Actor Mimic: Deep Multitask and Transfer Reinforcement Learning: A Review

Ashutosh Jha

March 2017

1 Introduction

Actor-Mimic networks provide with a framework for training multi-tasking agents with the help of task-specific expert networks such that the complexity of the AMN is nearly the same as the expert networks. It borrows ideas from Hinton's paper on Knowledge Distillation and applies it to deep reinforcement learning agents to train a student network which learns from the expert networks. The way the AMN framework is set up, it allows us to use the features learnt by AMN as a pre-training for Transfer Learning for a novel but a related task.

2 Details of Actor-Mimic Networks

Given a set of source games $S_1, S_2, \cdots, S_N$ the first goal is to obtain a single multitasking agent which can play any source game at near-expert level. To do this, we train a set of task-specific expert DQNs $E_1, E_2 \cdots, E_N$ corresponding to each game. The objective of training the AMN is then split into 2 objectives, Policy Regression objective and the Feature Regression objective.

2.1 Policy Regression Objective

The policy regression objective for task 'i' is defined as:

$$\mathbb{L}^i_{\text{policy}} = \sum_{a \in A_{E_i}} \pi_{E_i}(a|s) \log \pi_{\text{AMN}}(a|s;\theta)$$

i.e. cross entropy loss of the policy of AMN w.r.t. that of the expert DQN for the task i. A squared loss of Q-values of AMN and the expert DQN was not used because the Q-values could vary widely over games and thus a policy of the expert DQN was defined using the softmax of Q-values:

$$\pi_{E_i}(a|s) = \frac{e^{\tau^{-1} Q_{E_i}(s,a)}}{\sum_{a' \in A_{E_i}} e^{\tau^{-1} Q_{E_i}(s,a')}}$$

where A_{E_i} is the action space used by the expert DQN. To acquire training data, sampling can be done either from the expert network or the AMN action outputs to generate the trajectories used in the loss. Empirically, authors report that sampling from the AMN while it is learning gives the best results. It is however proved in the paper that in either case of sampling from the expert or AMN as it is learning, the AMN will converge to the expert policy using the policy regression loss, at least in the case when the AMN is a linear function approximator.

2.2 Feature Regression Objective

Let $h_{\text{AMN}}(s)$ and $h_{E_i}(s)$ be the hidden activations in the feature (pre-output) layer of the AMN and i^{th} expert network computed from the input state s, respectively. The dimensions of the two need not be same. We define a feature regression network $f_i(h_{\text{AMN}}(s))$ that, for a given state s, attempts to predict the features $h_{E_i}(s)$ from $h_{\text{AMN}}(s)$. f_i can now be trained using the following loss:

$$\mathbb{L}^i_{\text{FeatureRegression}} = ||f_i(h_{\text{AMN}}(s;\theta);\theta_f) - h_{E_i}(s)||^2$$

A justification for this objective is that if we have a perfect regression from multitask to expert features, all the information in the expert features is contained in the multitask features. Empirically we have found that the feature regression objective's primary benefit is that it can increase the performance of transfer learning in some target tasks.

2.3 Actor-Mimic Objective

The AMN's objective function can be defined as follows:

$$\mathbb{L}^i_{\text{Actor-Mimic}} = \mathbb{L}^i_{\text{policy}} + \beta * \mathbb{L}^i_{\text{FeatureRegression}}$$

Intuitively, the loss can be thought of as follows: Think of the policy regression objective as the expert network telling the AMN how they should act, while the feature regression objective is analogous to the expert telling the AMN why it should act that way.

3 Actor-Mimic as a pretraining to Transfer Learning

To use AMN as a pre-training to transfer learning, use the weights of AMN as an instantiation for a DQN that will be trained on the new target task. The pretrained DQN is then trained using the same training procedure as the one used with a standard DQN. Pretraining can be seen as initializing the DQN with a set of features that are effective at defining policies in related tasks. This will however work only if there are similarities between the new task and the set of old tasks.

4 Convergence Properties of Actor-Mimic

The authors show convergence of AMN training for the case where the output of the AMN is linear in features(which is alright because the DQN's are linear in the features). They also propose a performance guarantee which they claim is what is seen empirically.

5 Experiments with Actor-Mimic

In case of multitasking experiments, the β is set to 0. One result that was observed during training is that the AMN often becomes more consistent in its behaviour than the expert DQN, with a noticeably lower reward variance in most games.The AMN had the worst performance on the game of Seaquest, which was a game on which the expert DQN itself did not do very well. It is possible that a low quality expert policy has difficulty teaching the AMN to even replicate its own behaviour.

In the transfer experiments, it was empirically found that larger networks transferred knowledge much more easily than small networks, so the AMN architecture was made more complex. The results in Breakout and Video Pinball demonstrate that the policy regression objective alone provides significant positive transfer in some target tasks. The reason for this large positive transfer might be due to the source game Pong having very similar mechanics to both Video Pinball and Breakout. In particular, multitask pretraining for Robotank even seems to slow down learning, providing an example of negative transfer. This is because none of the previous tasks were similar to Robotank and thus caused negative transfer.

6 Drawbacks of the paper

The prime drawback of this paper is that you still need to train the expert networks which is a very time-consuming task (apart from the training of the student AMN). As a result, the training is also not online in fashion. There can be instances of destructive interference due to the various expert networks which was also not addressed in the paper. Ultimately, though it does provide with a theoretical guarantee for convergence, it does not give any theoretical guarantees on performance.