Transfer Learning for Continuous Control

Himani Arora
Columbia University
ha2434@columbia.edu

Rajath Kumar
Columbia University
rm3497@columbia.edu

Jason Krone
Columbia University
jpk2151@columbia.edu

November 20, 2017

Abstract

Transfer learning has been extremely successful in the computer vision domain. This exemplified by the fact that most state-of-the-art deep classification, detection, and segmentation networks are first pre-trained on the imagenet dataset and then fine tuned on the evaluation dataset. However, in the Reinforcement Learning domain, transfer learning has not exhibited the same level of success. Policies learned by agents are not easily transferable to new similar tasks. Moreover, most research in this area has been confined to discrete action spaces. In this work, we benchmark and explore variants of existing policy transfer methods on continuous control tasks, which until now have primarily been applied to to discrete action spaces.

Keywords: Continuous Control; Deep Reinforcement Learning; Transfer Learning.

1 Related Work

In recent years there have been a number of works, most consistently from Deep Mind, on methods of transfer learning and multi-task learning for Deep Reinforcement Learning agents. These works primarily focus on two approaches: knowledge distilation and feature reuse across tasks. Knowledge distilation, originally proposed in Cristian Bucila (2006), serves as the foundation for the methods put forward in Rusu et al. (2016a), Parisotto et al. (2016), and Teh et al. (2017). Rusu et al. (2016a) extends the distilation method, formulated in Hinton et al. (2014), to Deep Q Networks trained on Atari environments and demonstrates that policy disitilation can act as a form of regularization for Deep Q Networks. In Parisotto et al. (2016) the authors propose a novel loss function that includes both a policy regression term as well as a feature regression term. The former objective is traditionally used for distilation and the latter encourages intermediate representations from the student network to match those of the expert network. Teh et al. (2017) applies distaliation to the multi-task setting by learning a common (distilled) policy across a number of 3D environments. In addition, Teh et al. (2017) makes use of an entropy penalty and utilizes KL and entropy regularization coefficients to trade off between encouraging exploration or encouraging actions which have high probability under the distilled policy respectively. Rusu et al. (2016b) attack the problem of catastrophic forgetting wherein a networks ability to preform the original task used for pre-training is lost after the network undergoes a transfer learning process on a target task by reusing task specific hidden representations within a network. Henderson et al. (2017) introduced the environments we use in our experiments. These environments are extensions to the mujoco continuous control tasks available in Open AI Gym. Small variations between environments such as the strength of gravity and the length of an agent's body parts make the extensions an ideal test be for transfer learning and multi-task learning.

2 Approach

We assume that all tasks in this work take the form of a Markov Decision Process (S, A, T, R, γ) consisting of a set of states S, a set of actions A, a transition function $T(s'|s, a)$ that gives the probability of transitioning to state s when taking action a in state s, a reward function R that maps state action pairs to a scalar reward, and a discount factor γ. The behavior of our agent is determined by a policy $\pi(a|s)$ which maps a state to a distribution over possible actions. Furthermore, we definie the optimal approximate policy to be $\pi_{\theta^*}(a|s)$ where $\theta^* = \arg\max_\theta E_{\tau\sim p_\theta(\tau)}[\sum_t r(s_t, a_t)]$ and $\pi_\theta(\tau) = p(s_1)\prod_t^T \pi_\theta(a_t|s_t)T(s_{t+1}|s_t, a_t)$. To learn this optimal arppoximate policy we use the advantage actor-critic (A2C) algorithm. This algorithm maintains a policy π_θ and a value function estimate V_{θ_V}, and it performs updates of the form $\nabla_\theta \pi_\theta(a_t|s_t)A_{\theta,\theta_V}(s_t, a_t)$ where $A_{\theta,\theta_V}(s_t, a_t) = \sum_{i=0}^{k-1} \gamma^i r_{t+i} + \gamma^k V_{\theta_V}(s_{t+k}) - V_{\theta_V}(s_t)$. We use feed forward neural networks to approximate both π_θ and V_{θ_V}.

In our knowledge distilation experiments we transfer knowledge from a teacher model T to

a student model S. In this formulation, we assume that T is the policy of an A2C model that has been trained from scratch on a single environment and S is a newly initalized feed forward network. We perform knowledge transfer by training the student network S on the outputs of T, which we relax using a temperature τ. Specifically, if p_i is the probability of taking action a_i in state s_t under T,then S is trained to output the target value $\frac{p_i}{\tau}$.

3 Experiments

We conduct our experiments using the half-cheetah agent on morphologically modified variants of the Open AI gym extensions described in Henderson et al. (2017). We experiment with the following methods of transfer learning:

1. Fine-tuning a pre-trained policy: We train a policy from scratch on one (or more) agent(s) and then transfer it to the other agent by freezing the initial layers and only fine-tuning the final layers of the A2C network.
2. Knowledge distillation from teacher to student: We use an expert network to train a student network (on the same task or different task) using the different knowledge distillation techniques proposed in Rusu et al. (2016a). We also experiment with model compression technique on the student network.
3. Multi-task learning: We train a network by switching between episodes of different tasks to update the weights of the A2C network.
4. Using progressive network Rusu et al. (2016b): We train the initial weight columns on the source tasks and then train subsequent columns on the target tasks using lateral connections to facilitate knowledge transfer.

To determine the relative success of the above methods we compare against the performance of a model trained from scratch on a single environment. We report the mean and standard deviation of the cumulative reward across 20 sample rollouts on each target environment as done in Henderson et al. (2017). In addition, we plot the learning curves of each method in order to determine the sample efficiency of these approaches.

For the milestone, we foussed mainly on HalfCheetahSmallFoot-v0 and HalfCheetahSmallLeg-v0 from Henderson et al. (2017) which reduce the size of the agent's foot and leg by 25% respectively and ran experiments using the first method of transfer learning (fine-tuning). We used PyTtorch for implementing our models. Our actor and critic network consists of 2 and 3 fully-connected layers respectively of 64 units each. Since the Mujoco environment is modelled as a continuous control environmnet, the final action is sampled from a gaussian distribution with learnable mean and variance. Training is performed using RMSprop with an initial learning rate of 0.0007 an an update to the A2C network is performed after 5 forward steps. We first train a policy with random initialization of weights for 5M frames on each agent separately. We then transfer this policy on an agent by initializing its weights from the other agent and then fine-tuning only the final layers of the actor-critic networks for 5M frames.

4 Milestone Deliverables

We expect to achieve the following items by the project milestone date of Nov. 20.

References

Cristian Bucila, Rich Caruana, A. N.-M. (2006). Model compression. *ACM*.

Henderson, P., Chang, W.-D., Shkurti, F., Hansen, J., Meger, D., and Dudek, G. (2017). Benchmark environments for multitask learning in continuous domains. *arXiv preprint arXiv:1708.04352v1*.

Hinton, G., Vinyals, O., and Dean, J. (2014). Distilling the knowledge in a neural network. *Deep Learning and Representation Learning Workshop, NIPS*.

Parisotto, E., Ba, J., and Salakhutdinov, R. (2016). Actor-mimic deep multitask and transfer reinforcement learning. *ICLR 2016*.

Rusu, A. A., Colmenarejo, S. G., Gulcehr, C., Desjardins, G., Kirkpatrick, J., Pascanu, R., Mnih, V., Kavukcuoglu, K., and Hadsell, R. (2016a). Policy distilation. *arXiv preprint arXiv:1511.06295v2*.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. (2016b). Progressive neural networks. *arXiv preprint arXiv:1606.04671v3*.

Teh, Y. W., Bapst, V., Czarnecki, W. M., Quan, J., Kirkpatrick, J., Hadsell, R., Heess, N., and Pascanu, R. (2017). Distral: Robust multitask reinforcement learning. *arXiv preprint arXiv:1707.04175v1*.