ERLP: Ensembles of Reinforcement Learning Policies (Student Abstract)

Rohan Saphal,[1] Balaraman Ravindran,[1] Dheevatsa Mudigere,[2*] Sasikanth Avancha,[3] Bharat Kaul[3]
[1]Robert Bosch Centre for Data Science and Artificial Intelligence,
Indian Institute of Technology Madras, Chennai, India -600036
[2]Facebook AI Research, Facebook Inc, USA
[3]Parallel Computing Lab, Intel Labs, Bangalore
{rohansaphal, dheevatsa}@gmail.com, ravi@cse.iitm.ac.in, {sasikanth.avancha, bharat.kaul}@intel.com

Abstract

Reinforcement learning algorithms are sensitive to hyper-parameters and require tuning and tweaking for specific environments for improving performance. Ensembles of reinforcement learning models on the other hand are known to be much more robust and stable. However, training multiple models independently on an environment suffers from high sample complexity. We present here a methodology to create multiple models from a single training instance that can be used in an ensemble through directed perturbation of the model parameters at regular intervals. This allows training a single model that converges to several local minima during the optimization process as a result of the perturbation. By saving the model parameters at each such instance, we obtain multiple policies during training that are ensembled during evaluation. We evaluate our approach on challenging discrete and continuous control tasks and also discuss various ensembling strategies. Our framework is substantially sample efficient, computationally inexpensive and is seen to outperform state of the art (SOTA) approaches

Introduction

Traditionally, the idea of using ensembles in reinforcement learning settings is associated with combining multiple value functions or policies from different models. These models could be the same algorithm trained across different hyper-parameter settings or different algorithms altogether. Training multiple such models is an approach that cannot be used in practice owing to high sample complexity and computational cost. Our work tackles the above drawbacks by learning multiple models from a single training instance through directed perturbation of model parameters at regular intervals. We leverage the theory of cyclical learning rates (Loshchilov and Hutter 2016) for this purpose. When the model parameters are perturbed using larger learning rates, the directed motion along the gradient step prevents the optimizer from settling in any sharp basins and moves into the general vicinity of the local minima. Lowering the learning rates at such an instance leads the optimizer to converge to some final local minima. We leverage the diversity of the policies learned at these different local minima for the ensemble.

*Work done while at Intel Labs, Bangalore

ERLP

ERLP results in an ensemble of diverse policies obtained from a single training instance. In a traditional ensemble setting, if each agent requires N number of samples and the computational expense for training a single agent is C, then training M agents independently requires $M \times N$ samples and $M \times C$ in computational cost. If trained in parallel, only N samples are required, but the computational cost remains at $M \times C$. Though training multiple agents in parallel is a sound solution to tackle sample complexity, it is computationally expensive and limits the diversity among the learned policies, since every policy observes the same state at each instance.

Our approach saves policies during training at periodic intervals when the learning rate anneals to a small value and ensembles them during evaluation time. ERLP requires only N number of samples, the computational expense is C, and yet we obtain M models for the ensemble. Since the policies have been saved at different local minima, the policies are diverse in nature.

Learning policies

To learn multiple policies from a single training instance, we perturb the parameters of the model along the gradient direction at regular intervals . For learning M models, we split the training process into M different training cycles wherein each cycle the model starts at a high learning rate and anneals to a small value. The large learning rate is significant as it provides energy to the policy to escape the local minima and the small learning rate traps it into a well behaved local minima.

Diversity of Policies

We establish more concretely the diversity of the individual policies, by understanding the action distribution for each policy across states. We compute the KL divergence between the policies based on the action distribution across a number of states. The greater the KL divergence between the policies, the more diverse the policies are. From Figure 2, we can observe that, the ERLP policies are diverse in nature and have a gradual decrease in the diversity as new models are formed. Conversely, for the baseline models, the KL divergence between the independently trained policies is extremely large. The policies did not have much overlap

in the action space and hence ensemble techniques such as majority voting were unable to find a good action, thereby resulting in a poor ensemble. We can hence conclude that ERLP is able to generate policies with sufficient diversity for a good ensemble.

Ensemble techniques

Depending on the complexity of the action space, discrete or continuous, there are multiple strategies to ensemble the m policies in the environment. For discrete action space environments, we propose using majority voting over the actions from the policies. For continuous action space, we propose four methodologies, averaging, binning, density-based selection, and selection through elimination.

Experiments and Results

We consider the Breakout environment from the Atari 2600 game suite and Half Cheetah from Mujoco (Todorov, Erez, and Tassa 2012). We conduct our experiments on the following algorithms, A2C (Mnih et al. 2016), ACER (Wang et al. 2016), DDPG (Lillicrap et al. 2017), SAC (Schulman et al. 2017) and TRPO (Schulman et al. 2015). We compare the training and evaluation performance of ERLP with the normal training and three baseline ensemble methods, respectively. The first baseline ensemble strategy, B1, ensembles policies trained from the same algorithm. Strategy B2, ensembles policies trained from different algorithms and finally, B3, uses policies that are obtained through random perturbation of the model parameters at regular intervals. ERLP performance during training is atleast at par or better than the baseline as shown in Figure 1. The evaluation results using ERLP is better than the three baselines and in many cases, outperforms the state-of-the-art (SOTA) results, as shown in Figure 2 (SOTA score in Breakout is 681.9, ERLP score is 815).

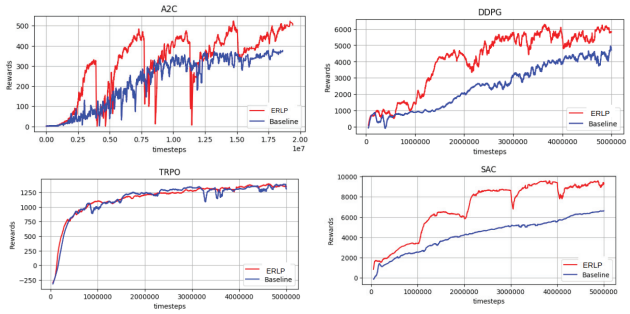

Figure 1: Comparison of training performance between ERLP and Baseline. Top : (Left) Breakout using A2C , (Right) Half Cheetah using DDPG , Bottom : Half Cheetah using TRPO , (Right) Half Cheetah using SAC

Conclusion

In this paper, we introduce ERLP, a framework to ensemble multiple policies obtained from a single training instance. ERLP outperforms the three baseline methods in complex

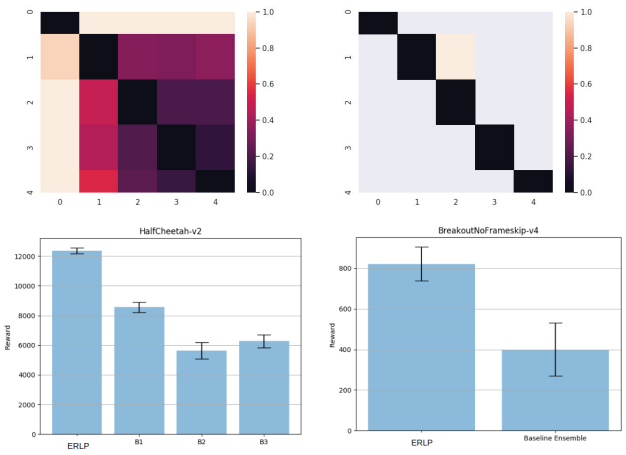

Figure 2: Top : (Left) KL Divergence between ERLP policies for Breakout using A2C (Right) KL Divergence between independently trained policies used in baseline, B1, for Breakout using A2C. Bottom : (Left) Comparison between ERLP and baseline ensembles for Half Cheetah using SAC. The ensemble strategy used is Binning (Right) Comparison between ERLP and baseline ensembles B1 for Breakout using A2C

environments having discrete and continuous action spaces. We show our results using various reinforcement learning algorithms and therefore claim that it is not limited to its performance in any particular setting and can be used with any new and upcoming algorithms.

References

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N. M. O.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. P. 2017. Continuous control with deep reinforcement learning. US Patent App. 15/217,758.

Loshchilov, I., and Hutter, F. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T. P.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. *CoRR* abs/1602.01783.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust region policy optimization. In *International Conference on Machine Learning*, 1889–1897.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *CoRR* abs/1707.06347.

Todorov, E.; Erez, T.; and Tassa, Y. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5026–5033. IEEE.

Wang, Z.; Bapst, V.; Heess, N.; Mnih, V.; Munos, R.; Kavukcuoglu, K.; and de Freitas, N. 2016. Sample efficient actor-critic with experience replay. *CoRR* abs/1611.01224.