MahjongMaster: A General Mahjong Al System



Kuaishou Technology Inc.



Team Introduction

AI Platform & Game Business Department

Empower Games With Al Came promotion: intelligent advertising system Game operation strategies: data-driven gaming matching, novice protection, etc. Game Al agent: GameAl Platform -> Mahjong, FightTheLandlord, etc.

Team Introduction





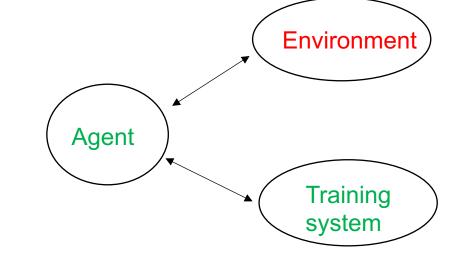
Outline

- Overview of MahjongMaster
 - General Mahjong Al System
 - One-step Decision Making Method
 - Feature & Model Structure
- Training the MahjongMaster
 - Distributed Deep Reinforcement Learning (RL) Framework
 - RL Model Initialization
 - Other Training Techniques

Overview: General Mahjong Al System

General Mahjong AI System

Generalized algorithm for different mahjong rules same training and inference framework same agent design only environment (simulator) is rule-specified Applied in online Mahjong game



serve 6+ mahjong rules with AI rating, surpass top human players

rank 1st in Tournament Round, 3rd in final round, IJCAI2020 Mahjong AI Competition

Overview: One-step Decision Making

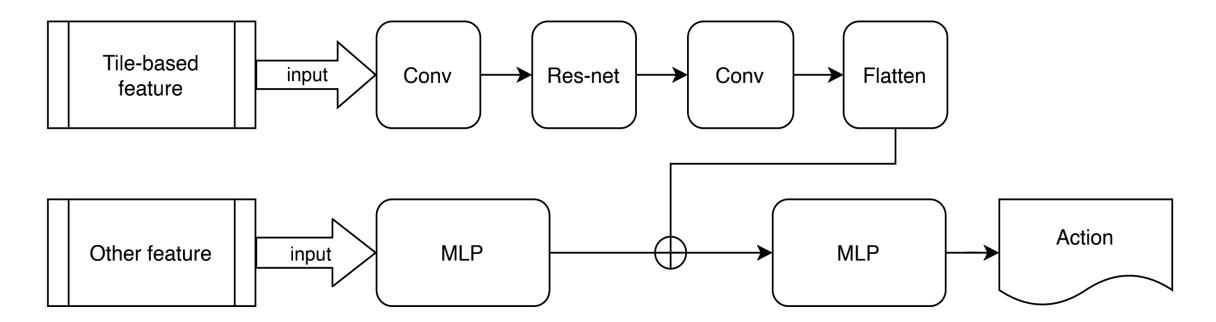
Decision Flow (Suphx) One-step (Ours) Other player Draw a tile discards a tile Winning Model Round over Can Wi Win or not Winning Model Round over Win or not? Win by othe No Generate available Win lavers(AddKong)2 how Mod Can Chov Draw a tile actions and Whether/What to he action is Chow? Discard observations Kong? Can Kong Model: ClosedKong /es→ Make Kong Kong or not Chow AddKop Pong Model Can Pong General Model: ether to Pone? Pong Take the Which action to do? proposed action Kong Riichi Model Declare a Can Riich Kong Model Riichi or noti Riichi State Can Kong ther to Kong Generate available Pass Other player Proposed action actions and errupted by oth discards a tile players? Propose the action observations >=1 Discard Mo #action with the largest Discard a til confidence score Yes Other players take actions, or Other players take actions round ends with no wall tiles left

- Similar to human player
- Rule-specific / Models are specialized

- Suitable for RL training
- Easy to extend
- Low cost for training and inference

Li, Junjie, et al. "Suphx: Mastering Mahjong with Deep Reinforcement Learning." arXiv preprint arXiv:2003.13590 (2020).

Overview: Feature & Model Structure



- Tile-based feature: handcards, tiles set, etc.
- Other features: dealer position, available actions, etc.

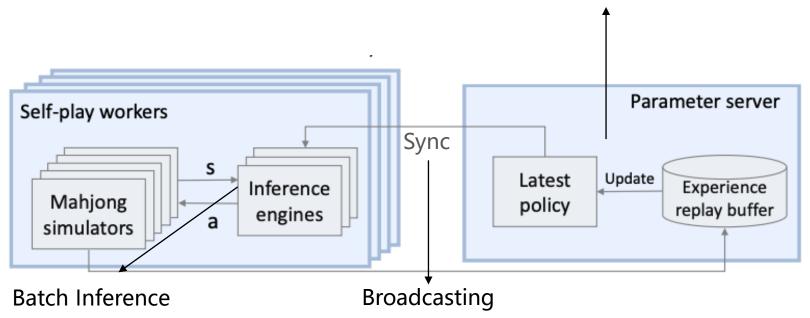
Same model fits well for all rule types, e.g. Chinese-standard, Sichuan, Two-player Mahjong, etc.

Outline

- Overview of MahjongMaster
 - General Mahjong Al System
 - One-step Decision Making Method
 - Feature & Model Structure
- Training the MahjongMaster
 - Distributed Deep Reinforcement Learning (RL) Framework
 - RL Model Initialization
 - Training Techniques

Training: Distributed Deep Reinforcement Learning

- Async training & simulation, deployed on multiple CPU & GPU Machines
- Optimize inference and sync efficiency
- 48 games/second per GPU card (2080ti)
- 20+times faster than Suphx



Support Different RL algorithms

Li, Junjie, et al. "Suphx: Mastering Mahjong with Deep Reinforcement Learning." arXiv preprint arXiv:2003.13590 (2020).

Training: RL Model Initialization

RL from random policy is hard

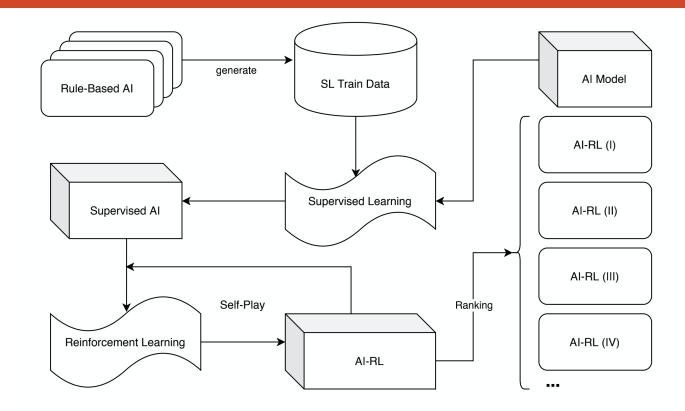
Good initialization is important

Common approach: learn from expert data

Lack of human data: learn from Rule-based AI

Both perform well

Choose suitable approach accordingly



RL based on Rule-base Initialization

Training : Other Techniques

- Reward design
 - Handcrafted reward to avoid sparse reward, useful for early training process
 - adapt the Duplicate Format and use the difference between a player's score as reward, reduce varience
- Entropy control:

$$\nabla_{\theta} J(\pi_{\theta}) = \mathop{\mathrm{E}}_{\substack{s, a \sim \pi_{\theta'} \\ \pi_{\theta'}(s, a)}} \left[\frac{\pi_{\theta}(s, a)}{\pi_{\theta'}(s, a)} \nabla_{\theta} \log \pi_{\theta} \left(a | s \right) A^{\pi_{\theta}}(s, a) \right] + \alpha \nabla_{\theta} H(\pi_{\theta})$$
 after certain time, gradually decay alpha to 0

• Oracle guiding: train an oracle model, then decay to normal model by distillation instead of masking

Thanks!

Contact us: chenzhihan@kuaishou.com