

B.Tech Project Report

DECISION MAKING USING DEEP REINFORCEMENT LEARNING

Submitted in partial fulfilment for the award of the Degree of
Bachelor of Technology in Computer Science and Engineering

Submitted by

Jayadeep K M (Roll No 13400030)
Kevin Joseph (Roll No 13400032)
Mohammed Nisham K (Roll No 13400038)

Under the guidance of
Mr. Vipin Vasu A V



Department of Computer Science and Engineering
COLLEGE OF ENGINEERING, TRIVANDRUM
KERALA
MAY 2017

CERTIFICATE



This is to certify that the thesis entitled “Decision making using deep reinforcement learning” is a bonafide record of the major project done by **Jayadeep K M** (Roll No 13400030), **Kevin Joseph** (Roll No 13400032) and **Mohammed Nisham K** (Roll No 13400038) under my supervision and guidance, in partial fulfilment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering from the University of Kerala for the year 2017.

Mr. Vipin Vasu A V
(Guide)

Asst. Professor

Dept. of Computer Science and Engineering

Mrs. Liji

Professor and Head

Dept. of Computer Science and Engineering

Place: Trivandrum

Date: 11-05-2017

Abstract

Creating a general purpose AI has been an area of research since the beginning of computers and programming. Reinforcement learning is a major step towards a general purpose AI.

This project is aimed at creating a program that can learn to make decisions in an environment that is defined by a high-dimensional input, and has sparse and time delayed rewards for these actions. Such programs can be useful in problems where decision must be made based on high dimensional sensory input such as camera feed. This project uses Q-learning algorithm to assign a quality value to each action in a state of the environment.

Atari games are used to demonstrate this approach, by training the program to play breakout game for upto 50 epochs and observing performance improvement. The trained neural network was saved and tested at the end of every epoch. The performance parameters like average q-value, average reward, games per epoch were also saved. The performance parameters showed a clear rise in performance for breakout (50 epochs) and space invaders (8 epochs).

The project has applications in the field of IOT, security, gaming, stock market analysis and traffic control systems. Any system that can be modelled as an environment with actions and rewards can be trained using this algorithm.

Contents

1	Introduction	1
1.1	Motivation and Overview	1
1.2	Literature Survey	1
2	Materials and Methodology	1
2.1	Algorithms	1
2.1.1	Q-Learning	1
2.1.2	Deep Q-Learning	1
2.2	Program Development	2
2.2.1	System Description	2
2.2.2	Class Diagram	2
2.2.3	Code Overview	3
3	Results and Discussions	3
3.1	Description of observed strategies	3
3.2	Screenshots	3
3.3	Result Visualization	3
4	Further Work	3
4.1	Gamification of Problem	3
4.2	Traffic Light Control	3
4.2.1	Problem	3
4.2.2	Minimal Input	3
4.2.3	Reward and Action set	3
4.2.4	Scalability	3
5	Conclusion	3

List of Figures

1 UML Class Diagram. 2

Abbreviations

AI Artificial Intelligence

RL Reinforcement Learning

1 Introduction

1.1 Motivation and Overview

Learning to control agents directly from high-dimensional sensory inputs like vision and speech is one of the long-standing challenges of Artificial Intelligence and Machine Learning. Most successful AI applications have relied on hand-crafted features combined with linear value functions or policy representations. Clearly, the performance of such systems heavily relies on the quality of the feature representation.

Recent advances in deep learning have made it possible to extract high-level features from raw sensory data, leading to breakthroughs in computer vision and speech recognition. These methods utilise a range of neural network architectures, including convolutional networks, multilayer perceptrons, restricted Boltzmann machines and recurrent neural networks, and have exploited both supervised and unsupervised learning.

However the main advantage of Reinforcement Learning is that it doesn't need huge amounts of hand-labelled data and does not depend too much on the feature representation. It learns from a reward signal that maybe delayed, noisy and sparse. The delay between actions and the rewards, which maybe thousands of timesteps seems like a particularly hard problem in RL when compared to direct association between action and reward in supervised learning. Another issue is that most problems consider all the data samples independent of each other, but in this case we need to consider the fact that the reward at the end of a session is not just the result of the last action but the result of the sequence of actions from the start of the session.

This project is aimed at using a Convolution Neural Network along with the Q-learning algorithm to solve the above problems and make decisions based on video input from the Atari Learning Environment. Our goal is to create a single neural network agent that is able to successfully learn to play atleast 2 games with no change in agent algorithm. The network was not provided with any game-specific information or hand-designed visual features, and was not privy to then internal state of the emulator, it learned from nothing but the video input, the reward and terminal signals, and the set of possible actions just as a human player would.

1.2 Literature Survey

2 Materials and Methodology

2.1 Algorithms

2.1.1 Q-Learning

2.1.2 Deep Q-Learning

Neural Networks

2.2 Program Development

2.2.1 System Description

2.2.2 Class Diagram

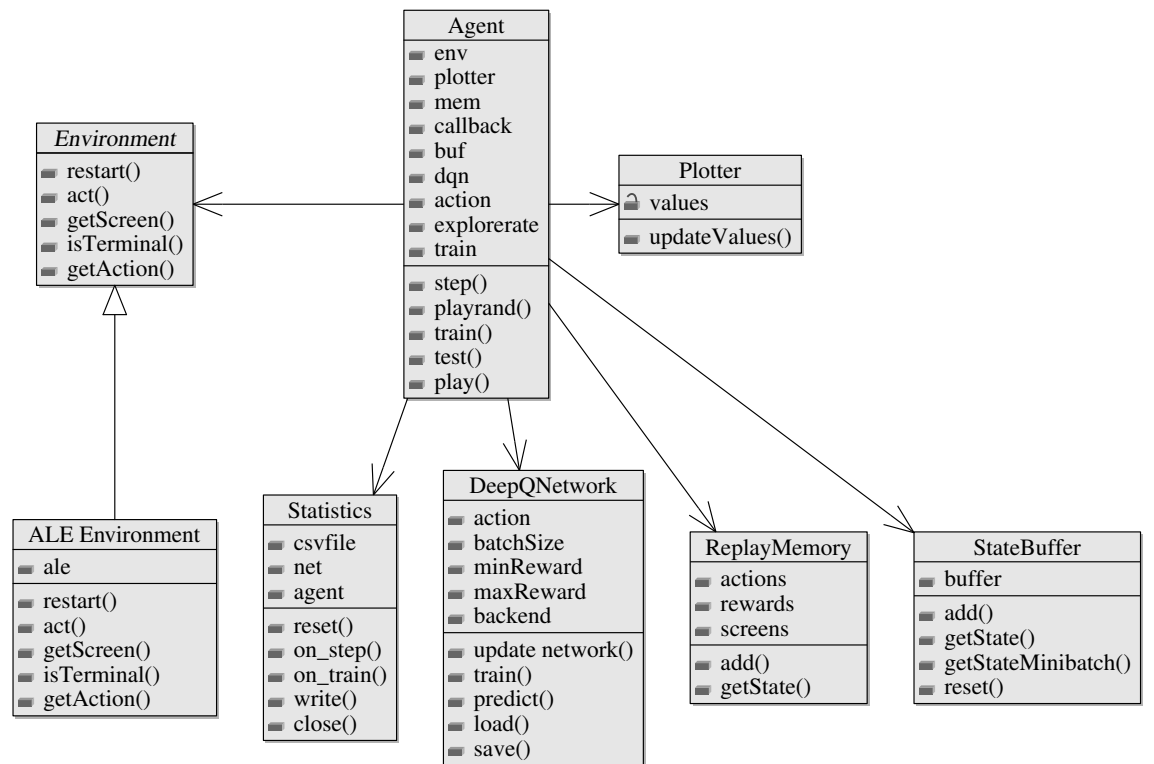


Figure 1: UML Class Diagram.

2.2.3 Code Overview

3 Results and Discussions

3.1 Description of observed strategies

3.2 Screenshots

3.3 Result Visualization

4 Further Work

4.1 Gamification of Problem

4.2 Traffic Light Control

4.2.1 Problem

4.2.2 Minimal Input

4.2.3 Reward and Action set

4.2.4 Scalability

5 Conclusion