Crawling in Rogue's dungeons with (partitioned) A3C

Andrea Asperti, Daniele Cortesi, Francesco Sovrano

University of Bologna
Department of Informatics: Science and Engineering (DISI)

Fourth International Conference on
Machine Learning, Optimization, and Data Science
September 13-16, 2018 Volterra, Tuscany, Italy

Learning to play Rogue through Reinforcement Learning

Rogue: a famous video games of the ’80, the ancestor of this gender.

The player (the rogue) must retrieve the amulet of Yendor inside a dungeon composed of many levels, collecting objects and fighting enemies.

We exclusively focus on **roaming** inside the dungeon: find the stairs and take them to descend to the next level.

Why games

Game-like environments, providing abstractions of real-life situations, have been at the core of many recent breakthroughs in Deep RL (mostly by Deep Mind):

- Atari Games: DQN [4], A3C [3] (Mnih et al.)
- Sokoban: Imagination augmentation [6] (Weber et al.)
- Labyrinth: ACER [5] (Wang et al.)

Mazes and labyrinths are a traditional topic of reinforcement learning, often requiring **memory**, **attention**, and the acquisition of complex, non-reactive behaviors based on long-term **planning**.

Why Rogue

Rogue has many challenging features for Deep RL:

- **no level replay**: dungeons are randomly generated and always different from each other
- **partially observable** (POMD): the map gets discovered during exploration
- **sparse rewards**

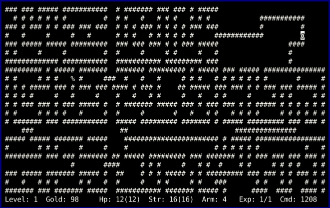
The ASCII interface allows us to focus on the really challenging aspects of the game, bypassing image detection problems (by now well understood).

Rogueinbox

In previous works [2, 1] we developed an API for Rogue, easing the development of automatic agents; the library was tested of many architectures, comprising Qlearning, A3C, and ACER.

Rogueinabox allows an easy configuration of many game parameters, such as:

- monsters
- traps and secret passages
- dark rooms and mazes
- starvation
- location of the amulet



a Rogue layer configured to just contain mazes

Achievements overview

Proviso:

- we just focus on **movement**: no monsters, objects, food, ...
- learning based on a **single level**: find and take the stairs
  - no dark rooms, no traps, no hidden passages
  - maximum steps: 500 moves

Achievements:

agent	random	DQN [2]	**this work**
succes rate	7%	23%	**98%**

Table: Achievements overview

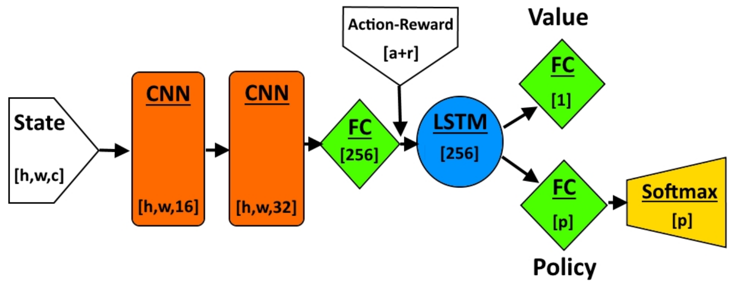
Main architectural ingredients

1. the adoption of A3C as learning framework
2. an agent-centered, cropped representation of the state
3. a supervised partition of the problem in a predefined set of *situations*, each one delegated to a different A3C agent

A3C (Mnih et al.) "On-policy" technique:

- Asyncronous: exploting a set of asynchronous agents
- Advantage: a formal notion expressing the convenience of an action in a given state
- Actor-Critic: the policy π is the actor and the value function V is the critic.

Neural network



Situations and rewards

Situations:

1. corridor
2. stairs in view
3. adjiacent to wall
4. other

Rewards:

1. $+1$ for entering a new door
2. $+1$ for discovering a new doors
3. $+10$ for descending the stairs
4. -0.01 for other actions

Situations and rewards are quite ad-hoc (weak!!)

Demo!

Figure: Agent's behaviour after 40 millions iterations

A longer version is available on **youtube**

Conclusions

- The rogue movement is not perfect, but satisfactory
- Some projectual choices are weak:
  - situations
  - rewarding mechanism
  - cropped view
- We are already working on these issues with promising results

Conclusions

- The rogue movement is not perfect, but satisfactory
- Some projectual choices are weak:
  - situations
  - rewarding mechanism
  - cropped view
- We are already working on these issues with promising results

thanks for your attention

Bibliography

A.Asperti, C.De Pieri, M.Maldini, G.Pedrini, and F.Sovrano.
A modular deep-learning environment for rogue.
WSEAS Transactions on Systems and Control, 12, 2017.

A.Asperti, C. De Pieri, and G.Pedrini.
Rogueinabox: an environment for roguelike learning.
International Journal of Computers, 2:146–154, 2017.

V.Mnih et al.
Asynchronous methods for deep reinforcement learning.
CoRR, abs/1602.01783, 2016.

V.Mnih et al.
Human-level control through deep reinforcement learning.
Nature, 518(7540):529–533, 2015.

Z.Wang et al.
Sample efficient actor-critic with experience replay.
2016.

T.Weber et al.
Imagination-augmented agents for deep reinforcement learning.
CoRR, abs/1707.06203, 2017.