DIAG
Dipartimento di Ingegneria
informatica, automatica e gestionale
Antonio Ruberti

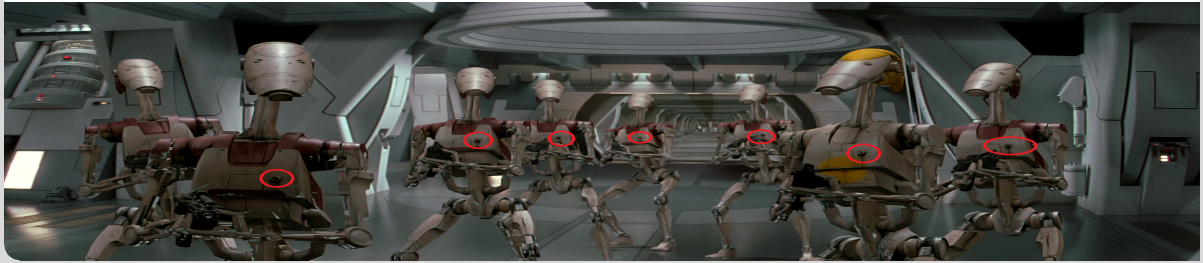
SAPIENZA
UNIVERSITÀ DI ROMA

Restraining Bolts for Deep Reinforcement Learning

`https://github.com/soto323/Restraining-Bolts-for-Reinforcement-Learning`

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt | May 26, 2020

REASONING AGENTS, 2019/2020 FINAL PROJECT, PROF. GIUSEPPE DE GIACOMO



Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

Outline

Markov Decision Processes - MDPs

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

Definition

- A model for sequential decision making.
- $\mathcal{M} = \langle S, A, Tr, R \rangle$ contains
  - a set of states S (world features).
  - a set A of actions.
  - a transition function $Tr : S \times A \longrightarrow Prob(S)$ that returns for every state s and action a a distribution over the next state.
  - a reward function $R : S \times A \times S \longrightarrow R$ that specifies the reward (a real value) received by the agent when transitioning from state s to state $s^{'}$ by applying action an a.

Markov Decision Processes - MDPs

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

Usage

- To model and solve many real-world problems, and to address the problem of learning to behave well in in unknown environments.
- A solution to an MDP is a function, called a policy.
  - Assigns an action to each state, possibly conditioned on past states and actions.
  - The value of a policy ρ at state s, denoted $v^{\rho(s)}$, is the expected sum of (possibly discounted by a factor γ, with $0 \leq \gamma < 1$) rewards when starting at state s and selecting actions based on ρ.

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

Reinforcement Learning

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

Definition

- RL is the task of learning a possibly optimal policy in an MDP.
- From an initial state s_0, and where only S and A are known.
- Tr and R are unknown.

Usage

- Typically, the MDP is assumed to be at s_0, so policy optimality is evaluated w.r.t. $v^{\rho(s)}$.
- Every MDP has an optimal policy p^*.
- In discounted cumulative settings, there exists an optimal policy that is Markovian $\rho : S \longrightarrow A$, i.e.,$\rho$ depends only on the current state, and deterministic (Puterman 1994).

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

LTL_f and LDL_f

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

LTL_f – Linear Temporal Logic on finite traces

- A sequence of propositional interpretations(De Giacomo and Vardi 2013).
- Given a set $\mathcal{P}$ of boolean propositions, here called *fluents* (Reiter 2001), LTL_f formulas φ are defined as follows:

$$\varphi ::= \phi | \neg\varphi | \varphi_1 \wedge \varphi_2 | \circ\varphi | \varphi_1 \mathcal{U} \varphi_2 \tag{1}$$

Examples

$$\varphi ::= A | \neg\varphi | \varphi_1 \wedge \varphi_2 | next\,\varphi | eventually\,\varphi | always\,\varphi | \varphi_1\, until\, \varphi_2 \tag{2}$$

This logic allows us to represent i.e. reachability, safety, reactiveness, until, precedence.

LTL$_f$ and *LDL$_f$*

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

LDL$_f$ – Linear Dynamic Logic on finite traces

- Proper extension of *LTL$_f$* which is as expressive as monadic second-order logic (MSO) over finite traces.
- Allows for expressing regular expressions over such sequences, hence mixing procedural and declarative specifications.
- Formally, *LDL$_f$* formulas φ are built as follows:

$$\begin{aligned} \varphi &::= tt \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \langle \varrho \rangle \varphi \\ \varrho &::= \phi \mid \varphi? \mid \varrho_1 + \varrho_1 \mid \varrho_1 ; \varrho_2 \mid \varrho_1^* \end{aligned} \tag{3}$$

Exmaples

Addressing tasks such as satisfiability, model checking, reactive synthesis and planning under/full partial observability (De Giacomo and Vardi 2013).

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

NMRDPs Introduction

Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

Motivation

The Markovian property makes it difficult to reward more interesting long-term behaviours.

Previous work

- (Bacchus, Boutilier, and Grove 1996) evaluate the entire sequence of the state leading to that state to check whether a past temporal formula has been satisfied. Example: Now I have a ticket to Rome, so the ”book a ticket” command was issued in the past.
- (Thiebaux et al. 2006) uses a temporal logic of the future to reward states that satisfies formulas. At every steps it checks whether the conditions is satisfied.

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

NMRDPs definition

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

Definition

A NMRDP is a tuple $\mathcal{M} =< S, A, Tr, R >$ where S, A, Tr are as MDP while R is redefined as $R : (S \times A)^* \rightarrow \mathbb{R}$ i.e. as a real-valued function over finite state-action sequences (*traces*).

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

NMRDPs Value function

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

Value function in NMRDPs

Given a (possibly infinite) trace π, the value of π is :

$$\upsilon(\pi) = \sum_{i=1}^{|\pi|} \gamma^{i-1} R(\langle \pi(1), \pi(2), \cdots, \pi(i) \rangle),$$

where $0 < \gamma \leq 1$ is the discount factor and $\pi(i)$ denotes the pair (s_{i-1}, a_i).

Value of a policy in NMRDPs

Given an initial state s_0 and a policy ρ defined as $\rho : S^* \rightarrow A$, we can define the value of a policy ρ as expected value of infinite trace:

$$\upsilon^{\rho}(s) = E_{\pi \sim \mathcal{M}, \rho, s_0} \upsilon(\pi)$$

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

Non-Markovian rewards using temporal logic

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

Designing reward functions

Designing reward functions is usually difficult, even if just considering finite traces. We can exploit intuitiveness of LTL_f/LDL_f to design the reward function R implicitly using a set of pairs $(\varphi_i, r_i)_{i=1}^{m}$.

Transforming LTL_f/LDL_f formula into a DFA

We can associate each LDL_f formula to a NFA $A_\varphi = \langle 2^{\mathcal{P}}, Q, q_0, \delta, F \rangle$ that accepts exactly the traces satisfying φ using Algorithm of (Brafman, Giacomo, and Patrizi 2018) to transform LDL_f formula into a NFA. Then, we can easily transform the NFA into a DFA A_φ by paying an exponential cost.

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

Building an Equivalent Markovian Model of NMRDP

Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

(Bacchus, Boutilier, and Grove 1996) theorem

An NMRDP $\mathcal{M} = \langle S, A, Tr, R \rangle$ is equivalent to an extended MDP $\mathcal{M}' = \langle S', A, Tr', R' \rangle$ if there exist two functions $\tau : S' \to S$ and $\sigma : S \to S'$ such that:

- $\forall s \in S : \tau(\sigma(s)) = s$
- $\forall s_1, s_2 \in S$ and $s'_1 \in S'$: if $Tr(s_1, a, s_2) > 0$ and $\tau(s'_1) = s_1$, there exists a unique $s'_2 \in S'$ such that $\tau(s'_2) = s_2$ and $Tr(s'_1, a, s'_2) = Tr(s_1, a, s_2)$
- For any feasible trajectory $\langle s_0, a_1, \cdots, s_{n-1}, a_n \rangle$ of $\mathcal{M}$ and $\langle s'_0, a_1, \cdots, s'_{n-1}, a_n \rangle$ of $\mathcal{M}'$ such that $\tau(s'_i) = s_i$ and $\sigma(s_0) = s'_0$, we have $R(\langle s_0, a_1, \cdots, s_{n-1}, a_n \rangle) = R'(\langle s'_0, a_1, \cdots, s'_{n-1}, a_n \rangle)$

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

Building an Equivalent Markovian Model of NMRDP

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

NMRDP to MDP with LTL_f/LDL_f formula rewards

We can now build a MDP $\mathcal{M}' =< S', A, Tr', R' >$ that uses rewards coming from satisfaction of temporal formulas equivalent to the previous NMRDP $\mathcal{M} =< S, A, Tr, R >$.

Equivalent MDP definition

Using the algorithm of De Giacomo et al. 2018, for each reward formula φ_i we build the corresponding (minimal) DFA $A_\varphi = \langle 2^{\mathcal{P}}, Q_i, q_{i0}, \delta_i, F_i \rangle$. The components of the seeked MDP $\mathcal{M}' =< S', A, Tr', R' >$ are the following:

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

Building an Equivalent Markovian Model of NMRDP

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

...

- $S' = Q_1 \times \cdots \times Q_m \times S$
- $A' = A$
- $Tr' : S' \times A' \times S' \to [0, 1]$ defined as:

$$Tr'(q_1, \cdots, q_m, s, a, q'_1, \cdots, q'_m, s') = \begin{cases} Tr(s, a, s') & if \forall i : \delta_i(q_i, s) = q'_i \\ 0 & otherwise \end{cases}$$

- $R' : S' \times A \to \mathbb{R}$ defined as:

$$R(q_1, \cdots, q_m, s, a) = \sum_{i:\delta_i(q_i,s)\in F_i} r_i$$

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

Equivalent MDP of NMRDP

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

Theorem Brafman, Giacomo, and Patrizi 2018

The NMRDP $\mathcal{M} =< S, A, Tr, R >$ is equivalent to the extended MDP $\mathcal{M}' =< S', A, Tr', R' >$

Theorem (Bacchus, Boutilier, and Grove 1996)

Given an NMRDP $\mathcal{M}$, let ρ' be an optimal policy for an equivalent MDP $\mathcal{M}'$. Then, policy ρ for $\mathcal{M}$ that is equivalent to ρ' is optimal for $\mathcal{M}$.

Reinforcement Learning in NMRDPs

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

Reinforcement Learning setting and objective

Given the NMRDP $\mathcal{M} =< S, A, Tr, (\varphi_i, r_i)_{i=1}^m >$ with Tr and $(\varphi_i, r_i)_{i=1}^m$ unknown to the agent but sampled during the learning, the RL problem consists of learning an optimal policy $\hat{\rho}$.

Theorem

Reinforcement Learning for LTL_f/LDL_f rewards over an NMRDP $\mathcal{M} =< S, A, Tr, (\varphi_i, r_i)_{i=1}^m >$, with Tr and $(\varphi_i, r_i)_{i=1}^m$ hidden to the learning agent can be reduced to RL over the MDP $\mathcal{M}' =< S', A, Tr', R' >$ defined above, with Tr' and R' hidden to the learning agent.

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

Restraining Bolts

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

Definition

Restraining Bolts are devices inspired by science fiction whose goal is to limit an agent's behaviour to a desired set of actions.

Desired Actions

The desired set of actions consists of a logical specification of traces, expressed in LTLf and LDLf. The formulae are specified over a set of logical propositions $\mathcal{P}$.

Rewards

Whenever a trace is desirable, the restraining bolt gives extra reward to the agent.

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

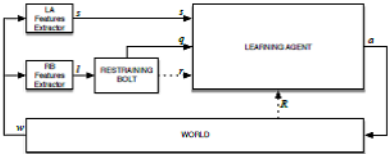
Structure of the Restraining Bolt

Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

Feature extraction

- There is assumed to be a feature extractor for the agent and one for the restraining bolt that are tasked with converting the world state into the traces for each. $w \rightarrow l$ and $w \rightarrow s$ respectively.
- The features generated are potentially different from each other and are only correlated by the fact that they represent different aspects of the same world.

Figure: Learning Agent and RB



Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

Structure of the Restraining Bolt

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

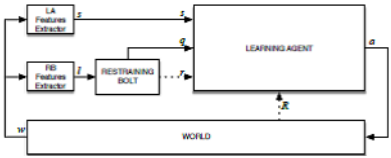
Agent's Representation

The agent is modelled by the MDP $\mathcal{M}_{ag} = \langle S, A, Tr_{ag}, R_{ag} \rangle$

RB's Representation

- The restraining bolt is specified as $RB = \langle \mathcal{L}, (\varphi_i, r_i)_{i=1}^{m} \rangle$
- To model the level of satisfaction of the goal, the minimum DFA $\mathcal{A}_{\varphi_i} = \langle 2^{\mathcal{P}}, Q_i, q_{i0}, \delta_i, F_i \rangle$ corresponding to the formula φ_i is used.

Figure: Learning Agent and RB



Structure of the Restraining Bolt

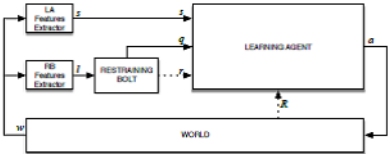
Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

Connection Between the Restraining Bolt and Learning Agent

- Rewards
- State of satisfaction of the goal.

Figure: Learning Agent and RB



Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

Reward Machines

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

Definition

Reward machines are automata-based representations that expose the reward function structure to the agent.

Rewards

The main intuition is to assign a reward to a goal specified in any formal language, easing the burden of complex reward function specification and learning from sparse rewards.

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

Structure of Reward Machines

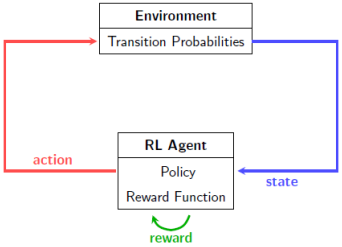
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

Labelling Function

It is assumed to be a labelling function that maps the world states to a set of truth assignments over a set of propositional symbols $\mathcal{P}$.

$$L : S \times A \times S \rightarrow 2^{\mathcal{P}} \tag{4}$$

Figure: Reward Machine



Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

Structure of Reward Machines

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

Reward Machine's Representation

Formally, a reward machine is a Mealy machine. A Mealy machine is a tuple $\mathcal{RM} = \langle Q, q_0, \Sigma, \mathcal{R}, \delta, \rho \rangle$. Where

- Q is a finite set of states
- $q_0 \in Q$ is the initial state.
- Σ is the finite input alphabet
- $\mathcal{R}$ is the finite output alphabet
- $\delta : Q \times \Sigma \to Q$ is the transition function
- $\rho : Q \times \Sigma \to \mathcal{R}$ is the output function

Connection to Agent

The reward machine is seen as a part of the learning agent.

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

Restraining Bolt vs Reward Machine

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

Sources

- Restraining Bolt: Giacomo et al. 2019
- Reward Machine: Camacho et al. 2019 (and others)

Comparison

Property	Restraining Bolt	Reward Machine
Definition	Logical specification of traces that are desirable, expressed in LTLf and LDLf.	Automata-based representation the exposes the reward structure to the agent.
Aim	Limit the agent's actions to a set of desired behaviors, avoiding the explicit specification of non-Markovian reward functions	Ease the burden of specifying complex reward functions and learning from sparse rewards.
Transition Systems	The agent is represented by an MDP that sees some features of the state of the world and rewards. In addition it also receives the rewards and the level of satisfaction of the goal from the RB.The RB is represented by the minimal DFA that corresponds to the goal.	The RM is considered to be part of the agent, which is now an MDPRM. However, the agent is now made up of two transition systems: and MDP and a Mealy machine. The MDP no longer receives rewards from the environment, only the reward machine provides it with rewards.
Expressive Power	The expressive power of LTLf is the same as FOL, and the expressive power of LDLf is the same as MSOL.	The expressive power is the same as the language reconized by DFA, which is equivalent to that of regular language, i.e. MSOL.

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

Implementation

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

Tools

- Python 3.6
- Tensorflow 1.x
- Open AI gym wrapper – Minimalistic Gridworld Environment. (MiniGrid) Chevalier-Boisvert, Willems, and Pal 2018

Code

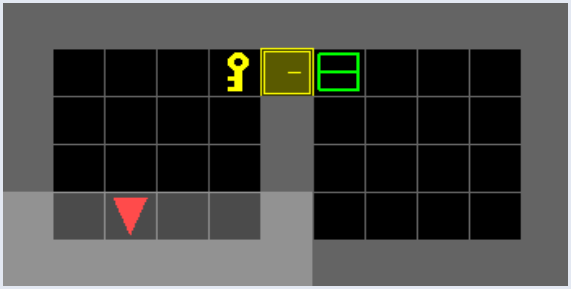
- All code for training and testing including our models is available on `https://github.com/soto323/Restraining-Bolts-for-Reinforcement-Learning/`

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

Environment and Algorithms

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

Enviroment: MiniGrid-Unlock-v0



Algorthims

- DQN
- DDQN
- Actor Critic (a2c)

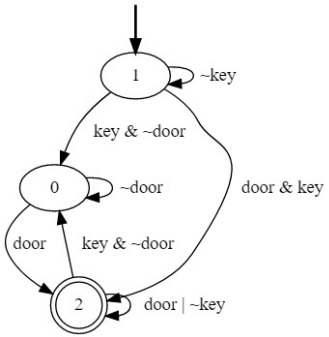
Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

Our Temporally Extended Goal

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

LTL_f formula

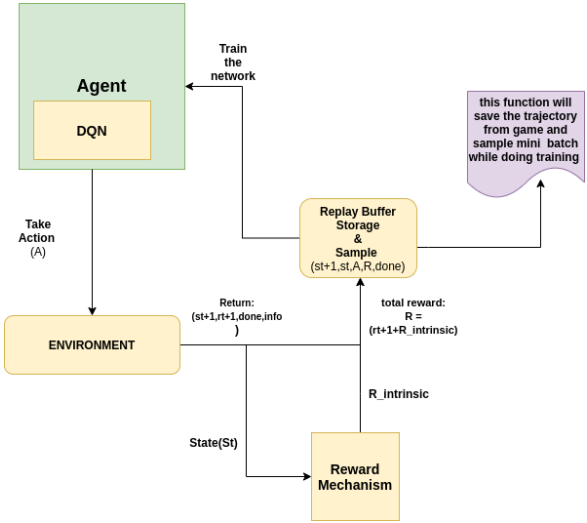
$$\varphi = \Diamond K \wedge \Box(K \implies \Diamond D) \qquad (5)$$



Pipeline

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti



Results – Tensorboard (rewards)

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

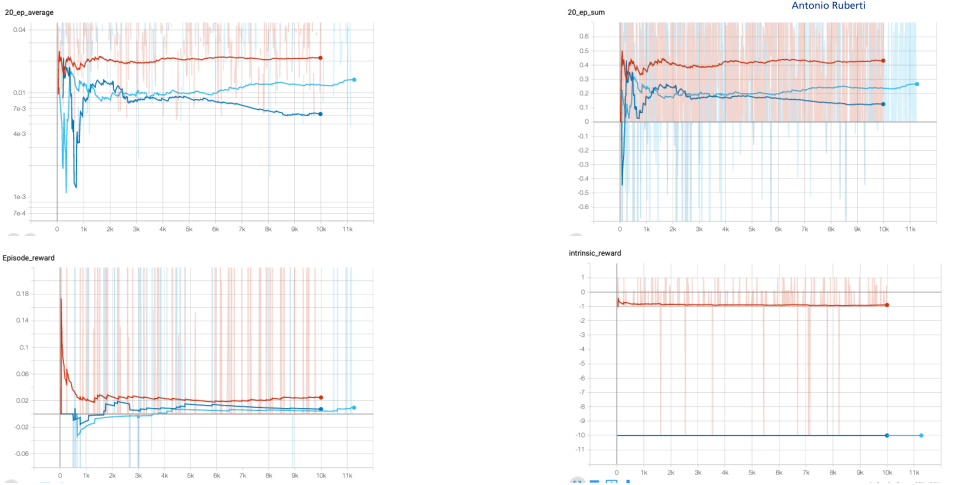
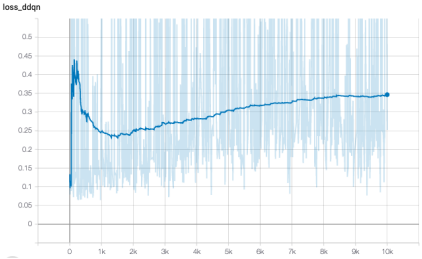


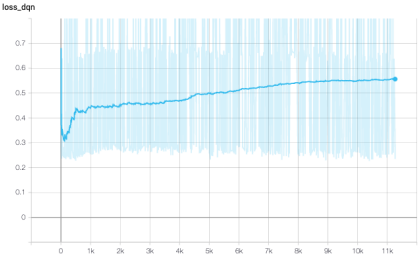
Figure: Orange – Actor Critic. Dark Blue – DDQN. Light Blue – DQN.

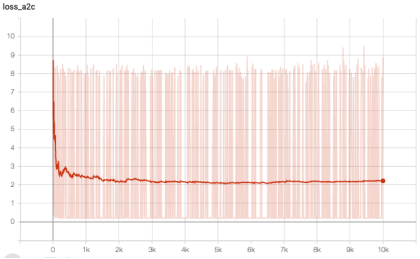
Results — Tensorboard (network losses)

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti





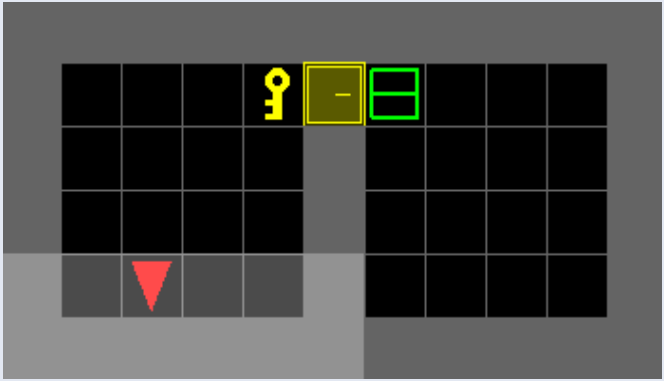


Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

Gameplay after training

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

Enviroment: MiniGrid-Unlock-v0



Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

Conclusions

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

Conclusions: paper

- We can define an agent that learns a Non-Markovian policy while still using classical Reinforcement learning algorithms that work on MDPS.
- The non-markovian goals are defined using formal langauges (LTL/f, LDL/f) and they're represented with DFA's.
- Reward Engineering is supposed to be eased, however despite being easier to define it's hard to find the appropriate value for those rewards to make the agent learn faster and according to established restrictions.

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

Conclusions

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

Conclusions: improvements

- Using algorithms such as DQN, DDQN, and Actor Critic did not prove sufficient. Some of the possible improvements would be:
  - Tuning the way we pass the rewards $\rightarrow$ reward system becomes another hyperparemter. Potentially we could use population based training to find the optimal values since the network is relatively lightweight ($<$ 300 mb of memory usage on GPU).
  - Hyperparameter tuning for the network itself as well.
  - More training time, Deeper network
  - More complex algorithms i.e. PPO

Anupam Nautiyal, Federico Vergallo, Esteban Soto, Michal Ostyk-Narbutt

References I

DIAG
Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

- Ronen I. Brafman, Giuseppe De Giacomo, and Fabio Patrizi. „LTLf/LDLf Non-Markovian Rewards“. In: *AAAI*. 2018.
- Alberto Camacho et al. „LTL and Beyond: Formal Languages for Reward Function Specification in Reinforcement Learning“. In: Aug. 2019, pp. 6065–6073. DOI: 10.24963/ijcai.2019/840.
- Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. *Minimalistic Gridworld Environment for OpenAI Gym*. https://github.com/maximecb/gym-minigrid. 2018.
- Giuseppe De Giacomo et al. *Reinforcement Learning for LTLf/LDLf Goals*. July 2018.
- Giuseppe De Giacomo et al. „Foundations for Restraining Bolts: Reinforcement Learning with LTLf/LDLf Restraining Specifications“. In: *ICAPS*. 2019.