CS7011 : Topics in Reinforcement Learning
Summary of Deep Q-Learning from Demonstrations : Hester, Vecerik, Pietquin, Lanctot
By Sahana Ramnath : EE15B109

1. Abstract

(Hester et al., 2017) proposes a novel algorithm *DQfD* that uses small amounts of demonstration data by an expert to massively accelerate the agent's learning and automatically assesses and uses the necessary ratio of demonstration data while learning by using a prioritized replay mechanism. This algorithm is especially useful in real world cases where the simulator is flawed(and so the agent must learn in the real world which is difficult), but there exists data of a previous controller(human or machine) operating the system whose performance was reasonably good. DQfD is tested on Atari games and is compared with other related algorithms.

2. Overview of the Algorithm

DQfD makes use of the demonstration data to pretrain the agent so that it performs well from the start of learning itself(like a jump start), and then continue improving using its self-generated data from the environment. Initially, the agent trains solely on the demonstration data using a combination of TD and supervised losses. The latter allows the agent to learn to imitate the demonstrator, whereas the former allows it to learn a self-consistent value function from which it can continue learning with RL. After this, the agent starts interacting with the domain with this learnt policy; it updates its network with a fixed ratio of both demonstration and self-generated data; the chosen ratio is critical to improve the agent's performance. It uses a prioritized replay mechanism to automatically control this ratio.

3. Related Work

This algorithm can be considered partially as an imitation learning algorithm, where the main concern is to match the performance of the demonstrator. DQfD has been built on and compared with various IL algorithms.

- RL with Expert Demonstrations(RLED) : Similar to this, DQfD combines TD and classification losses in a batch algorithm in a model free setting; and in contrast, DQfD is pretrained on the expert data and the self-generated data used for training grows over time.
- Alpha Go : Also pretrains on demonstration data like DQfD. However, it uses a model and rollouts whereas DQfD is a model free Q-Learning algorithm.
- Human Experience Replay(HER) : Here, agent samples from a replay buffer that is a mix of demonstration and agent data, similar to DQfD, but doesn't pretrain or use a supervised loss. Its alternative approach Human Checkpoint Replay requires the ability to set the environment's state, whereas DQfD doesn't require it and still performs better.
- Replay Buffer Spiking(RBS) :Replay buffer is initialized with demonstration data, but there's no pretraining and the demonstration data is not kept permanently.
- Accelerated DQN with Expert Trajectories(ADET) : It combines TD error and classification losses and uses a trained DQN to generate expert trajectories; it uses CE loss whereas DQfD uses large margin loss. However, there is no pretraining.

4. The Algorithm

The DQfD agent aims to learn as much as possible from the demonstration data before interacting with the real system; the goal is to learn to imitate the demonstrator with a value function that satisfies the Bellman equation, so that it can be updated with TD updates once the agent starts interacting with the environment. During pretraining, the agent samples mini batches from the demonstration data and updates the network by applying four losses: 1-step double Q-learning loss, n-step double Q-learning loss(forward view), supervised large margin classification loss, and L2 regularization loss on the network weights and biases. The Q-learning losses ensure that the network satisfies the Bellman equation and can be used as a starting point for TD learning. Now, the demonstration data will cover only a narrow part of the state space and so there is no data for grounding the unseen state-action values to a realistic value. Using just Q-learning updates for these would update the network towards the highest of these ungrounded variables, whereas if the agent uses just supervised learning there would be nothing constraining their values and the network would not satisfy the Bellman equation.Hence, a large margin classification loss is used : $J_E(Q) = max_{a \in A}[Q(s,a)+l(a_E,a)]-Q(s,a_E)$, where a_E is the action the demonstrator took in state s and $l(a_E,a)$ is the margin function that is 0 when $a = a_E$ and positive otherwise. This ensures the values of other actions are atleast a margin lower than the value of the demonstrator's action. This loss grounds the values of the unseen actions to reasonable values, and makes the greedy policy induced by the value function imitate the demonstrator. The L2 regularization loss is applied to prevent overfitting on the relatively small demonstration dataset.The overall loss : $J(Q) = J_{DQ}(q) + \lambda_1 J_n(Q) + \lambda_2 J_E(Q) + \lambda_3 J_{L2}(Q)$. Once pretraining is complete, the agent starts adding self-generated data from the environment to its replay buffer D_{replay} till its full after which it starts overwriting. However, it never overwrites the demonstration data. Different small positive constants, ϵ_a and ϵ_b, are added to the priorities of the agent and demonstration transitions to control the relative sampling of demonstration versus agent data. All the losses are applied to the demonstration data in both phases, while the supervised loss is not applied to self-generated data ($\lambda_2 = 0$).

5. Experiments and Results

DQfD was evaluated on Atari games and comparison was done across : Full DQfD, PDD DQN(with n-step returns) and plain IL with no environment interaction. All 3 used the dueling state-advantage convolutional network architecture. The reward function was changed to $r_{agent} = sign(r).log(1 + |r|)$ to make the reward function used by the agents and human more consistent. DQfD outperformed/performed on par with all previously published results and humans on most of the games especially in Hero, Pitfall and Road Runner. DQfD is also compared with HER,RBS and ADET and it outperforms all of them. Having a supervised loss seems to be critical for success since

ADET and DQfD perform much better than the other two.

References

Hester, Todd, Vecerik, Matej, Pietquin, Olivier, Lanctot, Marc, Schaul, Tom, Piot, Bilal, Horgan, Dan, Quan, John, Sendonaris, Andrew, Dulac-Arnold, Gabriel, et al. Deep q-learning from demonstrations. *arXiv preprint arXiv:1704.03732*, 2017.