Modular Multitask Reinforcement Learning with Policy Sketches

Yoonho Lee

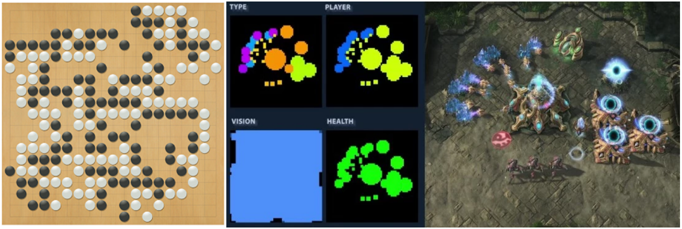
Department of Computer Science and Engineering
Pohang University of Science and Technology

August 09, 2017

Modular Multitask Reinforcement Learning with Policy Sketches

- ICML 2017 Best Paper Honorable Mention
- Hierarchical Reinforcement Learning
- Proposes a looser form of task supervision for RL agents (compared to e.g. reward shaping)
- Proposed form of task supervision is decoupled from the environment
- Proposes a new way to use NNs for hierarchical RL
- Natural extensions to zero-shot and unlabelled RL

Hierarchical Reinforcement Learning



- Why is starcraft harder than go for RL?
- Why is starcraft not harder than go for humans?

Hierarchical Reinforcement Learning

- Decision $\neq$ Action. RL algorithms are designed with decisions in mind, but operate on actions.
- Real world decisions have hierarchical structure(e.g. cook dinner $\rightarrow$ cut potato $\rightarrow$ activate arm muscle)
- Effective knowledge reuse
- Effecient credit assignment
- Open question: How do we discover salient/reusable decisions?

Previous Work

- Learn multiple timestep policy along with confidence[1]
- Learn controller network that sets desired direction of state change[2]
- Encode multiple sub-policies with an SNN[3]
- Actor-Critic algorithm for options[4]

---

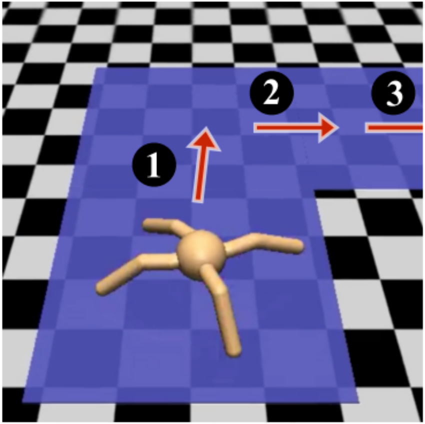
[1] Alexander Vezhnevets et al. "Strategic Attentive Writer for Learning Macro-Actions". In: *NIPS* (2016).

[2] Alexander Sasha Vezhnevets et al. "FeUdal Networks for Hierarchical Reinforcement Learning". In: (2017).

[3] Carlos Florensa, Yan Duan, and Pieter Abbeel. "Stochastic Neural Networks for Hierarchical Reinforcement Learning". In: *ICLR* 2017 (2017), pp. 1056–1064.

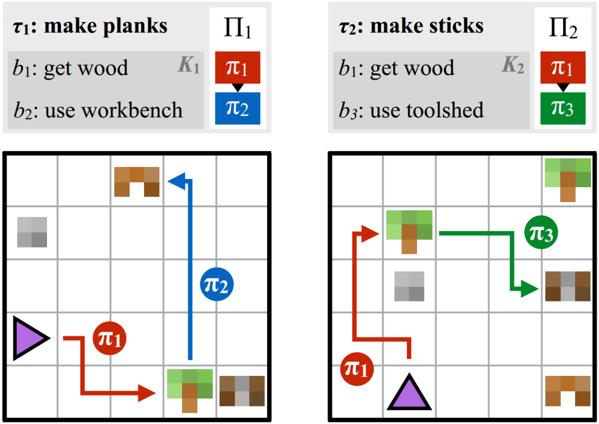
[4] Pierre-Luc Bacon, Jean Harb, and Doina Precup. "The Option-Critic Architecture". In: *AAAI* (2017).

Previous Work



- Previous hierarchical RL algorithms require reward engineering for high-dimensional environments

Proposed Approach

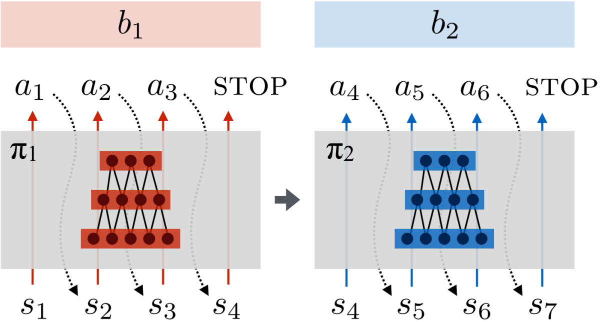


Proposed Approach

Sketches

Goal	Sketch				
Crafting environment					
make plank	get wood	use toolshed			
make stick	get wood	use workbench			
make cloth	get grass	use factory			
make rope	get grass	use toolshed			
make bridge	get iron	get wood	use factory		
make bed*	get wood	use toolshed	get grass	use workbench	
make axe*	get wood	use workbench	get iron	use toolshed	
make shears	get wood	use workbench	get iron	use workbench	
get gold	get iron	get wood	use factory	use bridge	
get gem	get wood	use workbench	get iron	use toolshed	use axe

Method



Method

Algorithm 1 TRAIN-STEP($\boldsymbol{\Pi}$, curriculum)

$\mathcal{D} \leftarrow \emptyset$
while $|\mathcal{D}| < D$ **do**
  // sample task τ from curriculum (Section 3.3)
  $\tau \sim \text{curriculum}(\cdot)$
  // do rollout
  $d = \{(s_i, a_i, (b_i = K_{\tau,i}), q_i, \tau), \ldots\} \sim \Pi_\tau$
  $\mathcal{D} \leftarrow \mathcal{D} \cup d$
// update parameters
for $b \in \mathcal{B}, \tau \in \mathcal{T}$ **do**
  $d = \{(s_i, a_i, b', q_i, \tau') \in \mathcal{D} : b' = b, \tau' = \tau\}$
  // update subpolicy
  $\theta_b \leftarrow \theta_b + \frac{\alpha}{D} \sum_d \big(\nabla \log \pi_b(a_i|s_i)\big)\big(q_i - c_\tau(s_i)\big)$
  // update critic
  $\eta_\tau \leftarrow \eta_\tau + \frac{\beta}{D} \sum_d \big(\nabla c_\tau(s_i)\big)\big(q_i - c_\tau(s_i)\big)$

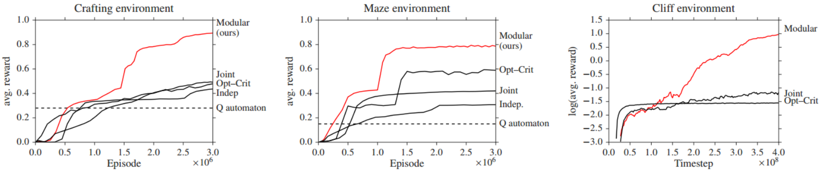
Method

Algorithm 2 TRAIN-LOOP()

// initialize subpolicies randomly
$\mathbf{\Pi} = \text{INIT}()$
$\ell_{\max} \leftarrow 1$
loop
  $r_{\min} \leftarrow -\infty$
  // initialize ℓ_{max}-step curriculum uniformly
  $\mathcal{T}' = \{\tau \in \mathcal{T} : |K_\tau| \leq \ell_{\max}\}$
  $\text{curriculum}(\cdot) = \text{Unif}(\mathcal{T}')$
  while $r_{\min} < r_{\text{good}}$ **do**
    // update parameters (Algorithm 1)
    $\text{TRAIN-STEP}(\mathbf{\Pi}, \text{curriculum})$
    $\text{curriculum}(\tau) \propto \mathbb{1}[\tau \in \mathcal{T}'](1 - \hat{\mathbb{E}} r_\tau) \quad \forall \tau \in \mathcal{T}$
    $r_{\min} \leftarrow \min_{\tau \in \mathcal{T}'} \hat{\mathbb{E}} r_\tau$
  $\ell_{\max} \leftarrow \ell_{\max} + 1$

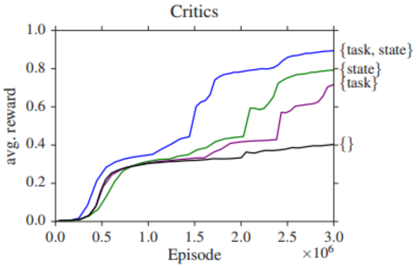
Experiments

Comparison to Baseline

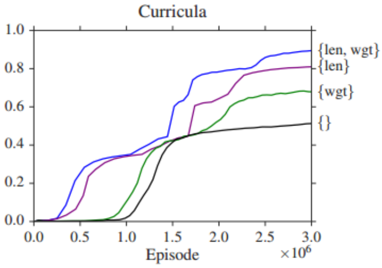


Experiments

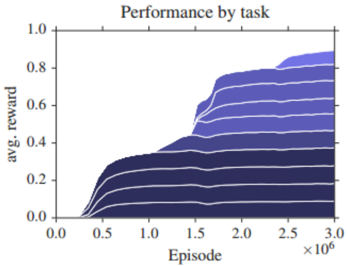
Ablation



(a)



(b)



Experiments

Accuracy

Model	Multitask	0-shot	Adaptation
Joint	.49	.01	–
Independent	.44	–	.01
Option–Critic	.47	–	.42
Modular (ours)	**.89**	**.77**	**.76**

Discussion

Weaknesses

- Only works for tasks that are strictly composed
- To label, one needs to understand which tasks are "RL-able"
- Labels needed: Fundamentally limited to tasks understood by humans

Discussion

Implications

- Zero-shot and Adaptation experiments had similar performance, what does this imply?
- Weight sharing between policies?
- Curriculum instead of policy sketches?
- Common theme of cooperating networks for hierarchical RL: Manager-Worker[5], Option-Policy[6], Policy modules[7]

[5] Alexander Sasha Vezhnevets et al. "FeUdal Networks for Hierarchical Reinforcement Learning". In: (2017).

[6] Pierre-Luc Bacon, Jean Harb, and Doina Precup. "The Option-Critic Architecture". In: *AAAI* (2017).

[7] Jacob Andreas, Dan Klein, and Sergey Levine. "Modular Multitask Reinforcement Learning with Policy Sketches". In: *ICML* (2017). arXiv: 1611.01796.

References I

[1] Jacob Andreas, Dan Klein, and Sergey Levine. "Modular Multitask Reinforcement Learning with Policy Sketches". In: *ICML* (2017). arXiv: 1611.01796.

[2] Pierre-Luc Bacon, Jean Harb, and Doina Precup. "The Option-Critic Architecture". In: *AAAI* (2017).

[3] Carlos Florensa, Yan Duan, and Pieter Abbeel. "Stochastic Neural Networks for Hierarchical Reinforcement Learning". In: *ICLR* 2017 (2017), pp. 1056–1064.

[4] Alexander Vezhnevets et al. "Strategic Attentive Writer for Learning Macro-Actions". In: *NIPS* (2016).

[5] Alexander Sasha Vezhnevets et al. "FeUdal Networks for Hierarchical Reinforcement Learning". In: (2017).

Thank You