Applying Modern Reinforcement Learning to Play Video Games

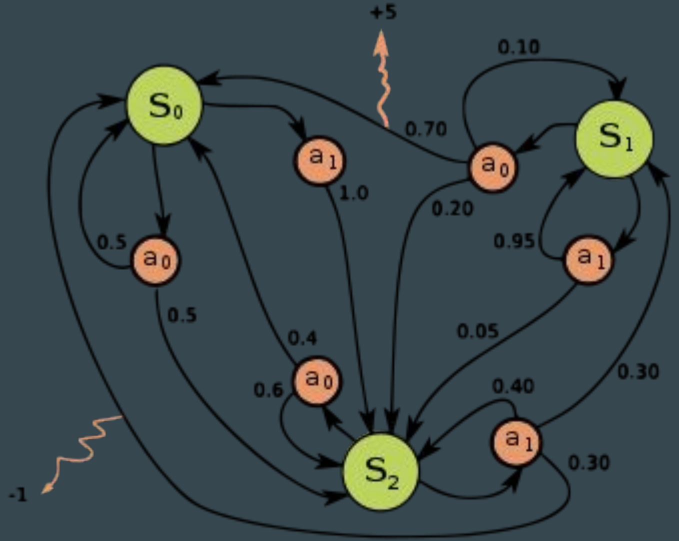
Computer Science & Engineering
Leung Man Ho
Supervisor: Prof. LYU Rung Tsong Michael

Outline

- Term 1 Review
- Term 2 Objectives
- Experiments & Results
- Online Evaluation Platform
- Future Work

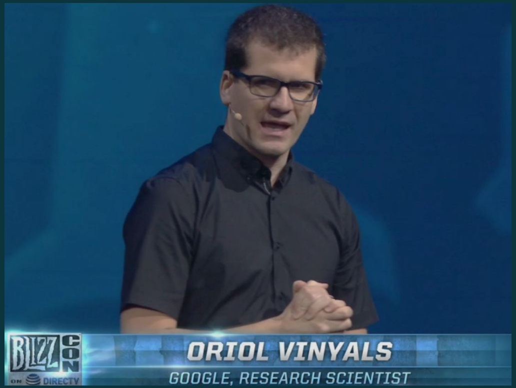
Term 1 Review - Background

- Reinforcement learning is learning what to do - Prof. Richard S. Sutton
- Often modelled as Markov Decision Processes
  - S: a finite set of states.
  - A: a finite set of actions.
  - T(s'|s, a): Transition model
  - R_a(s, s'): Reward model
  - γ: future discounted factor
- Objective
  - $\sum_{t=0}^{\infty} \gamma^t R_{a_t}(s_t, s_{t+1}),$
  - Maximize discounted future reward



Term 1 Review - Motivation

- Explore the boundary of modern RL
- Selected a challenging, unexplored and meaningful video game



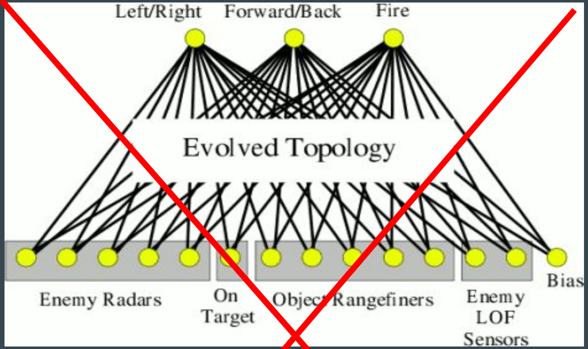
Why video game? Why is it meaningful?

"At DeepMind, our mission is to solve intelligence and use that to solve complex real world problems, but in order to do that, we need to test our algorithmic ideas in challenging environments." - BlizzCon on DeepMind x Starcraft II

Term 1 Review - Little Fighter 2

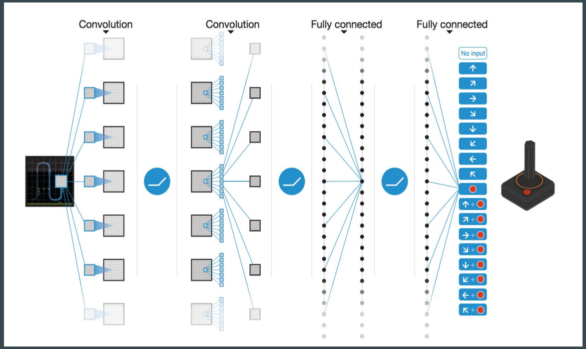
- LF2
  - Developed by CUHK Alumni
  - Visual fighting game
  - Very popular in HK
- Game
  - HP & MP
  - 7 keys, {up, down, left, right, attack, jump, defense}
  - Special abilities for each character, triggered by key sequences
  - Exploitable game objects

Term 1 Review - Methods



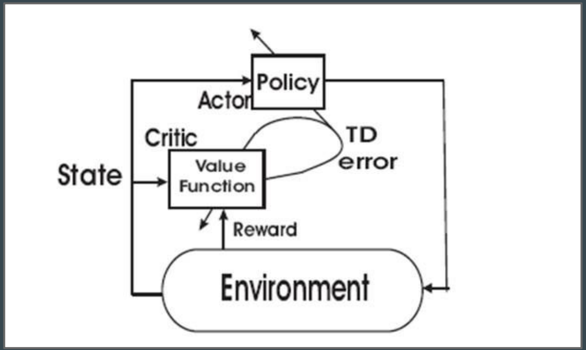
NeuroEvolution of Augmenting Topologies

- NEAT
- Proposed in 2002
- Evolutionary method



Deep Q-Network

- DQN
- Proposed in 2014
- Value iteration method



Actor Critic using Kronecker-Factored Trust Region

- ACKTR
- Proposed in 2017
- Actor critic method

Term 1 Review - Summary

- Implemented game environment
- Experimented RL algorithms
- Experimented different feature extractions, reward shaping
- Experimented various training curriculum
- Demo: https://www.youtube.com/watch?v=1LPVosNHaXE

Term 2 Objectives

- Focus on what worked
- AlphaGo-style self play (the proper way)
- Feature Augmentation
  - Frame Stacking
  - Action History
- Online AI Evaluation Platform

Experiments & Results - Overview

- ~~Phase 1: Static agent task~~
- ~~Phase 2: In-game AI~~
- ~~Phase 3: Self play~~
- Phase 4: Proper self play
- Phase 5: Feature Augmentation

Proper self play

- Motivation
  - Inspired by AlphaGo
  - Continuous learning -> more general strategy
  - Avoid catastrophic forgetting
  - Symmetric breaking
- Solution: Opponent sampling
  - Create snapshot agent every K steps
  - Switch opponent every Q steps

Proper self play - Result

- Tested on MLP-DQN on various parameters
- Double 128 - best (K, Q) = (50000, 10)
- Triple 256, the best combination of (K, Q) = (100000, 20)
- At first glance, not much difference?

Target / Network	Static Agent	In-game AI 0	In-game AI 1	In-game AI 2
ACKTR	85	15	55	70
DQN (Double 128)	0	10	75	5
DQN (Double 256)	70	5	80	30
DQN (Triple 256)	20	5	95	40
DQN (Triple 512)	5	5	70	60
DQN (Double 128) – Proper	5	0	85	10
DQN (Triple 256) – Proper	5	0	95	25

Proper self play - Result

Naive self play vs In-game AI 1

- Weird and uninteresting policy

Proper self play vs In-game AI 1

- General playing style
- Diverse skill - tracking, jump kick, tackling
- Aggressive

Proper self play - Result

- Tested on MLP-ACKTR
- Significant improvement
- Most general self play agent

Target / Network	Static Agent	In-game AI 0	In-game AI 1	In-game AI 2
ACKTR	85	15	55	70
DQN (Double 128)	0	10	75	5
DQN (Double 256)	70	5	80	30
DQN (Triple 256)	20	5	95	40
DQN (Triple 512)	5	5	70	60
DQN (Double 128) – Proper	5	0	85	10
DQN (Triple 256) – Proper	5	0	95	25
ACKTR – Proper	70	10	95	85

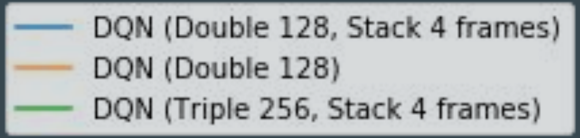
Feature Augmentation

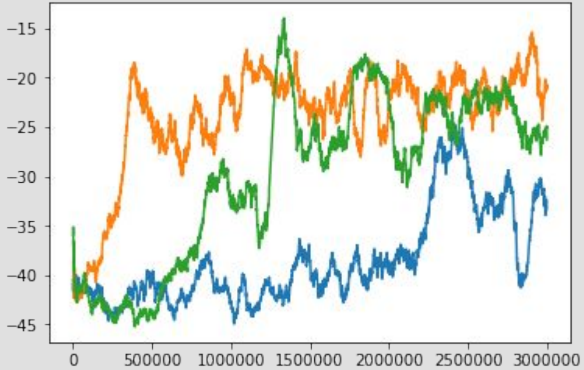
- Frame Stacking
- Action History

Frame Stacking

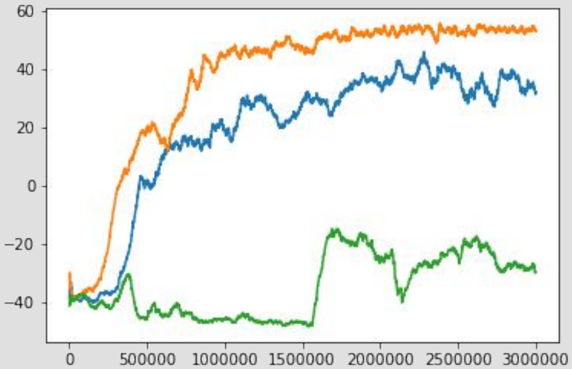
- Motivation
  - Inspired by DQN original paper
  - Capture dynamic information
  - Necessary for some Atari games
- Implementation
  - Environment wrapper
  - Maintain a state deque of size of 4

Frame Stacking - Result & Analysis

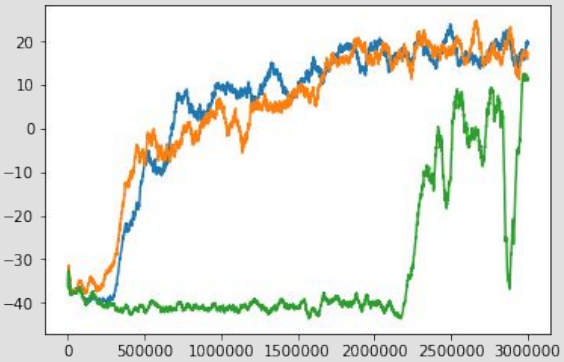




In-game AI 0



In-game AI 1

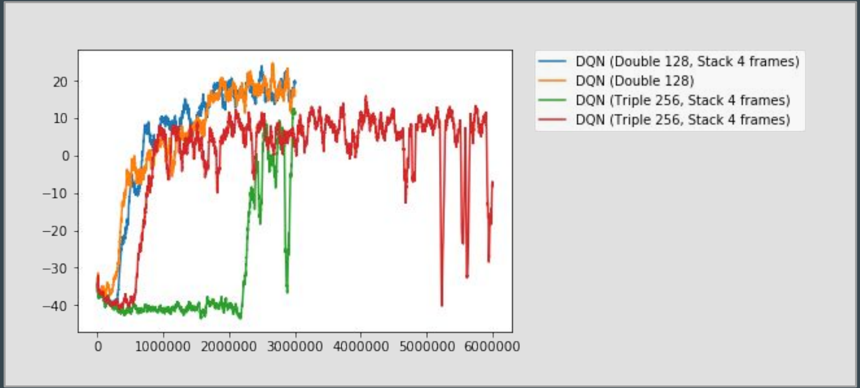


In-game AI 2

- No observable positive effects

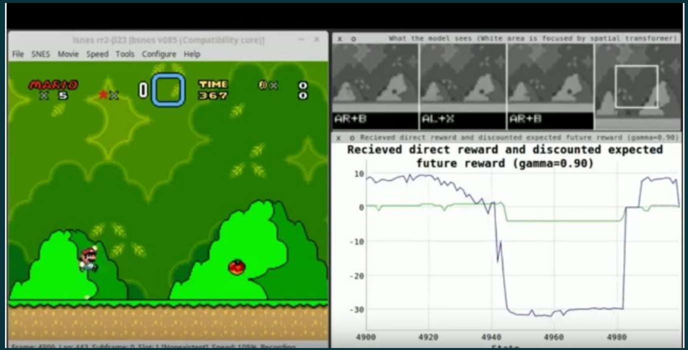
Frame Stacking - Result & Analysis

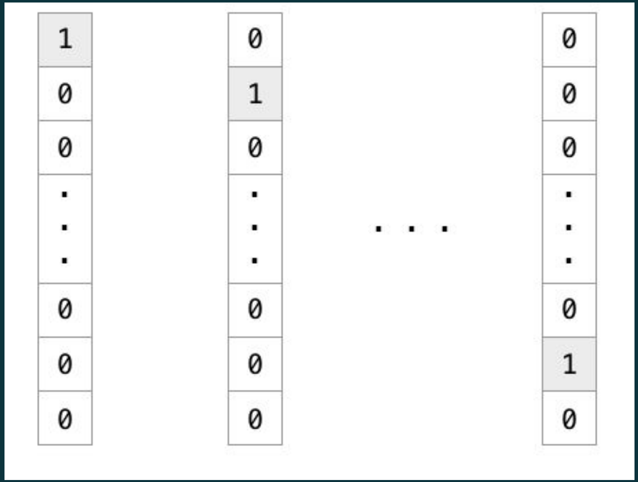
- Information gain is too sparse
- Too much redundancy within frames
- Does not worth 4x dimensionality



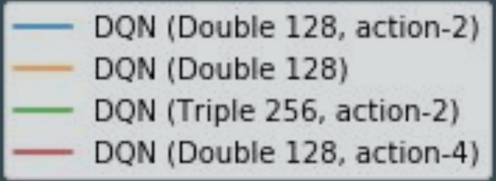
Action History

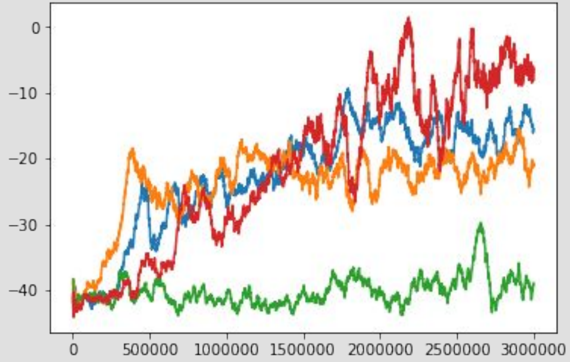
- Motivation
  - Inspired by aleju/mario-ai project
  - Improve action coordination
  - Special attacks discovery
- Implementation
  - Environment wrapper
  - Maintain an action history deque of size of k
  - Append k one-hot vectors into state



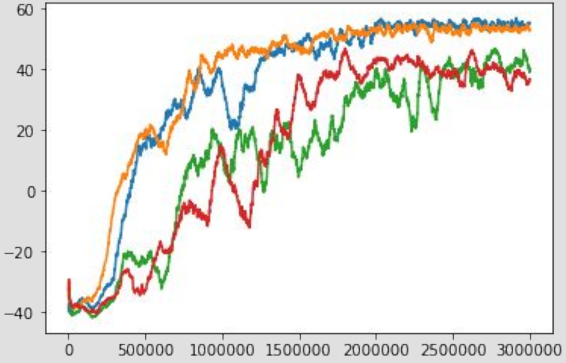


Action History - Result & Analysis

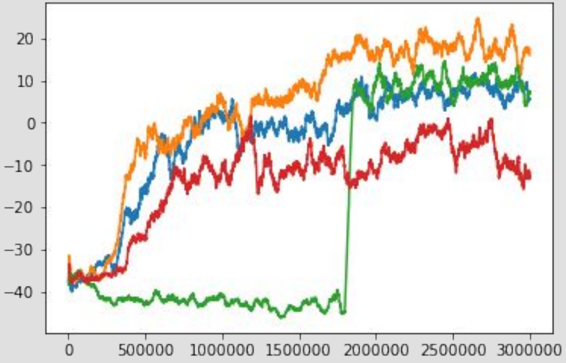




In-game AI 0



In-game AI 1



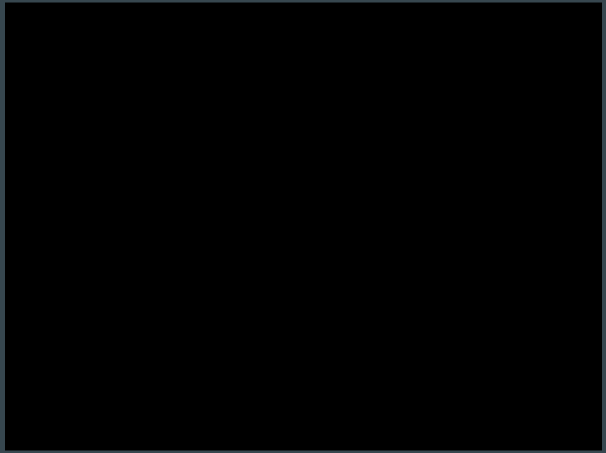
In-game AI 2

- Deeper topology does not help
- Action-2: Better against in-game AI 0, 1
- Action-4: Significantly better in in-game AI 0

Action History - Result & Analysis

Action-2 vs In-game AI 1

- Learned an entirely different policy
- One-Turn-Kill
- Fastest strategy against in-game AI 1



Action-4 vs In-game AI 0

- Fire blast special attack
- Win rate: 50%
- Best DQN agent against in-game AI 0

Action History - Result & Analysis

- Improve action coordination
- Special attacks discovery
- A tradeoff between dimensionality and the above

Online AI evaluation platform

- Motivation
  - Cannot objectively measure AI skills
  - Benchmark with a fixed set of in-game AI led to biased comparison
  - Performance against other RL agents could be unrepresentative
- Idea: Online platform for human to interact with the RL agent
- Key problems
  - Data collection is very expensive
  - Users come and go with various skills

Features

- Accurate rating prediction with sparse data
- Matchmaking
- Concurrent game sessions management
- Error Tolerance
- Low latency
- Informative UI

Trueskill

- A modern rating algorithm
  - Microsoft Research (Cambridge, UK)
  - Bayesian inference
  - Significant improvement over Elo
  - More data efficient
- Applications
  - XBox Live
  - OpenAI Dota AI tournament
- Rating structure
  - The mean skill of the player: μ
  - The degree uncertainty: σ

Technology Stack

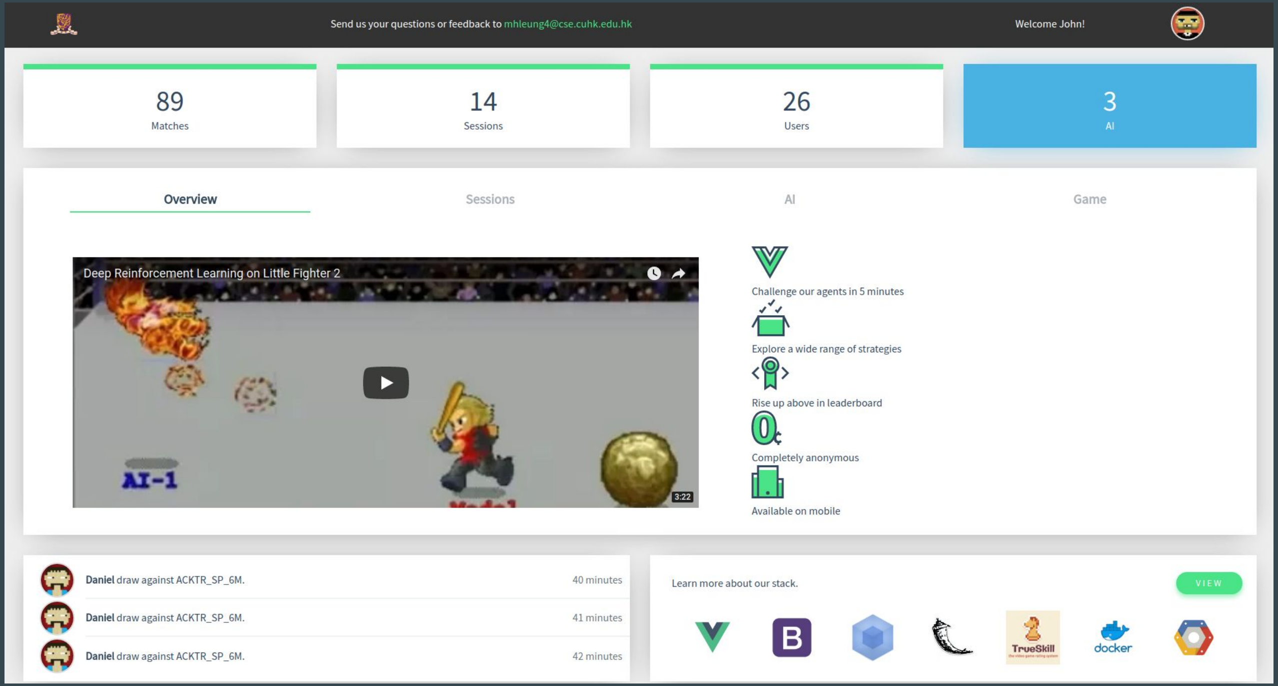
- Frontend
  - Language: ECMAScript 2015 (ES6)
  - Framework: VueJS 2.0
  - CSS Library: Vuestic Admin
  - Module bundler: Webpack
- Backend
  - Language: Python 3
  - Framework: Flask
  - Trueskill API

Deployment

- Google Cloud Platform
  - Zone: Taiwan
  - n1-standard-2
  - 2 Virtual CPUs
  - 7.5GB Memory
  - 30GB SSD Storage
- Docker
  - OS-level virtualization
  - Painless deployment
  - Designed two Docker images

Demo time

- http://104.199.146.210:8080/#/dashboard



Future Work - Diversify play style

- Motivation
  - Agents doesn't use special abilities (except one trained ACKTR agent)
  - No information in features regarding special abilities
  - Limited dynamics
- Ideas
  - Deep Recurrent Q-Network (DRQN)

Future Work - Launching online AI evaluation platform

- Motivation
    - Collect real data
- Milestones:
    - Pilot testing
    - Load test
    - Promotion

Q & A

In-game AI task - Provided targets

In-game AI 0

- Uses all special abilities
- Good at close and long range
- Unfair comparison
- Challenging to mid level player

In-game AI 1

- Move away from target
- Launch jump kicks from angles
- Challenging to mid level player

In-game AI 2

- Mainly close range
- Move back and forth and attack
- Challenging to amatuer level player