Hierarchical Reinforcement Learner

Nick: 蹶く者 (MogakuMono)

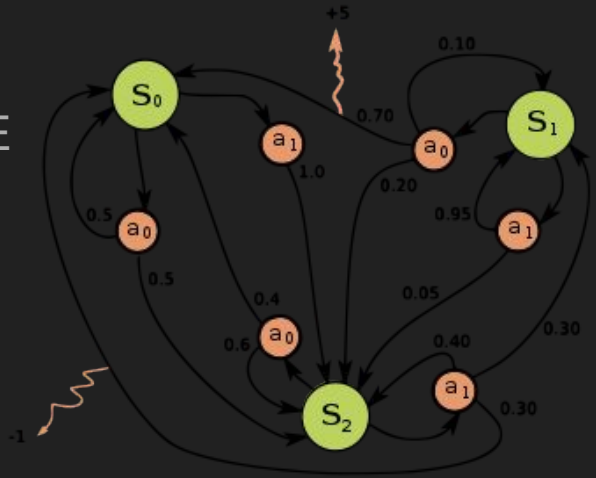
*Ivan de Jesus P.P. , Ginaldo Oliveira Morais
Graduate Students

*Intelligent Distributed Systems Laboratory(LSDI)
Federal University of Maranhão(UFMA) - Brazil

*navi1921@gmail.com , sr.ginaldo.oliveira@gmail.com

Reinforcement Learning

- We implemented a Reinforcement Agent that plays ICE FG
- Usually the Game is modelled as a Markov Decision Problem
  - State: players position, life, motions, etc.
  - Actions: game inputs: up,down,left,right,a,b
  - Rewards: Damage inflicted and received.
- Simple Reinforcement learning has to learn a optimal policy to solve the MDP
  - That policy has to learn many differents combos, and evaluate the best strategy with them



Hierarchical Reinforcement Learner

- Learns a SMPD, that allows extended actions
- We call these extended actions “behaviours”
  - Can also been seen as Options
- We as players of fighting games, used our domain knowledge to create these behaviours
  - Combos, movimentation
- We innovate by inducing behaviours from the Monte Carlo Search tree
- We manually create features describing the state
  - Ours and the opponent positions, distance from each other, projectiles on the field, etc

Hierarchical Reinforcement Learner

- We use a Perceptron as the Function approximator
  - Easy to debug and understand

- We use Q-Learning for the training

- Training against the Mcts Sample did not show better results.

Partial Results

- Zen learned a aggressive policy, with focus on Aerial attacks
  - Has combo corner, very specific situation
- Garnet plays a dance-like fight. Might not be too efficient, but it is beautiful.
- Lud plays a grab strategy, with damaging combos. They can been fled though, by jumping.
- All 3 AIs shows promise, with the flaw of being too much aggressive/pro-active.
- While the behaviours haven been made with domain knowledge, the AI chooses what and when to use.