ICANN19
28th International Conference on Artificial Neural Networks

Deep Recurrent Policy Networks for Planning under Partial Observability

Zixuan Chen[1]*, Zongzhang Zhang[2]
[1] School of Computer Science and Technology, Soochow University, Suzhou, China
[2] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
Contact Author:* cczxqueen@gmail.com

Abstract

QMDP-net is a recurrent network architecture that combines the features of model-free learning and model-based planning for planning under partial observability. The architecture represents a policy by connecting a partially observable Markov decision process (POMDP) model with the QMDP algorithm that uses value iteration to handle the POMDP model. However, as the value iteration used in QMDP iterates through the entire state space, it may suffer from the "curse of dimensionality". Besides, as the policies based on the QMDP will not take actions to gain information, this may lead to bad policies in domains where information gathering is necessary. To address these two issues, this paper introduces two deep recurrent policy networks, asynchronous QMDP-net and ReplicatedQ-net, based on the plain QMDP-net. The former takes advantage of the idea of asynchronous update into the value iteration process of QMDP to learn a smaller abstract state space representation for planning. The latter partially replaces the QMDP with the replicated Q-learning algorithm to take informative actions.

Method

Asynchronous QMDP-net is a recurrent policy network that applies the idea of asynchronous update into the QMDP planner to improve the plain QMDP-net for planning better under uncertainty. Asynchronous QMDP-net effectively reduces the possibility of meaningless sweeping that happens in the value iteration process of QMDP and alleviates the "curse of dimensionality". Since QMDP makes use of the underlying fully observable MDP to perform the computation of the final Q-values of POMDP, in asynchronous QMDP-net, we choose to use the Bellman error to favour the sampling of certain states in every round of update process so as to learn a much smaller abstract state set for more efficient planning. The Bellman error is the absolute value of the difference between the state value obtained before and after one round of value iteration. According to the defined state importance, states to be updated in each round of asynchronous update are sampled with a threshold, that is, states whose Bellman errors are greater than the threshold will be sampled, whereas others will not be sampled and their values remain unchanged. Based on this threshold, the network can effectively distinguish between the less important states and the more important states and successfully sample states with relatively higher priorities during each iteration, this allows the asynchronous updates to be performed more reasonably.

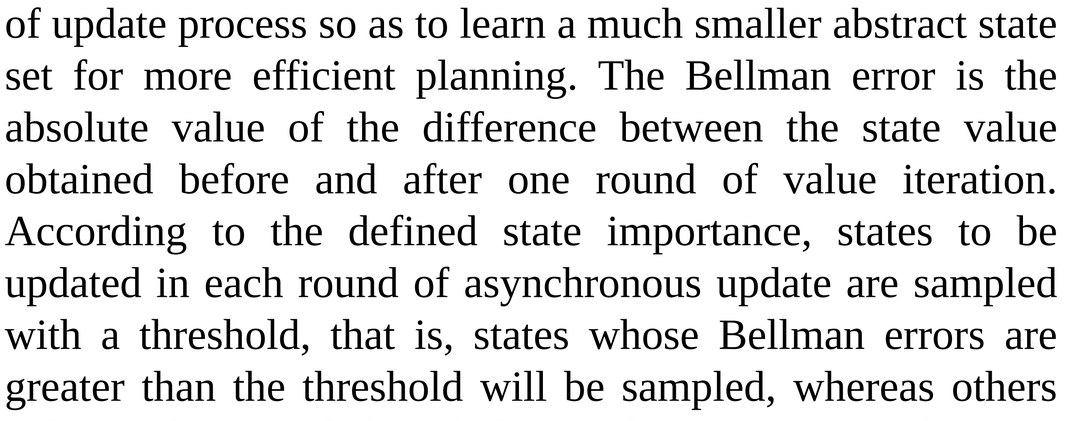

Figure 1. The Asynchronous QMDP planner module in asynchronous QMDP-net.

Same as QMDP-net and asynchronous QMDP-net, ReplicatedQ-net is also a recurrent policy network that is obtained by partially replacing the QMDP algorithm used in QMDP-net with the value update rule of replicated Q-learning. It is worth noting that, in order to prevent the algorithmic sophistication from increasing the difficulty of learning, instead of directly using the replicated Q-learning to solve POMDPs, in ReplicatedQ-net, we masterly combine the value update rule used in replicated Q-learning with the value iteration algorithm to achieve better planning performance while reducing the difficulty of learning.

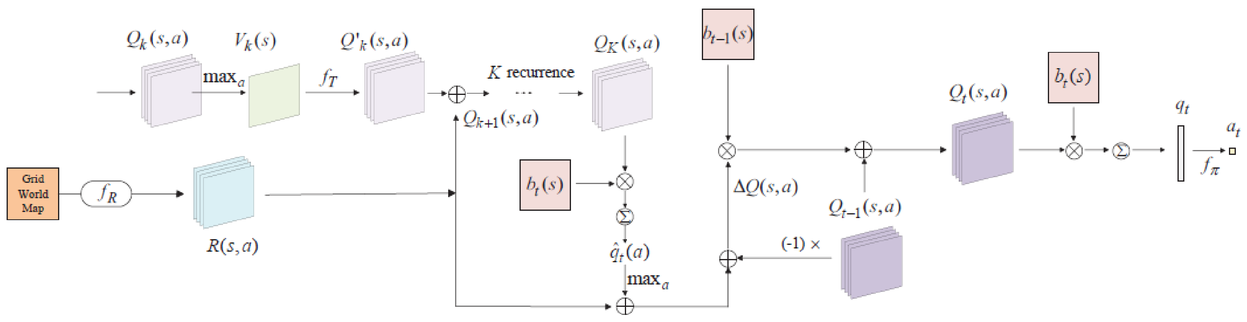

Figure 2. The ReplicatedQ planner module of ReplicatedQ-net

Experiments

The specific experimental environment is set up as a robot learning to navigate in partially observable grid worlds. The robot possess a map corresponding to the current grid world environment, and it has a belief over the initial state, but does not know where the exact initial state is. For the robot, only the local information around it will be observed, however, these local observations are ambiguous, which are insufficient for the robot to determine where it is the exact state.

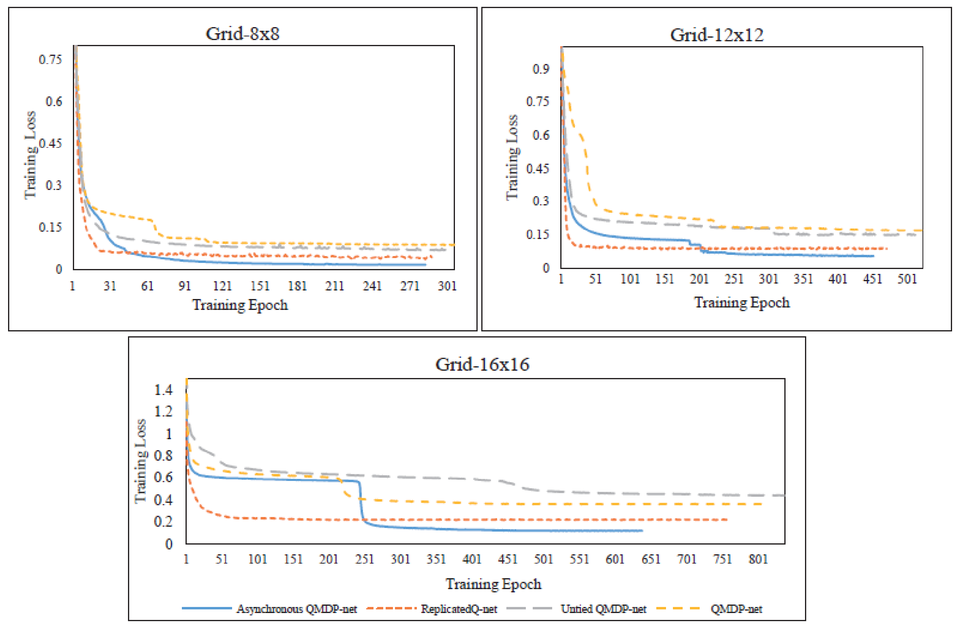

Figure 3. Comparison of training performance of the recurrent policy networks.

Grid	Asy. QMDP-net SR	Asy. QMDP-net Time (s)	ReplicatedQ-net SR	ReplicatedQ-net Time (s)	QMDP-net SR	QMDP-net Time (s)	Untied QMDP-net SR	Untied QMDP-net Time (s)
8×8	100.00	5.5	100	5.7	99.0	5.6	100	5.8
16×16	96.00	24.3	92.00	24.8	91.00	24.3	91.00	24.1
18×18	95.00	32.6	91.00	32.3	88.00	34.1	82.00	38.8
S-18×18	94.00	41.2	91.00	42.4	86.00	50.6	81.00	51.9

Table 1. Comparison of testing performance of the recurrent policy networks.The policy networks used for testing are trained under Grid-8x8.

Grid	Asy. QMDP-net SR	Asy. QMDP-net Time (s)	ReplicatedQ-net SR	ReplicatedQ-net Time (s)	QMDP-net SR	QMDP-net Time (s)	Untied QMDP-net SR	Untied QMDP-net Time (s)
16×16	98.00	24.0	97.00	24.1	86.00	33.5	78.00	42
32×32	98.00	110.5	95.00	104.4	58.00	253.9	47.00	339.7
S-32×32	97.00	117.6	86.00	188.8	59.00	290.8	22.00	377.7
64×64	92.00	1029.5	85.00	972.8	15.00	3207.6	-	-

Table 2. Comparison of testing performance of the recurrent policy networks.The policy networks used for testing are trained under Grid-16x16.