Hierarchical reinforcement learning

Shvechikov Pavel

National Research University Higher School of Economics,
Yandex School of Data Analysis

May 22, 2017

Overview

1. Motivation
2. FeUdal Networks
   - Introduction
   - Training
   - Results

Overview

Humans and animals are naturally hierarchical :

1. **pack** model: alpha, beta and omega dogs
2. **herd** models: single stallion

Every one in the hierarchy knows what and when to do.

1. idea of duties segregation is promising
2. especially when duties are explicitly assigned :)

But how to learn this hierarchy without knowing anything about the task beforehand?

Standard approaches to goal setting

End-to-end learning of hierarchical RL architectures

1. has many advantages
2. results either in sub-goals of length one
3. or in sub-goals of length equal to episode length

To combat this issue several approaches exists

1. explicitly predefine each sub-goal (Tessler et al., 2016)
2. introduce regularizers to (Bacon et al., 2016)

An alternative (Vezhnevets et al., 2017)

- Do not allow gradients to pass between Manager and Worker !

FuNs: FeUdal Networks for HRL (Vezhnevets et al., 2017)

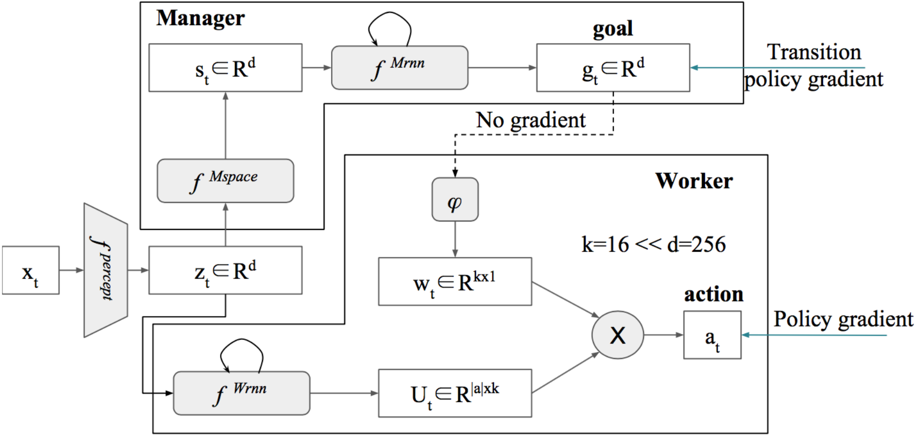
Beneficial when

1. sparse rewards
2. very long timescale credit assignment
3. non-Markovian environements that reqire memory

FuNs at a glance:

1. goal setting is **decoupled** from goal achievement
2. end-to-end **differentiable** model
3. novel **transition policy gradient** for Manager learning
4. **directional goals**, not absolute values

FuN: architecture



FuN: neural network dynamics

Common

$z_t = f^{percept}(x_t)$ is a prepossessed observation x_t

Manager

$$s_t = f^{Mspace}(z_t)$$
$$(h_t^M, \widehat{g}_t) = f^{Mrnn}(s_t, h_{t-1}^M)$$
$$g_t = \widehat{g}_t / ||\widehat{g}_t||$$

s_t – latent state representation
g_t – goal vector
h_t^M – Manager rnn hidden state

Worker

$$w_t = \varphi\left(\sum_{i=t-c}^{t} g_i\right)$$
$$h_t^W, U_t = f^{Wrnn}(z_t, h_{t-1}^W)$$
$$\pi_t = \mathrm{Softmax}(U_t w_t)$$

w_t – goal embedding
$\varphi(\cdot)$ – linear, **without bias**
U_t – matrix, output of rnn
h_t^W – Worker rnn hidden state

FuN: Training Manager

Goal of agent as a whole is $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k=1}$

$$\nabla g_t = (R_t - V_t^M(x_t, \theta)) \cdot d_{cos}(s_{t+c} - s_t,\ g_t(\theta)),$$

- $(s_{t+c} - s_t)$ – how much have we changed the direction
- g_t – direction we told Worker to go to
- $d_{cos}(\alpha, \beta) = \alpha^\top \beta / (|\alpha||\beta|)$ – cosine similarity

Note: we pretend s_t does not depend on θ

FuN: udpate rule motivation

Assume

1. Manager maintains high-level policy $\mu(s_t, \theta)$
2. let $o_t = \mu(s_t, \theta)$ be a sub-policy chosen by high-level policy
3. sub-policy is a fixed duration behavior (of c steps)
4. sub-policy induces distribution over states $p(s_{t+c} \mid s_t, o_t)$
5. high-level policy induces $\pi^{TP}(s_{t+c} \mid s_t) = p(s_{t+c} \mid s_t, \mu(s_t, \theta))$
6. $\nabla_\theta \pi_t^{TP} = \mathbb{E}\left[(R_t - V(s_t))\nabla_\theta \log p(s_{t+c} \mid s_t, \mu(s_t, \theta))\right]$
7. BUT! if

$$p(s_{t+c} \mid s_t, o_t) \propto e^{d_{cos}(s_{t+c} - s_t, \ \ g_t)}$$

we recover Manager updates!

FuN: udpate rule motivation

Assume

1. Manager maintains high-level policy $\mu(s_t, \theta)$
2. let $o_t = \mu(s_t, \theta)$ be a sub-policy chosen by high-level policy
3. sub-policy is a fixed duration behavior (of c steps)
4. sub-policy induces distribution over states $p(s_{t+c} \mid s_t, o_t)$
5. high-level policy induces $\pi^{TP}(s_{t+c} \mid s_t) = p(s_{t+c} \mid s_t, \mu(s_t, \theta))$
6. $\nabla_\theta \pi_t^{TP} = \mathbb{E}\left[(R_t - V(s_t))\nabla_\theta \log p(s_{t+c} \mid s_t, \mu(s_t, \theta))\right]$
7. BUT! if
$$p(s_{t+c} \mid s_t, o_t) \propto e^{d_{cos}(s_{t+c} - s_t, \ \ g_t)}$$
we recover Manager updates!

Eqivalent formulation

$s_{t+c} - s_t$ should follow von Mises-Fisher distribution with mean vector g_t

FuN: dilated LSTM

Note: f^{Mrnn} is not simple RNN !

Allows operation at lower temporal resolution.

1. introduce more *cores* constituting $h = \{\widehat{h}^i\}_{i=1}^{r}$
2. update
$$\widehat{h}_t^{t\%r},\ g_t = \mathrm{LSTM}(s_t,\ \widehat{h}_{t-1}^{t\%r};\ \theta^{\mathsf{LSTM}}) \tag{1}$$
3. $\widehat{h}_t^{t\%r}$ is pooled c steps backwards

FuN: Training Worker

Intrinsic reward for Worker

$$R_t^I = \frac{1}{c} \sum_{i=1}^{c} d_{cos}(s_t - s_{t-i},\ g_{t-i})$$

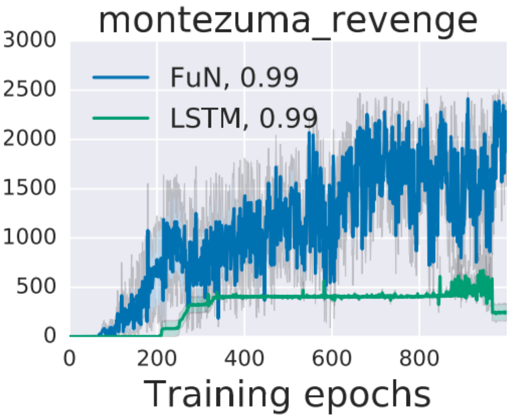
Advantage Actor Critic for Worker training

$$\nabla \pi_t = (R_t + \alpha R_t^I - V_t^D(x_t, \theta)) \ \cdot \ \nabla_\theta \log \pi(a_t \,|\, x_t; \theta)$$

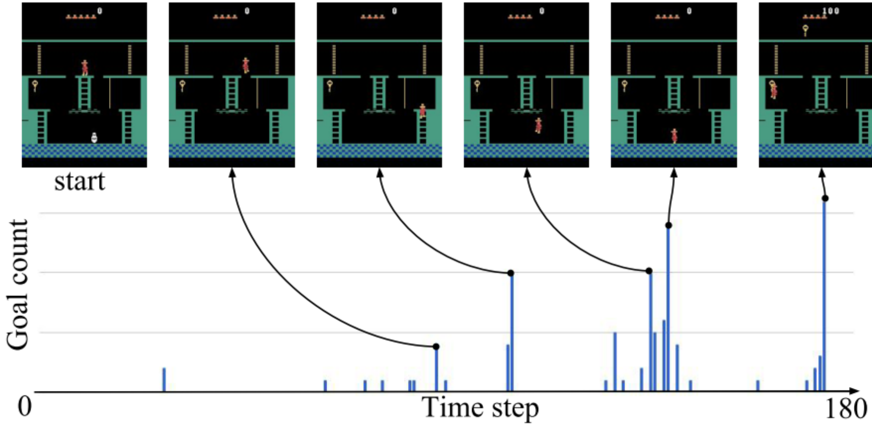
- α – hyper parameter regulating influence of intrinsic reward

Note: discount rate could be different for Manager and Worker

FuN: results


montezuma_revenge
FuN, 0.99
LSTM, 0.99
3000
2500
2000
1500
1000
500
0
0
200
400
600
800
Training epochs

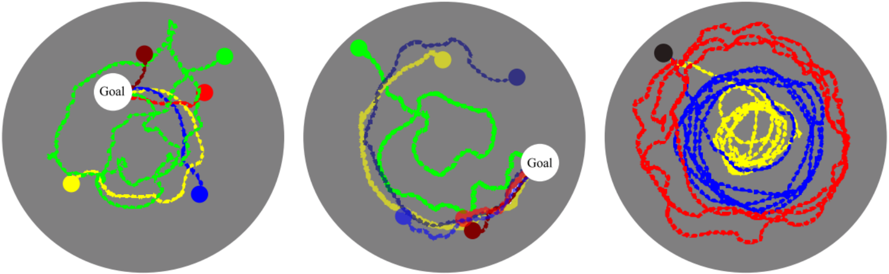
FuN: results



FuN: results


asterix
zaxxon
FuN
Option-Critic
FuN
Option-Critic
Score
250000
200000
150000
100000
50000
0
25000
20000
15000
10000
5000
0
0
50
100
150
Training epochs
0
50
100
150
Training epochs

FuN: results



Thank you!

References I

Bacon, Pierre-Luc et al. (2016). "The Option-Critic Architecture". In: *CoRR* abs/1609.05140. URL: http://arxiv.org/abs/1609.05140.

Tessler, Chen et al. (2016). "A deep hierarchical approach to lifelong learning in minecraft". In: *arXiv preprint arXiv:1604.07255*.

Vezhnevets, Alexander Sasha et al. (2017). "FeUdal Networks for Hierarchical Reinforcement Learning". In: arXiv: 1703.01161. URL: http://arxiv.org/abs/1703.01161.