall, the main features of the project are the following:

- Agents with raw pixel game data as input, outputting a set of values for the best action.
- Agents attempt to find an optimal model of the environment without any prior knowledge.
- Visualization of the agent "brain" to provide insight into what information the agents is learning.

1.1 Motivation

Over the past 10 years there has been significant improvement in the RL (reinforcement learning) algorithms. One reason is that the computing power has become cheaply available by using discrete graphics cards. For example, for my project I used a Nvidia GeForce GTX 1070 that provides 1920 CUDA cores that can be used to accelerate training of neural networks. Despite this, RL algorithms are massively computationally expensive and hence take a long time to train.

Over recent years one of the pioneers in this area is DeepMind, which was acquired by Google in 2014, and they developed the DQN (deep q-network) algorithm in 2013 which they demonstrated could learn directly from the raw pixel data of games in order to achieve either human-level or super-human level performance.

This research was expanded upon by DeepMind and OpenAI which is based on the original DQN by DeepMind. This research focused on trying to approximate a Q-function and thereby infer the optimal policy. On the otherhand, there has recently been a focus on other methods such as A3C and PPO which instead seek to directly optimise in the policy space of the environment.

[1] Agent. In our case, agent refers to a trained neural network that takes actions in a chosen environment.

1.2 Objectives

Further to what was described in section 1 there was a few main objectives of the project. Firstly, I chose three games on which I decieded to train the agents, Pong, Breakout, and Space Invaders which are shown below in Figure 1.1

TODO: include gantt chart

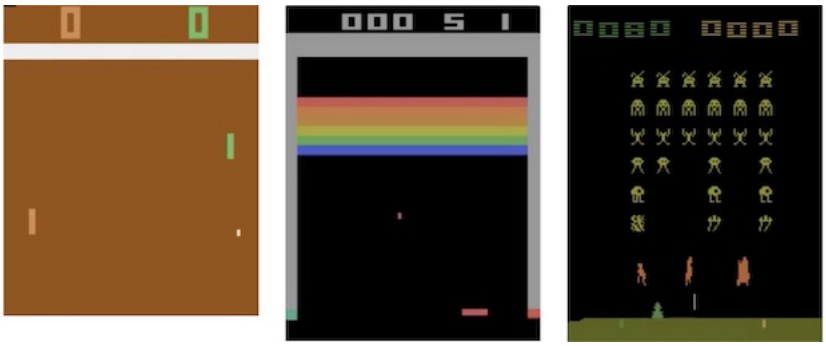


Figure 1.1: Screenshots of Pong, Breakout, Space Invaders. Left to right.

1.3 Report structure

My report is divided into three main sections. Firstly, describing the background of the problem, then going onto giving details of my implementation and finally project evaluations/conclusions.

Chapter 2

This chapter has a very long title that would be too long for using in headers and table of contents

2.1 Image processing

Because Image processing is such a large topic no attempt will be made here to describe the entire subject here but only areas directly relevant to the project as a whole. Note how we can cross reference sections of the document such as section 2.2

2.2 Low level operations

Here is an example of a list with bullets:

- *Mean* Take the mean value of the neighbourhood.
- *Median Filter* Take the median value of the neighbourhood.

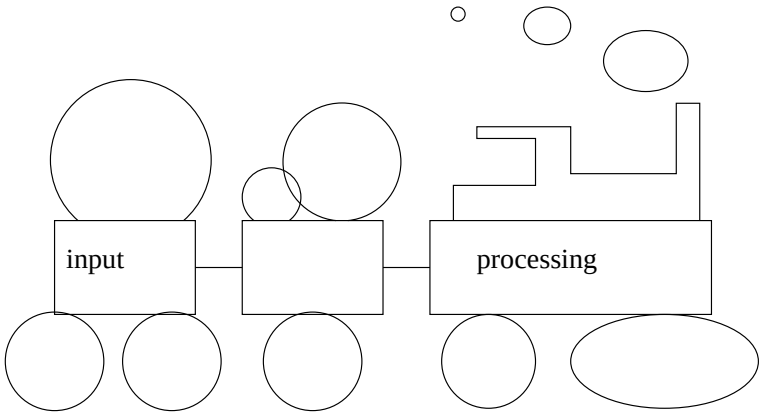
Here are a couple of mathematical formulae:

$$\Delta_1 = I(x,y) - I(x+1,y+1)$$
$$(x-a)^2 + (y-b)^2 = R^2$$

A table is just like a figure. Table 2.1 uses the tabular environment. Environments such as tabular can be used in ordinary text as well.

place	1991	1992	1993
CS Dept	1	99	199
Owens Park	1876	22	0
Academy	0	0	99999

Table 2.1: Distribution of Wombats in Greater Manchester



A caption for the picture

Figure 2.1: Final version of the system

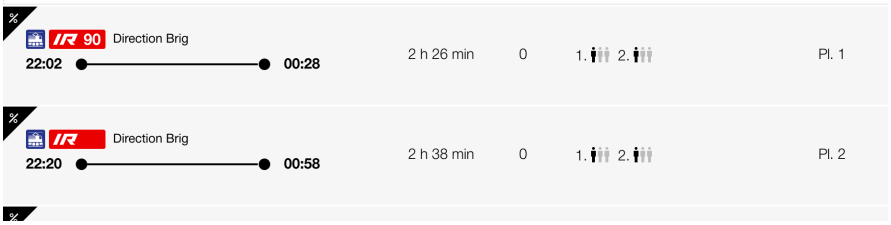


Figure 2.2: A screen dump

2.3 Creating Diagrams

Figure 2.1 is a figure previously prepared using the `xfig` interactive drawing package. With the latest version of `xfig` it is quite easy to incorporate `xfig` diagrams. The `xfig` package is menu driven and reasonably self explanatory.

Use the *Export* menu option to save an encapsulated PostScript version of your figure, which can then be included in the document using the `\includegraphics` command. Choose the *Portrait* orientation option. For example, if the figure is in the file figure1.fig, the exporting process will create a file figure1.eps, which can then be included, scaled to whatever height or width you want.

Note that, if you want to use such graphics facilities in some other document, which does not use either the `third-rep` or `muthesis` document class, you will need to put the command `\usepackage{graphicx}` after the `\documentclass...` command in the main file.

2.4 Screen Dumps

Screen dumps can often enhance the appearance and clarity of a report, although care should be taken not to overuse them. When using screen dumps it is useful to make use of LaTeX's capability for using compressed PostScript files, as in the example shown in figure 2.2.

The image is first captured using, for example, `xv`, and saved in a file, e.g. `screen.ps`. Next, extract the bounding box information from the head of this file. In this case it is

```
%%BoundingBox: -100 -16 697 860
```

and create a file with name given by adding `.bb` to the original file name (in this case the name is `screen.ps.bb`) which contains just this bounding box. You can now compress the original file, using `gzip`. The command `\includegraphics[width=12cm]{screen}` will look for either screen.ps or screen.ps.gz, so you can compress or uncompress the ps file without changing the latex source.

Alternatively, save as `.png` if using `pdflatex`.

The `draft` option can be used to exclude the actual figure, but leave an appropriate amount of space.

You may not be able to see the screen dump if you are viewing this file from WWW. This is due to an unfortunate 'feature' of the previewer xdvi. If you view the files directly from `/opt/info/doc/latex/3rd-yr`, things should work properly.

Chapter 3

Making a bibliography

Whenever you wish to refer to books or articles relevant to your report you should use a citation such as [Lam94]. You can also force entries to appear in the bibliography without a citation appearing in the document, by using `\nocite`.

Each document cited must have an entry in a .bib file. For this document we have only one, called refs.bib. These files are listed in the `\bibliography` command at the end of report.tex. Note that the .bib files can (and often do) contain many more entries than are actually cited in a partcular document; the only ones that appear in the bibliography are those that have been referenced using `\cite` or `\nocite`.

In order to generate the appropriate reference entries, you will need to run `bibtex` after `latex` has been run, using the command `bibtex report`. This will generate a file report.bbl, which contains the bibliography entries. Once that file is there, you do not need to run `bibtex` again unless you add new citations, but you will probably have to run `latex` twice after running `bibtex` the first time.

The TEX FAQ ([Tea12]) gives tips on how to cite URLs.

The file refs.bib provides an example of what can be done with BibTEX. You can find much more information in any book on LATEX, for example [Lam94, GMS94, KD95]

References

[BT88] R. D. Boyle and R. C. Thomas. *Computer Vision - A First Course*. Blackwell Scientific Publications, 1988.

[Dyk94] William Dyke. The nutter's guide to LaTeX. *Read Only*, 7, May 1994.

[GMS94] Michel Goosens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley, 1994.

[KD95] Helmut Kopka and Patrick W. Daly. *A guide to LaTeX 2ε*. Addison-Wesley, 2nd edition, 1995.

[Lam94] Leslie Lamport. *LaTeX – A document preparation system*. Addison-Wesley, 2nd edition, 1994.

[Tea12] LaTeX Team. URLs in BibTeX bibliographies. `http://www.tex.ac.uk/cgi-bin/texfaq2html?label=citeURL`, 2012.