Project Proposal - Mario Game Agent - Reinforcement learning

October 18, 2015

1 Abstract

For this project we will be implementing various reinforcement algorithms to create learning agents to effectively play a variant of the classic Super Mario Bros game. The game is challenging form a AI standpoint as the environment is stochastic as the levels are randomly generated and the state space is large. To address these challenges we will use Q-learning with various Q function approximations from simplistic feature extractors to deep learning implementations.

2 Task Definition

The game is defined by a simulator that receives actions from our controller and outputs various environment attributes. Our controller will extract a state from the environmental attributes and rewards. The rewards will be a combination of metrics such as positive rewards from coins collected, monsters destroyed, level completion and negative rewards for collision and death.

Actions - { Left, Right, Jump, Crouch, Run/Fire}

State

Status - Running, Win, Dead

Mode - Small, Large, Fire

Ground - True, False

isAbletoJump - True, False

hasShell? - True, False

ableToShoot - True, False

CreaturePositions - Array of [x,y] coordinates

Obstacles - 22x22 array of obstacle type

3 Challenges

The main challenge is to find a optimal policy for Mario to follow given that the map of the environment is not known in advance. The actions which are taken will lead to rewards in future timesteps. Additionally we do not know the transition function thus it would be difficult to use offline methods. We would also like our algorithm to run in real time which means we need to complete all calculations in at most 42ms. It is not apparent what the optimal feature extraction fuction is for this specific problem and this we will experiment with hand crafted feature selection as well as deep learning approaches.

4 Benchmarking

Our baseline method will be a random action method in which Mario picks a random action with a forward moving bias. We chose this as our baseline method as the controller makes moves regardless of state. Our oracle is an expert human player whos score we will take a max of over several games. We will use our own scores, played under the condition of slower frame frames, as a proxy for this expert human player. The difference between the two bounds should provide a good prospective on the improvement that can be made.