Front.Comput.Sci.
DOI

SupplementaryFile

Improving Deep Reinforcement Learning by Safety Guarding Model via Hazardous Experience Planning

Pai PENG [1], Fei ZHU (✉)[1], Xinghong LING [1], Peiyao ZHAO [1], Quan LIU [1]

1 School of Computer Science and Technology, Soochow University, Suzhou Jiangsu 215006, China

1 Experiment and analysis

This section describes the platforms and the experiments'parameters to test the proposed method in some strategic games from Atari 2600 on the Gym platform. DQN, DuDQN, DRQN, and A3C are used for comparison. The HE-DRL is based on the deep reinforcement learning algorithm and adapts the staged risk sample planning mechanism proposed in this paper. Different models shared the same set of parameters in the training process. Finally, according to the experimental results, the advantages and disadvantages of the HE-DRL and its application range are analyzed.

1.1 Description of experimental platform

To effectively evaluate the model's performance, we chose Atari 2600 on the Gym platform for the experiment. A brief introduction to the games is provided in Table.1.

Table 1 Brief introduction to Atari 2600 games

Name	Number of actions	Brief Description
Seaquest	18	survive as possible
Beamrider	9	shoot enemy
SpaceInvaders	6	avoid warships

1.2 Experimental parameter settings

Specific techniques were introduced to reduce instability that occurred during the training process. The agents selected actions based on their Q values. Frame-skipping technology was adopted to reduce calculation and four parameters were used related to frame skipping. The agent took actions for every four frames. If a terminal state was reached during frame skipping, the total reward was returned.

To improve the algorithm'stability and reduce the deviation of Q-value caused by large fluctuations, the loss function was cut into segments. If the absolute value of the loss was in the interval $[-1, 1]$; otherwise, the absolute-value function was used. In the experiment, all positive rewards were set to 1 and negative rewards to -1. Zero rewards remain unchanged to facilitate comparison among the algorithms.

To compare the performance of the original DQN, DuDQN, DRQN, and A3C with those equipped with hazardous experience planning, we took the same settings. The same pretreatment were applied and all of them used a three-layer convolutional neural network with the same parameters. Root mean-square propagation (RMSProp) was used to update the network parameters. The coefficient of momentum was set to 0.95 and the discount factor to 0.99. In the first 50,000 steps of training, the agent generated enough samples randomly and stored them in the general experience replay pool D_u.

Throughout the three-stage training, the greedy strategy $\varepsilon-greedy$ gradually changed in value from 0.9 to 0.1 owing to a gradual improvement of the model, and the value of K for the planning of hazardous experience increased from 3 to 9 and the network update parameter α decreased from 0.005 to 0.00025.

1.3 Results and analysis

In reinforcement learning, cumulative rewards are often used as criteria for judging the merits of the strategy. Due to the long training period of deep learning and the exten-

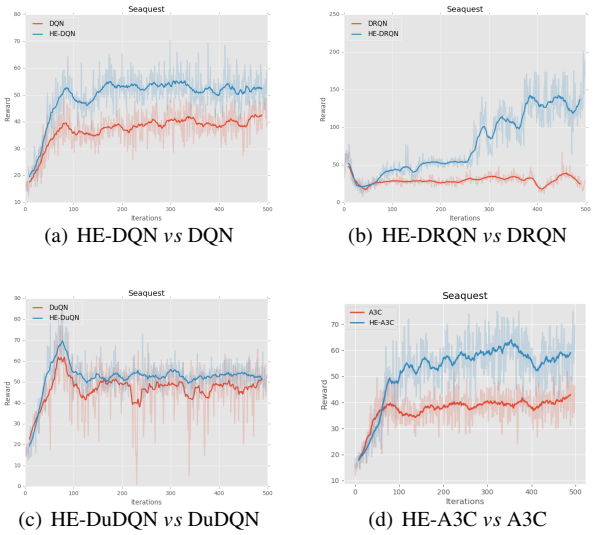
Received February 2, 2021; accepted September 17, 2021

E-mail: *zhufei@suda.edu.cn*

sive training data, the training effect is unstable. Usually, the cumulative reward obtained by one episode is used as the evaluation standard. The merits and demerits of the model are evaluated by the staged statistics of each episode reward size.

It is worth noting that the gym package provides several resurrection opportunities after the game fails. After the options are used up, one episode will be ended. The experiment in this paper mainly proves the effect of adding security to deep reinforcement learning. Samples are collected every several steps before each game fails. Therefore, a plot in reinforcement learning can be redefined from the begging to the agent's first failure.

In the experiment, the DQN, DuDQN, DRQN, and A3C network models were selected as the original network. The hazardous experience planning was applied to these three network models, denoted as HE-DQN, HE-DuQN, HE-DRQN, and HE-A3C, respectively.



(a) HE-DQN *vs* DQN (b) HE-DRQN *vs* DRQN

(c) HE-DuDQN *vs* DuDQN (d) HE-A3C *vs* A3C

Fig. 1 Comparison of average rewards between HE-DRL and other methods in Seaquest

1.3.1 Seaquest

Seaquest is a strategy game, in which agents get higher scores by continually drawing oxygen and avoiding obstacles. The HE-DRL performed well in shooting games and strategy games such as Seaquest, a game of underwater exploration with multiple objectives. Agents need to survive as long as possible to avoid hazardous goods and find oxygen supplies. In Seaquest, the game settings need to be connected with previous frames, which is more challenging. Fig. 1 compares the training results of three original networks (DQN, DuDQN, DRQN, and A3C) equipped with hazardous experience-planned networks (HE-DQN, HE-DuQN, HE-DRQN, and HE-A3C) in the Seaquest game. A total of 500 training episodes were operated, and each was set to 5000 steps. In the figure, the ordinate is the average reward per episode, and the horizontal coordinate is the number of training episodes. Strategic games are more difficult for agents, especially for DuDQN and DRQN, where the score may not converge. However, the HE-DRL focused on avoiding all dangers by utilizing previous hazardous experience-related information to ensure the agent'stability.

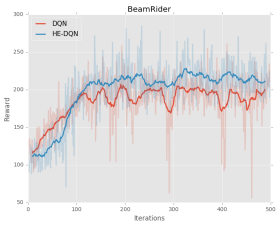
The result shows that the average reward for each episode of the HE-DRL was higher than those of the other three methods. In the Seaquest environment, HE-DQN, HE-DuDQN, HE-DRQN and HE-A3C based on the hazardous experience planning model proposed in this paper show improvement average reward compared to the original model, especially for HE-DQN, and HE-DRQN. The reason is that Seaquest is more complicated and enables more actions. In this case, the hazardous experience is significant. After the agent forms a hazardous experience plan, it can find a safe strategy in Seaquest faster, thus improving the survival rate and the final score.

1.3.2 BeamRider

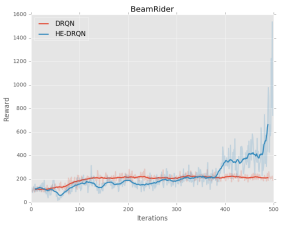
BeamRider is a shooting game, in which agents use shots to destroy obstacles for higher scores. Fig.2 compares the training results of the three original networks (DQN, DuDQN, DRQN, and A3C) with three hazardous experience-planned networks (HE-DQN, HE-DuQN, HE-DRQN, and HE-A3C) in the BeamRider game. A total of 500 training episodes were operated, and each was set to 5000 steps. In the figure, the ordinate is the average reward per episode, and the horizontal coordinate is the number of training episodes.

The BeamRider game contains nine actions. However, in the BeamRider environment, as the game proceeds, the enemy's bullets are getting denser, and the hazardous situation will be more complicated. Therefore, the general network model can hardly extract suitable action choices from the image, resulting in a lower score.

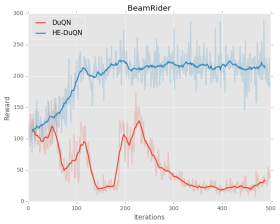
As can be seen from Fig.2, the average reward of the original network model in the BeamRider environment may cease to increase during the training. The initial network model can hardly learn effective strategies for the reason that in the BeamRider game environment. Images provide similar information, making it difficult for the original model to perform well. Despite this, the network average reward value using the hazardous experience planning model has been significantly improved, and the network model can capture detailed information, especially



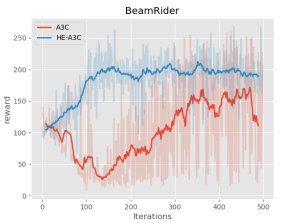
(a) HE-DQN *vs* DQN



(b) HE-DRQN *vs* DRQN

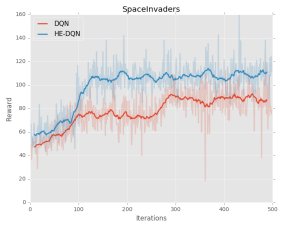


(c) HE-DuDQN *vs* DuDQN

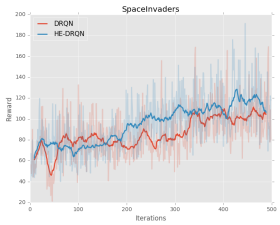


(d) HE-A3C *vs* A3C

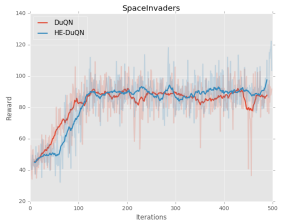
Fig. 2 Comparison of average rewards between HE-DRL and other methods in BeamRider



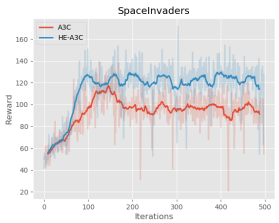
(a) HE-DQN *vs* DQN



(b) HE-DRQN *vs* DRQN



(c) HE-DuDQN *vs* DuDQN



(d) HE-A3C *vs* A3C

Fig. 3 Comparison of average rewards between HE-DRL and other methods in SpaceInvaders

hazardous states.

1.3.3 SpaceInvaders

In SpaceInvaders, agents manipulate the spacecraft to evade bullets using walls and defeat the enemy. In SpaceInvaders, the agent is at the bottom, and uses a city wall to avoid many enemy bullets. It fires shots and scores one point for each enemy killed until all enemies are dead. All evaded shots and ineffective shooting are awarded zero, and the game is suspended when the agent is shot.

Fig.3 compares the training results of the three original networks (DQN, DuDQN, DRQN, and A3C) with three hazardous experience-planned networks (HE-DQN, HE-DuQN, HE-DRQN, and HE-A3C) in the SpaceInvaders game. A total of 500 training episodes were operated, and each was set to 5000 steps. In the figure, the ordinate is the average reward per episode, and the horizontal coordinate is the number of training episodes.

As can be seen from Fig.3, the hazardous experience planning model can no longer accurately identify the critical state and the security state because the states of the SpaceInvaders game are very similar, resulting in limited performance improvement.

1.4 Analysis of the algorithm

The DQN considers only all empirical samples that are randomly sampled according to historical information without distinction. It explores and takes action using the $\varepsilon-greedy$ strategy. Because of inadequate information acquired in the game'suspension and the randomness of exploration, a high-reward action could not be easily reproduced.

The DuDQN is considered the dominant function of the agent's state so that the agent could obtain more information from different states. However, there was a lack of specific strategic choices when focusing on the dominant function of the state, and decision making was not sufficiently refined when more state information was available.

The DRQN that combines the DQN with LSTM fully resolves the memory limitation of the experience replay pool in the DQN algorithm, and relies on the complete state of the game. In shooting games such as SpaceInvaders, there are flying targets such as bullets, which have some partially observable Markov properties. The positions of enemy warships and shots were specified in the game, but bullet speed was not a parameter. The key to avoiding bullets and improving survival time was to know the bullet's velocity, that is, incomplete features and information concerning noisy states. The DRQN can use historical state information to solve information loss when encountering an unfinished state, but using the DRQN based on historical information H_t to H_{t+1} takes a long training time.

Because A3C uses an asynchronous method to generate sample data, it breaks the correlation of data without experience replay, ensures the network'stability, and has an excellent performance in the Arari 2600 game. However, in discrete action tasks, the actor-critic algorithm will produce convergence instability due to the maximum entropy regular term. Based on making full use of the hazardous experience sample pool, HE-DRL can make the estimation problem of reduced value function more stable.

The hazardous experience planning retains the advantage of the sample experience replay pool. Considering the

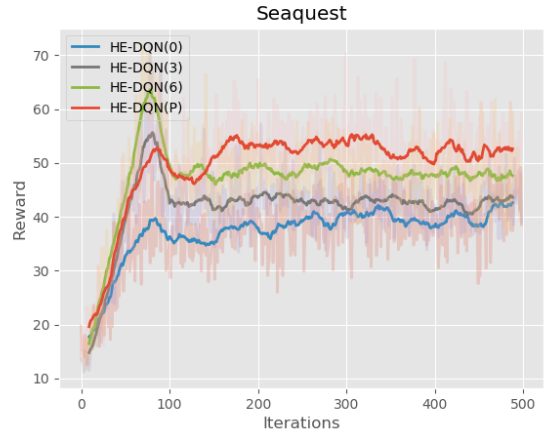
disconnect situation in terms of sample correlation, it adds a hazardous experience sample pool. Through multi-step planning of hazardous experience, the agent focuses on the state before failure in the training process to gain more information to avoid falling into danger again. The hazardous experience planning effectively improves the sensitivity of the agent to the hazardous state.

Agents with trained parameters are tested in the three games. Each game had three trial stages, and each trial lasted 80,000-time steps. Considering the fluctuation of rewards in different scenarios in the test stage, the standard deviation of rewards for the last 50 scenarios of the four models was calculated.

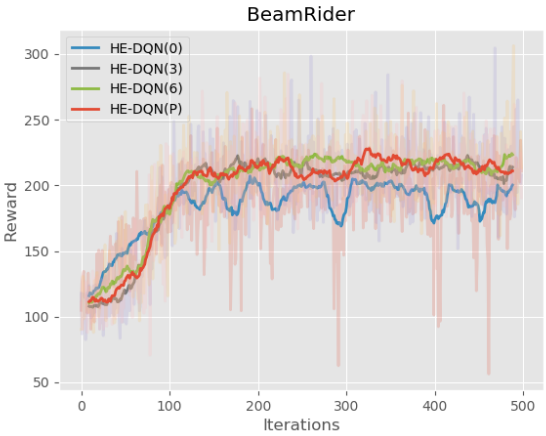
Besides, to prove the validity of the model of hazardous experience, we compare different values of M for planning in different games. In four models of the HE-DQN, M was fixed at zero, three, six, and phased at 0–6. Note that when the planning parameters were all zeroes, the HE-DQN degenerated into the classical DQN model. The results of training in terms of the average reward for each scenario of the specific model are shown in Fig. 4 where HE-DQN(P) represents more fine-grained planning for the hazardous experience.

Fig. 4 shows that in the three Atari games, it was more effective to plan for hazardous experience in stages than in fixed planning samples. It was more reasonable to explore hazardous experiences by using phased change planning. In the early stage of training, because the model had not been formed and the training results were not very stable, using small-scale hazardous experience planning is more appropriate. When the model was gradually created, it became necessary to improve the model with finer granularity. At this time, it is essential to increase the use of hazardous experience further. As shown in Fig. 4, the average reward for each scenario with a variable value of M was better than that of the classic DQN. By comparing the values of different planning parameters, we see that the variable model showed good stability and finally converged to the highest value. Therefore, the DQN model for hierarchical empirical programming with hazardous experience can significantly improve the algorithm'performance.

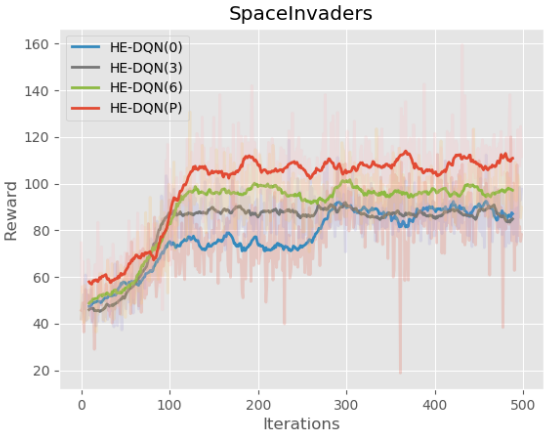
To prove the effectiveness of our algorithm, we also compare it with other secure deep reinforcement learning. As shown in Fig .5, the algorithm DDN adds a deep Q-network to the original network model to train the hazardous samples, uses the penalty term to describe the critical state, and uses the original network objective function and penalty term to calculate the objective function. We compared the number of rounds with the same step number and the same environment during the training process.



(a) Seaquest



(b) BeamRider

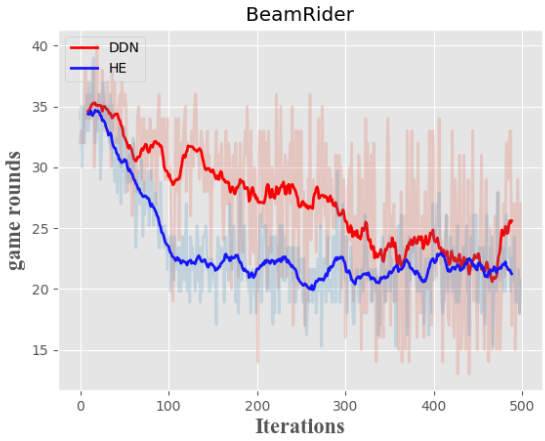


(c) SpaceInvaders

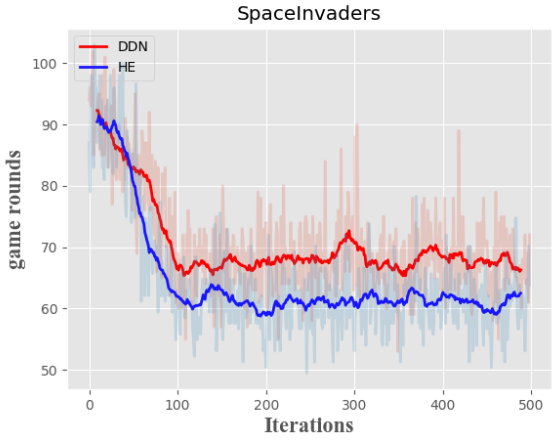
Fig. 4 The average rewards under different degrees of planning in three different games. HE-DQN(0) presents that the planning parameter M is fixed at zero which is degenerated into the classical DQN model. HE-DQN(3) and HE-DQN(6) presents M is fixed at 3 and 6 and HE-DQN(P) presents that the value of planning parameter M varies in stages

Under the same conditions, we can see from the figure that the HE algorithm (blue line) has played fewer rounds of the game. In other words, agents survive longer in each

(a) Seaquest



(b) BeamRider



(c) SpaceInvaders

Fig. 5 Number of game rounds with the same step number.Under the same number of game steps, the less rounds of the game, the longer the survival time of each game

round of the game, showing that the HE-DRL algorithm is feasible.