Summary of FeUdal Networks for Hierarchical Reinforcement Learning

Ashutosh Kumar Jha, ME14B148
Mechanical Engineering, IIT Madras
Topics in RL Course

***Abstract*—In this technical report, we summarize the contents and ideas presented in the technical paper titled *FeUdal Networks for Hierarchical Reinforcement Learning*[1]. We also present the possible shortcomings of the work.**

***Keywords*—Deep Reinforcement Learning, Hierarchical Reinforcement Learning, Feudal RL, Temporally Extended Actions**

I. Summary

Learning temporally extended actions is an active area of research in Deep Reinforcement Learning. Interest in such actions stems out of the problem of long term credit assignment in RL. Older work in RL describe the framework of Feudal RL where there are different levels of hierarchy *in the agent* and the different levels communicate with each other by using explicit goals.

In Feudal RL, the goals are generated in a top-down fashion. What this means is that the upper levels of hierarchy tell the lower levels on *what* they want the lower levels to achieve. *How* they achieve these goals is still up to the lower levels to learn. The lower levels thus have to learn temporally extended actions if the instruction from the upper level comes at a lower temporal resolution.

The paper works with a fully end-to-end differentiable two level hierarchal agent. The top level is called the Manager and the lower level is called the Worker. The Manager sets goals for the Worker and the Worker is motivated to follow these goals using an intrinsic reward. **No gradients are propagated between the Worker and the Manager**. Thus, the Manager has to learn to maximize extrinsic rewards.

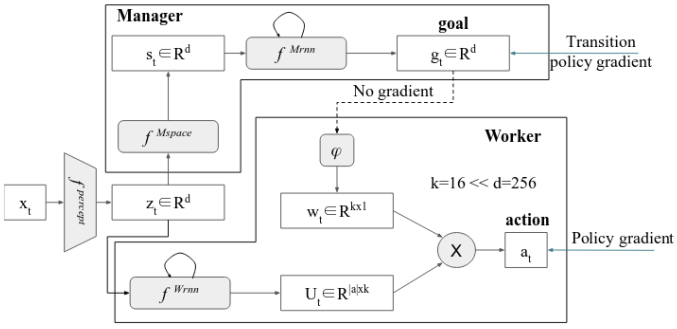

Fig. 1. Schematic of FuN architecture

The Manager sets goals for the Worker which are directional and **not** explicit in nature. Figure I shows a schematic of the entire architecture. Mathematically, the set of equations which describe the architecture are the following:

$$
\begin{aligned}
z_t &= f^{\text{percept}}(x_t) \\
s_t &= f^{\text{Mspace}}(z_t) \\
h_t^M, \hat{g}_t &= f^{\text{Mrnn}}(s_t, h_{t-1}^M) \\
g_t &= \frac{\hat{g}_t}{||\hat{g}_t||} \\
w_t &= \phi\left(\sum_{i=t-c}^{t} g_i\right) \\
h^W, U_t &= f^{\text{Wrnn}}(z_t, h_{t-1}^W) \\
\pi_t &= \text{softmax}(U_t w_t)
\end{aligned}
$$

Both the worker and the manager are recurrent neural networks. h^M and h^W are the internal states of the manager and the worker recurrent networks respectively.

The worker produces an embedding for each of the actions(as the rows of the matrix of U). The last c goals are then summed and projected in the vector w. The manager's goals(in the form of w) thus modulates the policy by a multiplicative interaction in the low k dimensional space.

While the entire architecture is end-to-end differentiable the training of the worker and the manager are separated to make sure the Manager learns directly from the environment. The manager is trained to set the goals in advantageous directions in the state space:

$$\nabla g_t = A_t^M \nabla_\theta d_{cos}(s_{t+c} - s_t, g_t(\theta))$$

The worker on the other hand is motivated to follow the goals of the manager with the help of an intrinsic reward:

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

The worker is then trained using the Actor-Critic algorithm using a weighted sum of the extrinsic and the intrinsic returns:

$$
\begin{aligned}
\nabla \pi_t &= A_t^D \nabla_\theta \log \pi(a_t | x_t; \theta) \\
A_t^D &= R_t + \alpha R_t^I - V_t^D(x_t; \theta)
\end{aligned}
$$

We also note that the worker and the manager can be trained using different values of γ and thus have different focus on

long term-short term rewards.

The manager can be trained as if it had a high level policy that selects sub policies $o_t = \mu(s_t, \theta)$. A high level policy can be composed with the transition distribution to give the "transition policy" and thus, be learned using policy gradients:

$$\nabla_\theta \pi_t^{TP} = \mathbb{E}[(R_t - V(s_t))\nabla_\theta \log p(s_{t+c}|s_t, \mu(s_t, \theta))]$$

The paper assumes a particular form of the transition model where the direction in the state space $(s_{t+c} - s_t)$ follows a von Mises-Fisher distribution. Mathematically this implies that if the mean direction of the von Mises-Fisher distribution is given by g_t then, we have:

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

A. Architectural Details:

The perceptual module f^{percept} is a CNN followed by a fully connected layer. The state space in which the Manager implicitly models in formulating its goals is computed via f^{Mspace} which is another fully connected layer followed by a rectifier non-linearity. The Worker's network f^{Wrnn} is a standard LSTM. For the manager's network, the paper proposes a dilated LSTM.

The dilated LSTM is defined as follows: For a dilation radius r let the full state of the network be $h = \{\hat{h}^i\}_{i=1}^r$ (composed of r cores). The governing equations for the network at time t is given by:

$$\hat{h}_t^{t\%r}, g_t = \text{LSTM}(s_t, \hat{h}_{t-1}^{t\%r}; \theta^{\text{LSTM}})$$

where % denotes the modulo operation. At each time step only the corresponding part of the state is updated and the output is pooled across the previous c outputs. As a result, the groups of cores inside the dLSTM can preserve memories for long periods(because of updates happening at a larger time period). The dLSTM as a whole however is able to process and learn from every input experience.

B. Experiments and Results:

The main baseline of the paper is essentially a simple A3C agent with 32 asynchronous threads. On Montezuma's Revenge, which is believed to be a very hard game unless temporally extended actions are learned, the FuN architecture achieves a score of 2600 (A3C achieves 500). The FuN agent also achieves good scores on similar games like Enduro and Frostbite where long-term credit assignment is important. On the games where the agent loses to A3C, the authors claim that long-term credit assignment isn't really required to solve the game.
The authors also compare to Option-Critic [2]. The FuN agent beats Option-Critic on all the games considered in the Option-Critic paper.

The paper also shows results on Labyrinth Tasks: Water Maze, T-Maze and Non-Match. FuN consistently outperforms the LSTM baseline - it learns faster and also reaches a higher final reward. Interestingly, the LSTM agent doesnt appear to use its memory for water maze task at all, always circling the maze at the roughly the same radius.

The authors also perform an Ablative Analysis of the FuN agent to show that the each component of the agent helps it achieve superior performance to the compared baselines.

C. Transfer Learning Experiments:

Since there is clear separation between the Manager and the Worker, the transition policy is invariant to the way the agent's primitive actions translate into state space transitions. Potentially, the transition policy can be transferred between agents with different embodiment.
The paper empirically shows transfer learning by initializing a FuN system with parameters of an agent trained with action repeat of 4. The discounts of this FuN agent is then adjusted accordingly, the dilation and the goal horizon of the manager is increased by a factor of 4. The agent is then trained without action repeat. The baselines considered are an LSTM agent transferred in a similar way(with adjusted discounts) as well as FuN and LSTM agents trained without action repeat from scratch. The transferred FuN agent outperforms all the baselines by a margin. Furthermore it shows positive transfer on each environment, whereas LSTM only shows positive transfer on Ms. Pacman.

II. Shortcomings

In this section we make a note of possible short comings of this paper:

- The use of von Mises-Fisher distribution to model the transitions is not really justified.
- In the original Feudal RL paper, there were deeper hierarchies possible. Here however the framework only allows for a 2 level hierarchy.

III. Conclusion

This is clearly a brilliant paper in terms of the experiments and novelty in framework used in terms of the use of a dilated LSTM. The experiments were clear and well motivated. The ablative analysis empirically showed the importance of each contribution of the paper. I personally enjoyed the paper a lot and was disappointed by only the above two shortcoming (the second of which is in fact an interesting direction for future work).

References

[1] FeUdal Networks for Hierarchical Reinforcement Learning, Alexander Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver and Koray Kavukcuoglu. International Conference on Machine Learning (ICML), 2017.

[2] The Option-Critic Architecture, Pierre-Luc Bacon, Jean Harb and Doina Precup. Thirthy-first AAAI Conference On Artificial Intelligence (AAAI), 2017