Expert-augmented actor-critic
for Vizdoom and Montezuma's Revenge

Michał Garmulewicz, Henryk Michalewski, Piotr Miłoś
University of Warsaw
mg304742@students.uw.edu.pl

Abstract

We propose an expert-augmented actor-critic algorithm, which we evaluate on two simulated environments with sparse rewards: Vizdoom and Montezuma's Revenge. On Montezuma's Revenge, an agent trained with this algorithm achieves higher average reward than any previous approach, consistently scoring results above 8000 points and in some experiments solving the first world. In the case of Vizdoom, the agent learns to navigate a complicated maze in a scenario which is too difficult to be solved by model-free algorithms not augmented by expert data.

Introduction

Deep reinforcement learning has shown impressive results in simulated environments. However, as the cost of random exploration increases rapidly with the distance of rewards, current approaches often fail when rewards are sparse. This inhibits using reinforcement learning methods in real-world applications. Take as an example robotics, where in many cases rewards are calculated once a task is completed and thus are binary and sparse. The situation is aggravated if no simulation is available, making sample efficiency a key factor of success.

One way to improve the efficiency of exploration is to utilize expert data. The standard behavioral cloning often suffers from compounding errors when drifting away from the supervisor's demonstrations. While this can be mitigated by iterative methods like DAgger, the cost is cumbersome data collection process. In recent [1] authors analyze performance of behavioral cloning on Atari 2600 games. In the challenging example of Montezuma's Revenge their method reaches on average only 575 points despite being trained on demonstration trajectories that score 30 000 points.

Our approach is based on the Actor-Critic using Kronecker-Factored Trust Region (ACKTR) algorithm [3]. This algorithm uses natural gradient techniques to accelerate the gradient ascent optimization by changing parameters in the direction that minimizes the loss with respect to small step in the distribution of network output (in our case policy), as opposed to small step in the parameter space metric. Natural gradient approaches proved to be successful in increasing speed and stability of learning. We modify ACKTR so that it utilizes expert data. We believe that expert data guides agent's exploration. In our evalutation, on Montezuma's Revenge and Vizdoom environments this substantially accelerates the learning process.

Expert-augmented ACKTR

Our modification of the ACKTR algorithm introduces a new term $L_t^{\text{expert}}(\theta)$ to the the loss function:

$$L_t(\theta) = \underbrace{\text{adv}_t \log \pi_\theta(a_t|s_t) + \frac{1}{2}(R_t - V_\theta(s_t))^2}_{L_t^{\text{A2C}}(\theta)} + \lambda_{\text{expert}} \underbrace{\text{adv}_t^{\text{expert}} \log \pi_\theta(a_t^{\text{expert}}|s_t^{\text{expert}})}_{L_t^{\text{expert}}(\theta)}$$

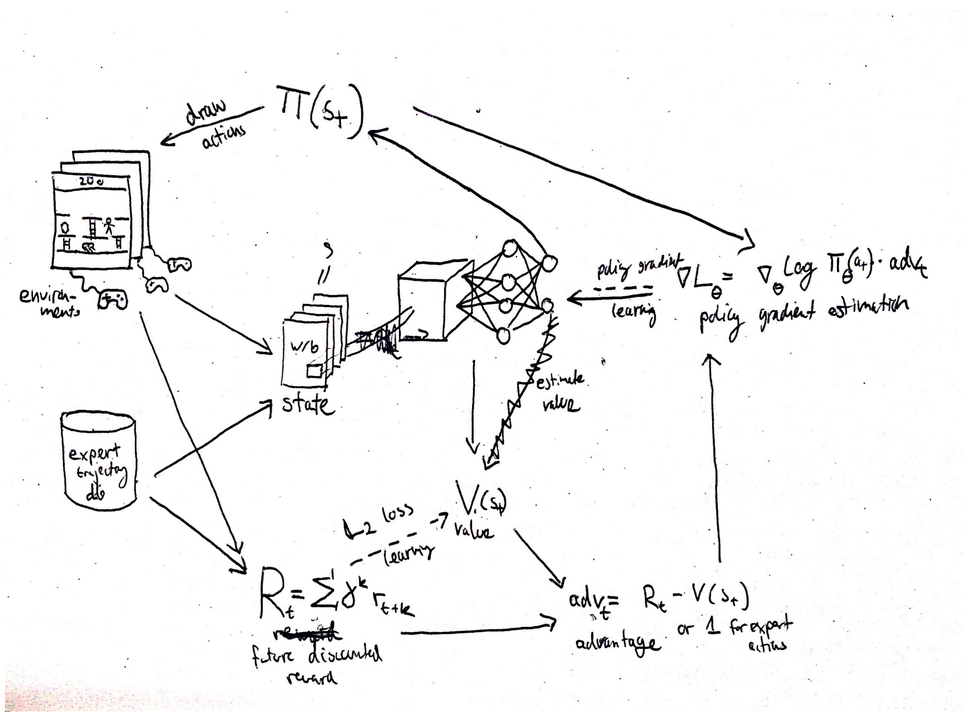

Figure 1: Visual representation of the algorhitm.

The expert data is sampled from a fixed dataset of rollouts achieving high rewards. We consider 3 variants of the expert advantage estimators: *reward:* $\text{adv}_t^{\text{expert}} = \sum_{s\geq 0} \gamma^s r_{t+s}^{\text{expert}}$ *actor-critic:* $\text{adv}_t^{\text{expert}} = \left[\sum_{s\geq 0} \gamma^s r_{t+s}^{\text{expert}} - V_\theta(s_t)\right]_+$, where $[x]_+ = max(x, 0)$ and *simple:* $\text{adv}_t^{\text{expert}} = 1$. Expert data is not used in ACKTR estimation of inverse Fisher, but still during the Kroncker optimization step g_{expert} is projected to the natural gradient direction.

Data: Parameter vector θ;
Dataset of expert transitions $(s_t^{\text{expert}}, a_t^{\text{expert}}, s_{t+1}^{\text{expert}}, r_t^{\text{expert}})$
for $iteration \leftarrow 1$ **to** *max steps* **do**
  for $t \leftarrow 1$ **to** T **do**
    Perform action a_t according to $\pi_\theta(a|s_t)$
    Receive reward r_t and new state s_{t+1}
  end
  for $t \leftarrow 1$ **to** T **do**
    Compute discounted future reward: $\hat{R}_t = r_t + \gamma r_{t+1} + ... + \gamma^{T-t+1} r_{T-1} + \gamma^{T-t} V_\theta(s_t)$
    Compute advantage: $\text{adv}_t = \hat{R}_t - V_\theta(s_t)$
  end
  Compute A2C loss gradient $g_{A2C} = \nabla_\theta \sum_{t=1}^{T} \left[-\text{adv}_t \log \pi_\theta(a_t|s_t) + \frac{1}{2}(\hat{R}_t - V_\theta)^2\right]$
  Sample mini batch of k expert state-action pairs
  Compute expert advantage estimate $\text{adv}_t^{\text{expert}}$ for each state-action pair.
  Compute expert loss gradient $g_{\text{expert}} = \nabla_\theta \frac{1}{k} \sum_{i=1}^{k} \text{adv}_i^{\text{expert}} \log \pi_\theta(a_i^{\text{expert}}|s_i^{expert})$
  Update ACKTR inverse Fisher estimate.
  Plug in gradient $g = g_{A2C} + \lambda_{\text{expert}} g_{\text{expert}}$ into ACKTR Kronecker optimizer.
end

Algorithm 1: Expert-augmented ACTKR

Results

Montezuma's Revenge

We use our Expert-augmented ACKTR algorithm to train an agent playing Montezuma's Revenge. We choose future discounted reward as expert advantage estimate, conjecturally this excludes suboptimal expert actions leading to agent's loss of life.

Approach	Mean Score
Expert aug. AC	6560 @ 200Mfr
DQfD	4740 @ 200Mfr
Beh. clon.	575
Ape-X DQfD	29,384 @ 200Mfr

Table 1: Comparison of performance on Montezuma's Revenge

The agent reaches the average 6560 points, median 8000 after training on 200M frames, which is more sample efficient than most of previous methods, such as DQfD [1], but less efficient then the newest Ape-X DQfD [2], see Table 1. Our agent consistently explores most of the rooms of the first world. In videos we present a selection of good evaluation rollouts. Standard non-augmented A2C and ACKTR fail to produce any decent results in Montezuma's Revenge. Further, supervised learning the expert data yields average score of 570 in [1].

Experiments with expert state resets We observed that often the trained actor is not able to get through a particularly tricky central room and therefore is stuck on score of 8000. To resolve this issue we run experiments with random starting points sampled from expert trajectories. We repeat the experiment with two random seeds. The results of this experiments can be seen in Figure 4. Interestingly, the agent trained this way is often able to solve the whole first world, scoring 44 000 points, see videos. The drawback is that training is less stable, the reason of that is yet to be investigated.

Figure 2: Expert restart training generates strong performance.

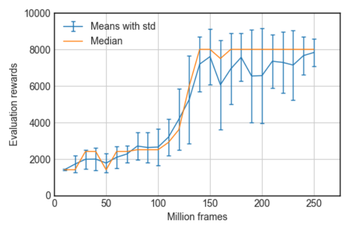

Figure 3: Evaluation without expert restarts.

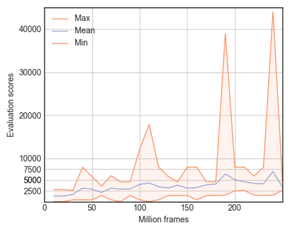

Figure 4: Evaluation with restarts.

Vizdoom

We experiment with a deterministic version of the Vizdoom MyWayHome environment, in which the agent is always spawned in the room that is furthest to reward, see Figure 6. This version is particularly hard for non-expert augmented methods, as random exploration is extremely unlikely to find any reward. In this case we use simple expert advantage estimate, which is equivalent to supervised "classification" of expert advantage.

Expert-augmented ACKTR, Algorithm 1, is very efficient in this setting, achiveing perfect performance in just 5 million frames, correspoding to only one hour of training, see videos. In comparison, both behavioral cloning of expert transitions and vanilla actor-critic fail to learn anything useful.

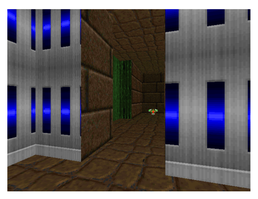

Figure 5: Screenshot from Vizdoom MyWayHome environment.

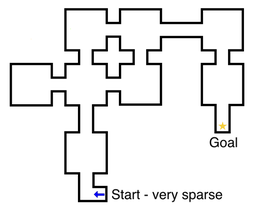

Figure 6: Map of Vizdoom MyWayHome environment.

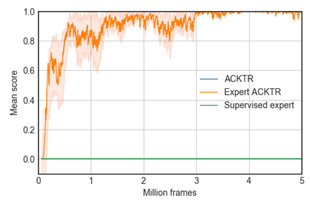

Figure 7: Vizdoom. Mean training rewards for 2 different random seeds.

Conclusions & Forthcoming research

Based on experimental results we hypothesize that the presented algorithm is a practical method of getting good performance in cases when multiple interactions with environment is possible and good quality expert data exists. It could be particularly useful in settings when neither supervised learning from expert data nor random exploration yield good results, such as in Montezuma's Revenge.

We leave the following extensions to future work:

1. Running Expert-augmented ACKTR on the rest of classic Atari environments.
2. Measuring the training efficiency dependence on the quality of expert data.
3. Experiments in simulated robotics, where are sparse rewards are common - e.g. currently ongoing work on simulated driving in CARLA

Furthermore, for the sake of simplicity we did not use importance sampling of expert actions (which in principle should be used as expert data is off-policy). Implementing this might be challenging as it requires estimating distribution of the expert policy only from trajectories. On the other hand algorithm using importance sampling would be consistent with the theory behind ACKTR and thus potentially more efficient.

References

[1] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Gabriel Dulac-Arnold, Ian Osband, John Agapiou, Joel Z. Leibo, and Audrunas Gruslys. Deep q-learning from demonstrations, 2017.

[2] Tobias Pohlen, Bilal Piot, Todd Hester, Mohammad Gheshlaghi Azar, Dan Horgan, David Budden, Gabriel Barth-Maron, Hado van Hasselt, John Quan, Mel Vecerík, Matteo Hessel, Rémi Munos, and Olivier Pietquin. Observe and look further: Achieving consistent performance on atari. *CoRR*, abs/1805.11593, 2018.

[3] Yuhuai Wu, Elman Mansimov, Shun Liao, Roger Grosse, and Jimmy Ba. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation, 2017.