Project Thesis
in Computer Science

Multi-Agent Reinforcement Learning with StartCraftII

Patrick Matthäi

Aufgabensteller:	Prof. Dr. Matthias Schubert
Betreuer:	Sabrina Friedl
Abgabedatum:	99.99.9999

Declaration of Authorship

I hereby declare that the thesis submitted is my own unaided work. All direct or indirect sources used are acknowledged as references.
This paper was not previously presented to another examination board and has not been published.

München, 99.99.9999

.....
Patrick Matthäi

Abstract

Dieses Dokument dient als Muster für die Ausarbeitung einer Project Thesis am Lehrstuhl für Datenbanksysteme am Institut für Informatik der LMU München. Der Abstract sollte nicht mehr als 300 Wörter enthalten.

Contents

Chapter 1

Introduction

Chapter 2

Presets

2.1 Introduction to StarCraftII Reinforcement Learning

2.1.1 Game mechanics

2.1.2 SMAC - A StarCraftII multi-agent environment

2.1.3 PyMARL - A multi-agent reinforcement learning algorithm framework

2.2 Reinforcement Learning Basics

2.2.1 MDP - Markov Decision Process

2.2.2 Dec-POMDP - Decentralized Partially Observable MDP

A Concise Introduction to Decentralized POMDPs.
Optimal and approximate q-value functions for decentralized pomdps

2.3 Basic Reinforcement Learning Algorithms

2.3.1 Q-Learning

helps with learning without the explicit information about the dynamics of the environment or the rewards. allows to estimate the value or quality of an action in a particular state of the environment.

$$Q_{i+1}(s_t, a_t) = (1 - \eta_t)Q_i(s_t, a_t) + \eta_t(r_t + \gamma max_a Q_i(s_t, a_t))$$
$$\eta_t \in (0, 1) \text{ is the learning rate}$$

Learning from delayed rewards

2.3.2 DQN - Deep Q-Network

Model free, approximate Q-value function with CNN

Squared Temporal Difference (TD) Error: $\mathcal{L}(\theta) = \sum_{i=1}^{b}[(y_i^{DQN} - Q(s, u; \theta))^2]$

$$y_i^{DQN} = r + \gamma max_{u'} Q(s', u'; \theta^-)$$

where θ^- are parameters from a target network copying θ periodically .

Reinforcement learning for robots using neural networks.
Playing atari with deep reinforcement learning.
Human-level control through deep reinforcement learning. Nature,
Prioritized experience replay

2.3.3 Actor-Critic

2.3.4 A3C - Asynchronous Actor-Critic Agents

2.3.5 BPTT - Truncated back-propagation through time

2.4 Problems in MARL

2.4.1 Credit Assignment Problem

Because the TD error considers only global rewards, the gradient computed for each actor does not explicitly reason about how that particular agents actions contribute to that global reward. Since the other agents may be exploring, the gradient for that agent becomes very noisy, particularly when there are many agents. -COMA

2.4.2 Stability of experience replay

2.4.3 Non-stationarity of the environment

Changing policies of agents affect each others learning-¿ no convergence exploration,

Agents are part of other agents environment - MACKRL Tan

2.4.4 Centralized learning of decentralised policies

decentralised policies = executed independently = policy conditions on agents own action-observation history

learned centralised (work (Rashid et al., 2018; Foerster et al., 22016, 2017, 2018; Kraemer Banerjee, 2016; Jorge et al., 2016),)

2.4.5 Lazy agent problem

TODO - "The distributed nature of the learning offers new benefits but also challenges such as the definition of good learning goals or the convergence and consistency of algorithms [Sch14, BBDS08]" IQL

- "multiagent case ,environment state transitions and rewards are affected by joint action of all agents. -¿ value of an agent?s action depends on actions of others -¿ each agent must keep track of each of other learning agents, possibly resulting in an ever-moving target. In general, learning in the presence of other agents requires a delicate trade-off between the stability and adaptive behavior of each agent." IQL

-"decentralised poli- cies, which severely limit the agents? ability to coordinate their behaviour."-MACKRL

"Often agents are forced to ignore information in their individual observations that would in principle be useful for maximising reward, because acting on it would make their behaviour less predictable to their teammates. This limitation is particularly salient in IL, which cannot solve many coordination tasks (Claus Boutilier, 1998)."

2.5 Motivation for multi-agent reinforcement learning

single agents typically fare poorly on such tasks, since the joint action space of the agents grows exponentially with the number of agents. - COMA

Chapter 3

Related work on MARL

3.1 JAL - Joint Action Learning

Claus Boutilier 1998 - MACKRL

3.2 IL - Independent Learning

Tan 1993 - MACKRL

3.3 IQL - Independent Q-Learning

Two agents controlled by independent Deep Q-Networks as described in [MKS+15], "simplest method consists of using an autonomous Q-learning algorithm for each agent in the environment, thereby using the environment as the sole source of interaction between agents."

3.4 VDN - Value-decomposition Networks

The main idea behind VDN arises from the assumption that joint-action-value functions can be decomposed into d single-agent value functions which are then composed via summation into the join-action-value function.

$$Q((h^1, h^2, ..., h^d), (a^1, a^2, ..., a^d)) \approx \sum_{i=1}^{d} \tilde{Q}_i(h^i, a^i)$$

Thus each agents value-function $\tilde{Q}_i$ depends solely on it's local observation. The history of agent i is denoted as h^i while a^i corresponds to the agent's current action. The history $h_t = a_1 o_1 r_1, ..., a_{t-1} o_{t-1}, r_{t-1}$ of an agent includes

it‘s action, observation and reward at a given timestep up until a the last registered timestep $t-1$.

The agent‘s value-function is learned via back-propagation of the gradients resulting from applying the Q-learning rule on the joint reward and the local observations. $\tilde{Q}_i$ is therefore independent of specific rewards.

As a result each agent independently performs greedy actions based on it‘s own $\tilde{Q}_i$.

3.5 QMIX - Monotonic Value Function Factorisation

The basic idea behind QMIX results from the insight that full factorisation as performed in VDN can be replaced with a constraint on the joint-action-value function:

$$argmax_u Q() = \begin{pmatrix} argmax_u^1 Q_1() \\ \vdots \\ argmax_u^n Q_n() \end{pmatrix}$$

TODO: explain constraint

Further monotonicity is required due the constraint:

$$\frac{\partial Q}{\partial Q_a} \geq 0, \forall a \in A$$

In order to fulfill the above requirements, various agents networks as well as a hypernetwork and mixing network are deployed. Each agent network is comprised as a DRQN representing its individual value function Q_a. This network is provided with the current observation o_t^a and the last action u_{t-1}^a of agent a at each timestep.

The mixing network receives the agent network outputs and returns Q by mixing the outputs monotonically. The monotonicity contraint is enforced via non-negative weights in the mixing layer.

The weights (of one layer??) in return are produced by hypernetworks which are supplied with the state. This networks consist of a single linear layer and a absolute function (non-negative weights). Biases are independently generated.

3.6 COMA - Counterfactual Multi-Agent Policy Gradients

actor-critic policy-gradient method, decentralised policies, centralised critic to estimate Q-function and decentralised actors to optimise agents policies.

counterfactual baseline marginalises single agents action while keeping other agents actions fixed.

3.7 Central-V

3.8 ASN - Action Semantic Networks

3.9 MACKRL - Multi-Agent Common Knowledge Reinforcement Learning

"stochastic actor-critic algorithm that learns a hierarchical policy tree."

higher tree levels = groups lower levels = subgroups (richer common knowledge) lowest level = independently learnt decentralised policies

"centralised training of decentralised policies. During learning the agents can share observations, parameters, gradients, etc. without restriction but the result of learning is a set of decentralised policies such that each agent can select actions based only on its individual observations"

3.9.1 Common Knowledge Concept

Agents share common knowledge as soon as the see each other defined on their field of view.

Without common knowledge, decentralised coordination of agents has to fall back on implicit communication. This includes observation of actions and their resulting effects in the environment and other agents.

problems of implicit: "typically require multiple timesteps to execute, can limit the agility of control during execution (Tian et al., 2018)"

Bibliography

[1] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure. In *Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'99), Philadelphia, PA*, pages 49–60, 1999.

[2] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD'96), Portland, OR*, pages 226–231, 1996.