**SOFTWARE REQUIREMENTS SPECIFICATION (SRS) FOR**

# Decision making using Reinforced Deep Learning


**November 13, 2016**

**Prepared for:**
Btech Project Interim Evaluation

**Prepared by:**
Kevin Joseph, Jayadeep K M, Mohammed Nisham K

**Guide:**
Vipin Vasu
vipin@cet.ac.in

**College Of Engineering, Trivandrum**

## TABLE OF CONTENTS

**Section Page**

# 1 INTRODUCTION

The project is aimed at creating a deep learning model capable of taking high-dimensional data as input and taking a particular action as output. The major part of the model is a Neural Network, trained using Q-learning. To demonstrate the model we use the model to play multiple games taking the visual pixel data as input.

## 1.1 IDENTIFICATION

The software system being considered for development is referred to as Samantha. Samantha will serve as a decision making AI in places where the environment is given as input in the form of high-dimensional unstructured data. This is a new project effort, so the version under development is version 0.1.

## 1.2 PURPOSE

The project will serve as a decision making AI in places where the environment is given as input in the form of high-dimensional unstructured data. For example it can be used for automated power control in IOT Home/office environment. In such a scenario it will learn the patterns present in the environment and corresponding actions being taken and then will replicate it.

## 1.3 SCOPE

The project will be demonstrated using multiple graphical games, the game screens will be given as input to the program. Initially the it will play like a dumb player, but after each episode it will play better and better till it converges to the best possible game play. The program will mimic the way a human learns to make decisions in a new environment.

## 1.4 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

| Term or Acronym | Definition |
|---|---|
| Alpha test | Limited release(s) to selected, outside testers |
| Beta test | Limited release(s) to cooperating customers wanting early access to developing systems |
| Final test | Acceptance test, release of full functionality to customer for approval |
| DFD | Data Flow Diagram |
| SDD | Software Design Document, aka SDS, Software Design Specification |
| SRS | Software Requirements Specification |
| NN | Neural Network |
| MDP | Markov Decision Process |

## 1.5 REFERENCES

[1] D. S. A. G. I. A. D. W. M. R. Volodymyr Mnih, Koray Kavukcuoglu, "Playing atari with deep reinforcement learning," Dec 2013.

[2] "Demystifying deep reinforcement learning." https://www.nervanasys.com/demystifying-deep-reinforcement-learning/.

[3] "Deep reinforcement learning: Pong from pixels." http://karpathy.github.io/2016/05/31/rl/.

[4] "q-learning path-finding-q-learning-tutorial."
http://mnemstudio.org/path-finding-q-learning-tutorial.htm.

[5] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The arcade learning environment: An evaluation platform for general agents," *Journal of Artificial Intelligence Research*, vol. 47, pp. 253–279, 06 2013.

# 2 OVERALL DESCRIPTION

The project is aimed at creating a human like learning model that is able to learn from data similar to audio-visual data. The input will be 256x256 pixel screens of simple atari games. The idea is to observe the game state and make decisions based on expected rewards, some actions may result in an instant reward, some sequence of actions may result in a future reward, for both these cases the program must observe and modify an NN to do the best action next time. It must also consider taking random actions initially inorder to learn, later the confidence of actions must increase causing random moves ton decrease and finally the program should converge to a best player state.

## 2.1 PRODUCT PERSPECTIVE

The practical use of this project can be seen in applicaations where decisions need to be made on the basis of a complex environment represented in the form of large amounts of sensory data. Examples are IOT homes, automated vehicles, stock market. It is evident in these application that it may not be possible to provide a defined feature set making it impossible to use normal machine learning approaches that make predictions based on the values of a predefined feature set, such an algorithm is only good if the feature set is good. This project takes the whole data instead and forms a relavent feature set by itself.

## 2.2 PRODUCT FUNCTIONS

- It should be able to play multiple games without any change in the architecture or algorithm.
- It should converge in a finite number of episodes.
- It should approximate best action even in states it has not encountered before.
- It should learn by playing the game repetitively without user intervention.
- It should have an generic interface with the game both on the input and the control end (Atari Learning Environment).

## 2.3 USER CHARACTERISTICS

The users of this program are developers who need to include a decision making AI in their application. Users must be experts in their domain and must have a clear idea of which part of their data represent environment states, number of controls,etc

## 2.4 CONSTRAINTS

- Processor performance.
- Game Speed.

## 2.5 SYSTEM FEATURES

The project views the game as an MDP with a sequence of actions, rewards and states. Every state has a set of valid actions and each action leads to a new state, some actions result in a reward and some result in future rewards. ANN is used to output a quality value for each action and the best action is found. In the training phase the best action may not always be chosen, this is done inorder to solve the exploration-exploitation problem.
The above process must be repeated till the player converges. This must happen in a finite number of episodes in a finite amount of time. The algorithm must act as a blackbox algorithm and must be able to play multiple games with same set of controls without any modifications.