Summary for Curiosity-driven Exploration by Self-supervised Prediction

Siddharth Nayak

1 Introduction

When we learn something new on our own, generally we get a very sparse reward or no reward at all. But even without that we explore different cases due to our curiosity. The paper 'Curiosity-driven Exploration by Self-supervised Prediction' investigates an RL model which learns without any extrinsic rewards. An RL agent receives the reward only at the end of the episode. And with the initial random exploration, it is quite rare that the agent might end up in the GOAL state. So the agent might never receive the reward and thus may never learn. Things like this happen even when human beings learn new things and this is when the human curiosity comes into picture.

2 Model

The agent is composed of two subsystems: a reward generator that gives curiosity based intrinsic rewards and a policy that outputs a sequence of actions which try to maximise the rewards. The intrinsic reward generated by the agent at time t is denoted by r_t^i and the extrinsic reward is denoted by r_t^e.

The policy is trained with reward $r_t = r_t^i + r_t^e$. Here, r_t^e is mostly zero. The policy $\pi(s_t, \theta_P)$ is represented by a deep neural network. θ_P is optimised to maximise the expected sum of rewards, $\underset{\theta_P}{\max}\mathbf{E}_{\pi(\mathbf{s_t},\boldsymbol{\theta_P})}[\Sigma_t r_t]$. The prediction error of the actions and the state encodings are used as the intrinsic curiosity rewards. A feature encoder is used to encode the statess_t to $\phi(s_t)$ and s_{t+1} to $\phi(s_{t+1})$. A neural network takes in action taken a_t and $\phi(s_t)$ and gives out a predicted estimate of next state $\tilde{\phi}(s_{t+1}) = f\Big(\phi(s_t), a_t; \theta_F\Big)$ and they are optimised using the loss function $L_F = \frac{1}{2}||\tilde{\phi}(s_{t+1}) - \phi(s_{t+1})||_2^2$. This sub module is called the forward model.

The intrinsic reward is given by, $r_t^i = \frac{\eta}{2}||\tilde{\phi}(s_{t+1}) - \phi(s_{t+1})||_2^2$.

Another neural network is used to get the predicted estimate of the action $\tilde{a}_t = g(\phi(s_t), \phi(s_{t+1}), \theta_I)$. The parameters θ_I are optimised by $\underset{\theta_I}{\min}L_I(\tilde{a}_t, a_t)$. Here, L_I is the discrepancy between the action a_t and the estimated action$\tilde{a}_t$. This module is called the inverse model.

So the overall optimisation problem used for learning the policy is as follows:

$$\min_{\theta_P,\theta_F,\theta_I}\Big[-\lambda\mathbf{E}_{\pi(s_t;\theta_P)}[\Sigma_t r_t] + (1-\beta)L_I + \beta L_F\Big]$$

where $0 \leq \beta \leq 1$ is a scalar and $\lambda > 0$.

This whole model is called the Intrinsic Curiosity Model(ICM). Overall the curiosity model gets intrinsic rewards according to the error of the agent in predicting the consequences of its own actions. So this model encourages the agent to take actions which reduce the uncertainty in the agent's ability to predict the consequences of it's own action.

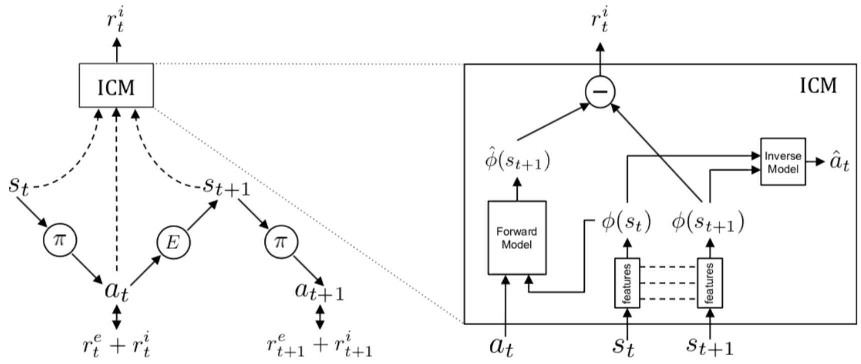

Figure 1: The Intrinsic Curiosity Model[1](ICM)

3 Experiments

The authors evaluate the ICM on two different environments: VizDoom and Super Mario Bros. The agents are trained with visual inputs. The input RGB images are converted to gray-scale and resized to 42 x 42. The state for the agent is the current frame concatenated with three previous frames. The authors used A3C with ADAM optimizer where the workers did not share the parameters.

3.1 VizDoom

Varying the degree of reward sparsity: The agent is evaluated with different locations of spawning. In all three cases(very-sparse, sparse and dense) the ICM model performs better than the vanilla A3C by getting a higher extrinsic reward per episode. They define another model called the 'ICM-Pixels' which is without the inverse model. So in this case the estimated actions are not penalised. The 'ICM-Pixels' performs better than ICM in the sparse reward setting by converging faster than the ICM. The reason for this is not given in the paper. The authors also try out augmenting the state for the agent by adding a fixed region of white noise to the image. Ideally the agent should not be affected by this white noise but ICM-Pixels suffers significantly. But ICM performs well even in this setting. In the end they try out the 'No Reward' setting which explores more number of rooms in VizDoom than a vanilla RL agent with extrinsic rewards.

3.2 Super Mario Bros.

The authors train the agent on the Level-1 of the game without any extrinsic rewards and the agent is able to complete 30% of the Level-1 and surprisingly it learns to dodge and kill the enemies. The agent does not perform well in Level-2 but does well on Level-3 because of the similarity in the first and the third level. Note that this performance degradation is only when the agent is trained without any fine-tuning above the originally trained model. After fine-tuning with curiosity only, the agent does well even in Level-2. This performance is better than the model which is trained from scratch on Level-2. The reason for this being that the basic skills such as moving and jumping is learnt in a better way in Level-1 than in Level-2. The authors claim that the agent's performance deteriorates when it is fine-tuned on Level-3 from Level-1. The reason they give for this is that the agent gets "bored" of exploring as it has already learned about parts of the environment. But the Level-3 has quite a few new elements as compared to Level-1 like the trampoline, Venus fly-trap from the pipe, etc. And thus the agent should have explored and obtained some curiosity rewards which does not happen in this case.

4 Possible Improvements

One of the problems with this model is that the agent's curiosity gets saturated after sometime just like a human being and because of this the agent stops exploring. So, to tackle this problem, we can train the agent on the previous level so that it can use it's new knowledge about the game. This new knowledge might make the agent better in the previous levels. This is somewhat similar to how human beings go on to the previous level and come back to that level, whenever they do not make any progress in the current level of a game.

5 Conclusion

The paper was quite interesting especially the method of having no extrinsic rewards for the agent to learn different policies. I also found the method of having a forward and an inverse model which penalise the agent's (in)ability to predict consequences of it's own actions, quite interesting. This model does tackle the problem of having very sparse external rewards quite well and the results show that this performs better than vanilla RL models in complex environments.

6 References

Pathak, Agrawal, Efros, Darrell. Curiosity-driven Exploration by Self-supervised Prediction. Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 2017.