# Teach AI play the Atari ice hockey game by videos

**Yangyang Xu**

A report submitted for the course
COMP8755 AND Individual Computing Project
Supervised by: Dr. Penny Kyburz
The Australian National University

May 2019

Except where otherwise indicated, this report is my own original work.

Yangyang Xu
13 May 2019

# Contents

Chapter 1

# Atari Gym:Ice Hockey

Put your introduction here. You could use \fix{ABCDEFG.} to leave your comments, see the box at the left side.

You have to rewrite your thesis!!!

## 1.1 Problem Statement

Describe the problem your project addresses.

## 1.2 Motivations

## 1.3 Project Scope

Describe the problem your project addresses.

## 1.4 Report Outline

How many chapters you have? You may have Chapter **??**, Chapter **??**, Chapter **??**, Chapter **??**, and Chapter **??**.

Chapter 2

# Learning from Demonstration

## 2.1 Playing hard exploration games by watching YouTube

## 2.2 Player Experience Extraction from Gameplay Video

## 2.3 Learning Montezuma's Revenge from a Single Demonstration

## 2.4 Deep Q-learning from Demonstration

How many chapters you have? You may have Chapter **??**, Chapter **??**, Chapter **??**, Chapter **??**, and Chapter **??**.

Chapter 3

# State Embeddings

## 3.1 Dynamic image encoder

## 3.2 Stack of difference of frame video-clip encoder

## 3.3 Grad-CAM ++

The Class Activation Mapping has been generalised into Gradient Class Activation Mapping (Grad-CAM). The interpretability of Grad-CAM in deep network can lead researchers to failures of model. For the interpretability, the Grad-CAM can discriminate the image classes by localising the region of interest (ROI) in a image.

Grad-CAM uses CNN architecture, the gradient mappings obtained from CNN can be used in the final fully connected layer to significant the regions of interest in a image.

Where to get the gradient information

$$\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_i^k j}$$

$\alpha_k^c$ :

$\frac{1}{Z}$ : global average pooling

$y^c$ :

$A_i^k j$ : A matrix of K feature map

$\frac{\partial y^c}{\partial A_i^k j}$ :

$$L_{Grad\_CAM}^c = ReLU(\sum_k \alpha_k^c A^k)$$

$ReLU$: it's good to see the positive result

Grad-CAM ++ is better for video-based training.

Why "++" better for video? How differ from Grad_CAM?

## 3.4 VCG

How many chapters you have? You may have Chapter **??**, Chapter **??**, Chapter **??**, Chapter **??**, and Chapter **??**.

Chapter 4

# RL

## 4.1 Definitions

Agent: An agent takes actions

Action (A): A is the set of all possible moves the agent can make

Environment: The world through which the agent moves. The environment takes the agent′ current state and action as input, and returns as output the agent′s reward and next state

State (S): A state is a concrete and immediate situation in which the agent finds itself

Reward (r): A reward is the feedback by which we measure the success or failure of an agent′ s actions

Discount factor ($\gamma$): The discount factor is multiplied with future rewards as discovered by the agent in order to dampen their effect on the agent′ s choice of action. It makes future rewards worth less than immediate rewards

Policy ($\pi$): The policy is the strategy the agent employs to determine the next action based on the current state. It maps states to actions

Value (V): The expected long-term return with discount, as opposed to the short-term reward r. $V\pi(s)$ is defined as the expected long-term return of the current state under policy $\pi$

Q-value or action-value (Q): Q-value is similar to Value, except that it takes an extra parameter, the current action a. $Q\pi(s,a)$ refers to the long -term return of the current state′ s, taking action a under policy $\pi$. Q maps state-action pairs to rewards

## 4.2 Off-Policy

## 4.3 On-Policy

## 4.4 Model Based

## 4.5 Model Free

## 4.6 DQN

Use the immediate reward we receive and a value estimate of our new state to update the value estimate of original state-action pair. We only had the learned value function Q-function and the policy we followed was simply taking the action that maximised the Q-value at each step

## 4.7 A3C

Actor-critic methods combine policy gradient methods with a learned value function. we learn two different functions: the policy (or "actor"), and the value (the "criti"). The "policy" adjusts action probabilities based on the current estimated advantage of taking that action, and the value function updates that advantage based on the experience and rewards collected by following the policy

How many chapters you have? You may have Chapter **??**, Chapter **??**, Chapter **??**, Chapter **??**, and Chapter **??**.

Chapter 5

# IRL

## 5.1 Apprenticeship Learing

## 5.2 Bayesian Inverse Reinforcement Learning

## 5.3 Maximum Entropy Reinforcement Learning

## 5.4 Generative Adversarial Imitation Learning

## 5.5 Project Scope

Describe the problem your project addresses.

## 5.6 Report Outline

How many chapters you have? You may have Chapter **??**, Chapter **??**, Chapter **??**, Chapter **??**, and Chapter **??**.

Chapter 6

# Policy Optimisation

## 6.1 Dynamic image encoder

## 6.2 Stack of difference of frame video-clip encoder

Using a stack of differences of frames to capture motion and dynamics from a video clip. It was helpful to learn odd-one-out

How many chapters you have? You may have Chapter **??**, Chapter **??**, Chapter **??**, Chapter **??**, and Chapter **??**.

# Bibliography