

Chapter 1

General MARL/PPO

1.1 The Surprising Effectiveness of MAPPO in Cooperative, Multi-Agent Games

https://www.researchgate.net/publication/349727671_The_Surprising_Effectiveness_of_MAPPO_in_Cooperative_Multi-Agent_Games

PPO sample efficiency

1 GPU desktop, 1 Multicore CPU for training

centralized value function - global state instead of local observations

environments: Particle world environment

PPO used because seen as sample less efficient, hence for academic purposes
MADDPG and value-decomposed Q-learning

Minimal hyperparameter tuning and no domain specification

Decentralized learning each agent its own policy, suffer from non-stationary transitions

two lines of research - CTDE (this) and value decomposition

in single agent PG advantage normalization is crucial

considered implementation details - input norm, value clipping, orthogonal init, gradient clip - all helpful and included

another - discretization action space for PPO to avoid bad local minima in continuous, layer normalization

MLP vs Recurrent

5 implementation details:

Value norm: running average over value estimates, value network regress to normalized target values (Pop art technique)

Agent-specific global state: concat of all o.i as input to critic
(agent specific global cannot be used in QMix, which uses single mixer network common to all agents)

Training Data Usage: importance sampling to perform off-policy correction ??

multiple epochs may suffer from non stationarity - 15 to 5 epochs (easy to hard)

No mini-batches - more data to estimate gradients - improved practical performance

Action masking: unavailable actions when computing action probabilities
- both forward and backward

Death masking: zero states with agent ID as value input for dead agents

Chapter 2

Overcooked related

https://github.com/HumanCompatibleAI/overcooked_ai

2.1 On the Utility of Learning about Humans for Human-AI Coordination

<https://arxiv.org/abs/1910.05789>

agents assume their partner to be optimal or similar fail to be understood by humans

gains come from having agent adapt to human's gameplay

effective way to tackle two-player games is train agent with set of other AI agents, often past versions

collaboration is fundamentally different from competition (we need to go beyond self-play to account for human behavior)

incorporating human data or models into training leads to significant improvements (behavior cloning model)

Population Based Training is online evolutionary alg, adapts training hyperparameters and perform model selection agents, whose policies are parametrized by NN and trained with DRL alg. each PBT iteration pair of agents are drawn, trained for number of steps and have performance recorded at end of PBT iteration, worst performing agents are replaced with copies of best and parameters mutated

human behavior cloning performed better than with Generative Adversarial Imitation Learning (GAIL)

PBT better than PPO self-play because they are trained to be good with population coordination

Agents designed for humans. Start with one learned human BC as part of environment dynamic and train single agent PPO.

start with ppo self-play and continue with training with human model

planning alg A^*

two human behavior cloning models Hproxy used for evaluation and PPOBC learned against learned human models

best result self-play with self-play

for human interaction was best PPOBC with HProxy...PPOBC is overall preferable

PPOBC outperforms human-human performance

SP agents became very specialized and so suffered from distributional shift when paired with human models

future work - better human models, biasing population based training towards humans

READ AGAIN if interested

2.2 PantheonRL: A MARL Library for Dynamic Training Interactions

<https://iliad.stanford.edu/pdfs/publications/sarkar2022pantheonrl.pdf>

PantheonRL new software package for marl dynamic

Combination of PettingZoo and RLLIB - customization of agents

prioritizes modularity of agent objects - each has separate replay buffer, own learning alg, role

(other powerful DRL library - StableBaselines3)

The modularity of the agent policies combined with the inheritance of StableBaselines3 capabilities together give users a flexible and powerful library for experimenting with complex multiagent interactions

2.3 Investigating Partner Diversification Methods in Cooperative Multi-agent Deep Reinforcement Learning

https://www.rujikorn.com/files/papers/diversity_ICONIP2020.pdf

PBT have diversity problem - PBT agents are not more robust than self-play agents and aren't better with humans

creating diversity by generating pre-trained partners is simple but effective

(partner sampling - playing with uniformly sampled past versions of partner - lacks diversity, past versions have similar behavior)

(population-base training, pre-trained partners)

testing self-play and cross-play with these agent types (SP, SPpast, PBT, PTseeds, PTdiverse)

PTdiverse(hyperparameters) and PTseeds come from self-play

ref: ustesen, N., Torrado, R.R., Bontrager, P., Khalifa, A., Togelius, J., Risi, S.: Illuminating generalization in deep reinforcement learning through procedural level generation. arXiv preprint arXiv:1806.10729 (2018)

IDEA to do: combine pretrained agents with maximum cross entropy? Encorporate maxium cross entropy into ppo??

LOVED THIS ARTICLE for it's simplicity

2.4 Evaluating the Robustness of Collaborative Agents

<https://arxiv.org/abs/2101.05507>

how test robustnes if cannot rely on validation reward metric

unit testing (from software engineering) - edge cases, eg. where soup was cooked but not delivered

incorporating Theory of mind to human model

human modal diversity by using population of human models

state diversity - init from states visited in human-human gameplay

test suite provides significant insight into robustness that is not correlated with average validation reward

”improved robustness as measured by test suite, but decrease in average validation reward”

simple ML metrics are insufficient to capture performance and we must evaluate results base on human judgement

domain randomization - some aspects of env are randomized - behavior can vary significantly based on small randomization of ”irrelevant” factors

Theory of mind??? - each step agent decides what task/strategy to do (eg. deliver soup), then choose low-lever action (motion) to persue this goal.

Population of BC models, ToM models, or mixture of two

recurrent networks for all deep rl training procedures

”once the trained policy has found a good strategy for getting reward, it is not incentivized to explore other areas of the state space”

sampling initial state from human-human data (diverse starts)

Future work - how evaluate robustness in cases of ambiguous behavior

Or evaluating of proposals that populations with BC had positive effect, but negative for ToM

Meta learning for any kind of game layout, not just those prefabricated (online learning)

2.5 Interaction Flexibility in Artificial Agents Teaming with Humans

https://www.researchgate.net/publication/351533529_Interaction_Flexibility_in_Artificial_Agents_Teaming_with_Humans

too psychological, empirical studies of real people experience when playing with self-play / human BC agents

2.6 Maximum Entropy Population-Based Training for Zero-Shot Human-AI Coordination

<https://arxiv.org/abs/2112.11701>

”problem of training a Reinforcement Learning (RL) agent that is collaborative with humans without using any human data”

"To mitigate this distributional shift, we propose Maximum Entropy Population-based training (MEP). In MEP, agents in the population are trained with our derived Population Entropy bonus to promote both pairwise diversity between agents and individual diversity of agents themselves, and a common best agent is"

Comparing MEP with PPO self-play, PBT, Trajectory diversity and Dicitious CO-play

diverse and distinguishable behaviors between all agent pairs utilizes average KL divergence between all agent pairs

each agent in population is rewarded to maximize centralized population entropy.

we train best response agent by pairing it with agent sampled by difficulty to collaborate with (prioritizing)

each agent has maximum entropy bonus (to reward) to encourage policy itself to be exploratory

Population diversity = average entropy of each agent + average KL divergence of pairs

Population entropy = bounded and efficient surrogate for optimization = entropy of mean policies of population

PE is lower bound for PD

Not using PPO, but custom Entropy loss functions

Population entropy (effective linear pairwise kl divergence) as part of objective

MEP shares intuition with domain randomization... (MEP can be seen as domain randomization technique over partners policies)

(TrajeDi = encourages trajectories difference between agents - Jensen-Shannon divergence between policies)

(Diversity-Inducing Policy Gradient = formulated for single agent setting)

Bridges maximum entropy RL and PBT... entropy maximization for achieving robustness

Combining MEP with other MARL algorithms could be Future work

Idea: $r = r + \alpha * \text{population entropy}$?

Idea: ("Maximum entropy approach adds the dense entropy to the reward for each time step, while entropy regularization adds the mean entropy to the surrogate objective") "Note that Entropy regularization is not, in general, equivalent to the maximum entropy objective, which not only optimizes for a policy with maximum entropy, but also optimizes the policy itself to visit states where it has high entropy. Put another way, the maximum entropy objective optimizes the expectation of the entropy with respect to the policy's state distribution, while entropy regularization only optimizes the policy entropy at the states that are visited, without actually trying to modify the policy itself to visit high- entropy state" https://garage.readthedocs.io/en/latest/user/algo_ppo.html

INTERESTING article

2.7 Assisting Unknown Teammates in Unknown Tasks: Ad Hoc Teamwork under Partial Observability

<https://arxiv.org/abs/2201.03538>

Ad hoc teamwork under partial observability (ATPO)

unknown teammates performing unknown task without pre-coordination protocol

ad hoc teamwork has three parts - task identification, teammate identification and planning

What is Zero-shot coordination?? - studies how independently trained agents may interact with another on first-attempt

2.8

2.9 Work progress

7.11.2022

manipulace set agent id pro jednotlivá env, jakým způsobem se zpracovávají odesílané obs a přijímané actions je potřeba toto pak jakkoliv resit na

urovni stable baselines strev? vypada to, ze ne ... Na strane RUNNER: v *obs*[0] je vzdy obs pro self.model a v *obs*[1] je vzdy obs pro self.other agent model Runner vzdy vytvori joint actions jako (self actions, other agent actions) a posle je do env Env je zpracuje, podle agent id budto necha, nebo spravne prohodi do joint actions Env s joint actions provede step, ziska obs a opet spravne dle agent idx budto necha nebo prohodi a vraci (*obs*0, *obs*1), A opet jsme na zacatku ... Runner se muze spolehnout ze v *obs*[0] ma obs toho trenovaneho modelu self.policy a v *obs*[1] ma obs pro other agent model (de facto embedded into environment)

kolik env steps provadi Runner? nelze nastavit pevne jedna epizoda == 400? =i parametr v PPO()

proc overcooked env reset musi resetovat mlam? Zatim zakomentovano, TODO: proverit

learning rate zatim neni annealovany

zakomentovany nektre metriky, kterym zatim nerozumim nebo se mi nemeni (clip fraction, clip range, learning rate)

15.11.2022

Struktura inspiravana projektem max population entropy, take pouzito stable baselines

stable baselines posledni oficialni podpora TF1, neoficialni podpora TF2

nove doporuocene Stable baselines3 ktere pouziva pytorch

S pytorch moc neumim, ale pozmenit reward a pridat do loss rozdil oproti populaci asi neni problem, takze zalezi jak moc velke zmeny ocekavam

Adaptace na SB3 docela jednoduchou, vektorizovane prostredi

Other agent jako soucast "single agent" prostredi

Technicke problemy:

Prostredi se vzdy resetuje do stejneho stavu

Struktura CNN, MLP vs reprezentace stavu lossless, featurize state mdp

RNN, Frame stacking, nebo staci reakni agent?

Jak vyhodnocovat agenty deterministicky arg max dopadne kazda epizoda naprosto stejne, nedeterministicky teoreticky nedostavam nejlepsi vysledek, nebo vyhodnocovany agent deterministicky a other agent nedeterministicky? Zatim nevim jak to resi v ostatnich projektech

Annealing entropy coeficient?, annealing sparse r coef

Napevno 5M env steps staci? Da se o tom rict "with little loss of generality"? Nebo tohle muze byt zkreslujici pro vysledek mych experimentu?

5 zajimavych map

diff bonus: k dispozici $\log_p \text{rob}$ pro danou akci, $e^{\log_p \text{rob}} = \text{prob}$, $\text{diff} = \min_p \text{op}(p(a) - a(a))^2 = (\text{mean}(\text{pop}(a) - a(a)))^2$

Konkretni plan: Zacit s jednou mapou, natrenovat si 10 self-play agentu (pripadne dalsich 10 s ruznymi hyper-parameters), ktere si necham pro nezávisle testovani, Pak zacit trenovat tim pridavacim zpusobem populaci a kazdeho noveho jedince otestovat vuci vsem 10 testovacim. Teorie je takova ze s kazdym pozdeji pridany agentem ma byt vysledek lepsi vuci test agentum. Zacit s diff bonusem = 0 pro porovnani a pak zkouset bonus zvysovat (0.1). To same pro dalsi mapu a zkosit najit nejakou zajimavou hodnotu pro diff