

1704.03012 - Stochastic Neural Networks for Hierarchical Reinforcement Learning

- Yunqiu Xu
 - Other reference:
https://github.com/DanielTakeshi/Paper_Notes/blob/master/reinforcement_learning/Stochastic_Neural_Networks_for_Hierarc
 - Further reading: 1710.09767 - Meta Learning Shared Hierarchies
-

1. Introduction

- Challenges: sparse rewards and long horizons, this is same with [Feudal Network](#)
- Our work:
 - Learns a span of skills in a pre-training environment
 - The environment is employed with only proxy reward signal
 - The design only requires very minimal domain knowledge about downstream tasks
 - Proxy reward:
 - A form of intrinsic motivation
 - Encourage the agent to explore its own capabilities + Do not need any goal information or sensor readings specific to each downstream task
 - The set of skills can be used later for other tasks
 - Use Stochastic NN to learn the span of skills
 - Can easily represent multi-modal policies
 - Achieve weight sharing among different modes

2. Related Work

- Hierarchy over actions: **FeUdal Net is recent work**
 - Composing low-level actions into high-level primitives
 - Search space can be reduced exponentially
 - Require domain-specific knowledge and careful hand-engineering
- Use intrinsic rewards to guide exploration:
 - Do not require domain-specific knowledge
 - Hard to generalize → high overall complexity

3. Problem Statement

- Assumptions:
 - For each MDP $m \in \mathcal{M}$, the state space \mathcal{S}^m can be factored into \mathcal{S}_{agent}^m and \mathcal{S}_{rest}^m , which are weakly interact with each other
 - \mathcal{S}_{agent} is same for all MDPs
 - All MDPs share same action space
- Goal: given a collection of tasks satisfying the assumptions, minimize the total sample complexity required to solve these tasks
- We take advantage of a pre-training task that can be constructed with minimal domain knowledge, and can be applied to the more challenging scenario

4. Method

- Construct pretraining environment:
 - Let the agent freely interact with the environment in a minimal setup
 - Skills learned depend on the reward given to the agent → we use a generic proxy reward as the only reward signal to guide skill learning
 - The design of the proxy reward should encourage the existence of locally optimal solutions
 - It encodes the prior knowledge about what high level behaviors might be useful in the downstream tasks, rewarding all of them roughly equally
 - Every time we train a usual uni-modal gaussian policy in this environment, it should converge towards a potentially different skill
- Learn skills via stochastic NN → represent multi-model policies

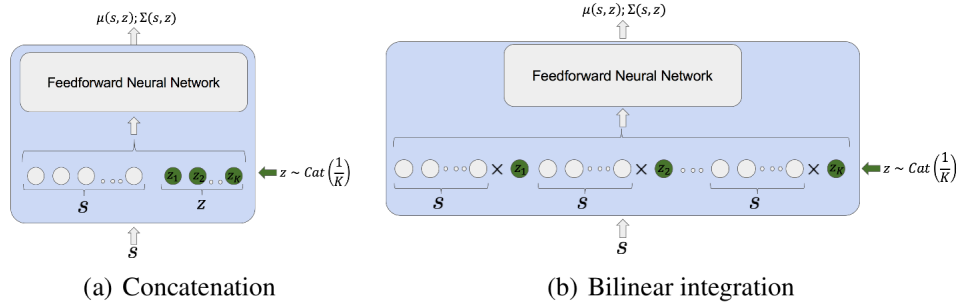


Figure 1: Different architectures for the integration of the latent variables in a FNN

- Information-theoretic regularization → prevent multi-model policies from collapsing into a single mode
- Learn high-level policies : instead of learning from scratch the low-level controls, we leverage the provided skills by freezing them and training a high-level policy (Manager Neural Network) that operates by selecting a skill and committing to it for a fixed amount of time steps

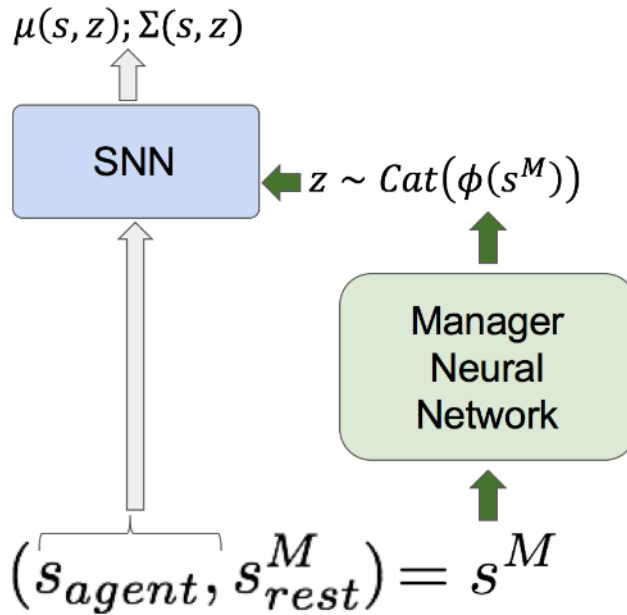


Figure 2: Hierarchical SNN architecture to solve downstream tasks

- Policy optimization: TRPO

5. Experiment

- Locomotion + Maze and Locomotion + Food Collection (Gather)

4. My thoughts

- More abstract than FeUdal Network, and I wonder whether this work can be better
- Maybe I can find more interesting later :D