Improved Value Iteration Network for Path Planning

Shijia Chai [1, *]

[1]Department of Electronic Engineering, Tsinghua University, Beijing, 100084, China
[*]Corresponding author's email: chaisj15@tsinghua.org.cn

Abstract. In this study, a series of improved path planning algorithms are designed for path planning tasks in autonomous control based on deep reinforcement learning. The Value Iteration Network (VIN) is used to deal with the path planning problem. Origin VIN performs well on small size maps, but when it comes to a bigger size of map on test set, the success rate decreased. In order to solve the problem that origin VIN lacks long-distance multi-step planning ability on large maps and generalization ability is insufficient, a three-step improvement was made. First of all, in view of the inconvenient data flow and the disappearance of gradients caused by the network being too deep, the jump connection structure is used to obtain the deeper VIN, in which the accuracy of the experiment is improved. Secondly, with the purpose of solving the problem that the complexity of the model is greatly increased due to the deepening of the network, Batch normalization is used to obtain a new network with dueling architecture plus batch normalization layer, which further accelerates the convergence speed of the network. Third, to deal with the global path planning problem on the big map, the hierarchical network structure is adopted for hierarchical value iteration, and the Hierarchical Structure VIN is obtained. In Hierarchical Structure VIN, the long-term planning ability and generalization ability of the algorithm have been significantly improved, and the algorithm could figure out the large-scale and complex path planning problem.

Keywords: Path Planning Algorithm; Deep Reinforcement Learning; Deep Learning; Hierarchical Structure.

1. Introduction

Traditional underwater vehicles with cables are limited by cables. Therefore, cableless underwater vehicles, that is, autonomous underwater vehicles (AUVs), are gradually emerging. Artificial intelligence and other advanced computing technologies are integrated to achieve autonomous control. That is, real-time autonomous assessment, autonomous decision of current obstacle avoidance actions, and autonomous local or global path planning.

Path planning is a primary task of autonomous control for agents. Artificial intelligence technology, especially reinforcement learning algorithm, has brought new progress in path planning algorithm. And there are some traditional methods, such as Dijkstra algorithm [1].

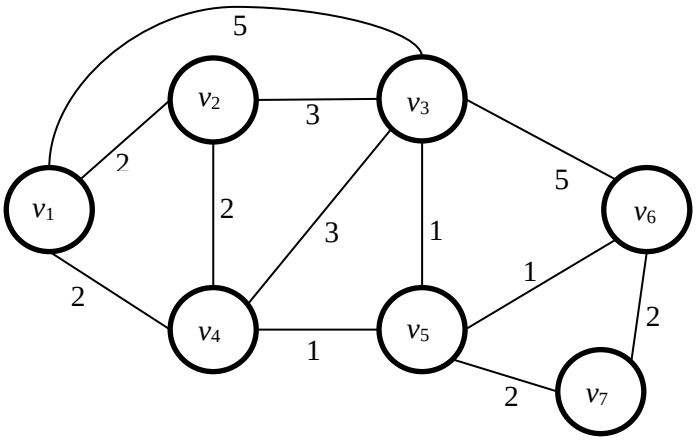


Figure 1. Dijkstra Algorithm.

But the traditional path planning algorithm has some problems. Because the separation of graph recognition and path planning, it is not an end-to-end model, the performance of the algorithm has bottlenecks [2]. And they have no learning ability, no generalization ability and no intelligent understanding of the problem.

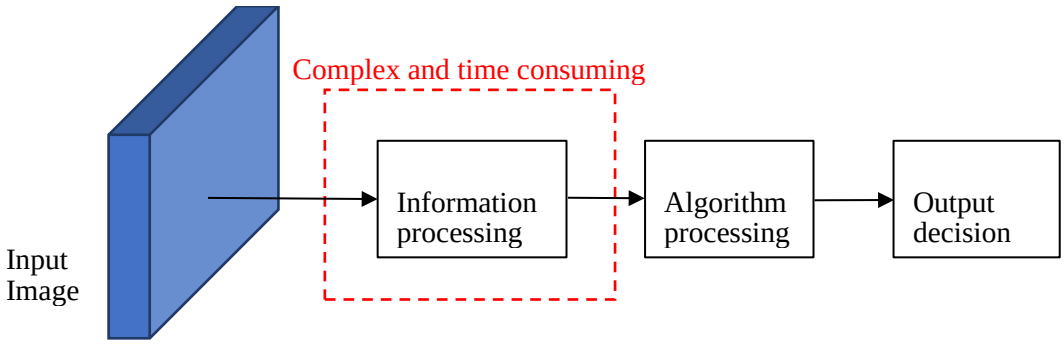


Figure 2. Traditional algorithm processing flow.

With the introduction of reinforcement learning algorithms, particularly the application of deep reinforcement learning algorithms, artificial intelligence has been able to solve many complex practical problems, such as, allowing computers to reach the same level as humans in Atari games. In the path planning task, the application of deep reinforcement learning algorithm can make it have an intelligent understanding of the problem, and can be applied to complex application scenarios.

Therefore, the intelligent algorithm on the basis of deep reinforcement learning has been performed. It hopes to get an end-to-end training and prediction model. Moreover, the model has the ability of learning and generalization. And also, it has intelligent understanding of the problem. In this work, a network that can effectively learn to plan is proposed, that is, value iteration network (VIN) [3].

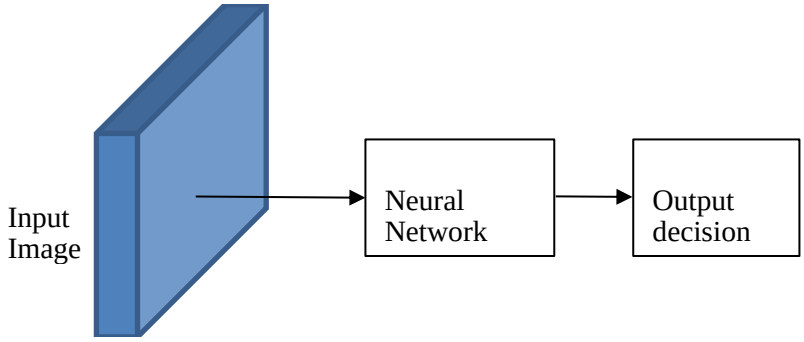


Figure 3. Algorithm processing flow based on Neural network.

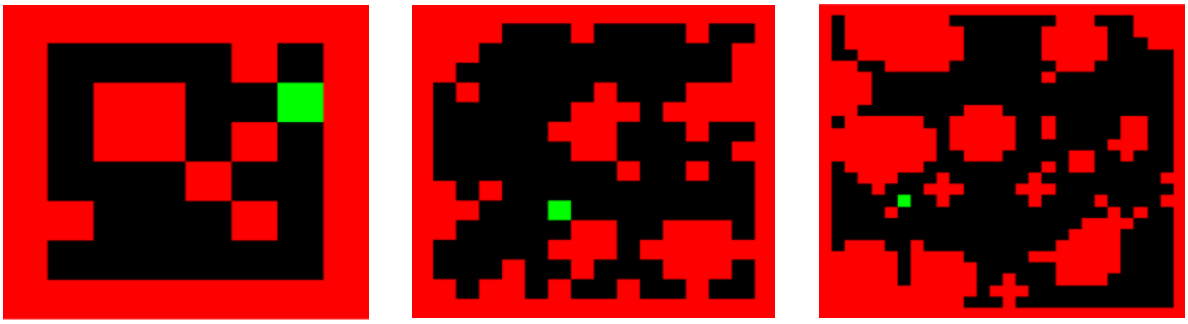
VIN captures the essential mathematical properties of reinforcement learning algorithms, the Markov Decision Process (MDP) [3]. The network cleverly combines the deep learning network structure with the Markov decision process, and corresponds to a series of mathematical operations and network structures such as convolution and pooling. When the data passes through the network structure, it is equivalent to performing mathematical operations and operations on the data. Therefore, the network structure has strong mathematical interpretability, and the theoretical support is very reliable [3].

In the current study, a deep reinforcement learning algorithm based on VIN is adopted for dealing with the path planning task, and a series of improvements are made on the basis of the original VIN. The improvements make the algorithm perform better in terms of data flow and network generalization.

2. Methods

2.1. Experimental environment

The maps used in the experiments are two-channel images. Among them, one channel records the position of the goal, and the other records the information of the background obstacle. The experimental map display of three sizes is shown in Figure 4:



(a) 8×8 (b) 16×16 (c) 28×28

Figure 4. Maps of different sizes.

In Figure 4, the black area is the barrier-free area; the red area is the obstacle area, which needs to be avoided; the green goal is the target position. When the agent moves freely in the unobstructed area, 8 neighbourhoods around it are reachable, as shown in Figure 5.

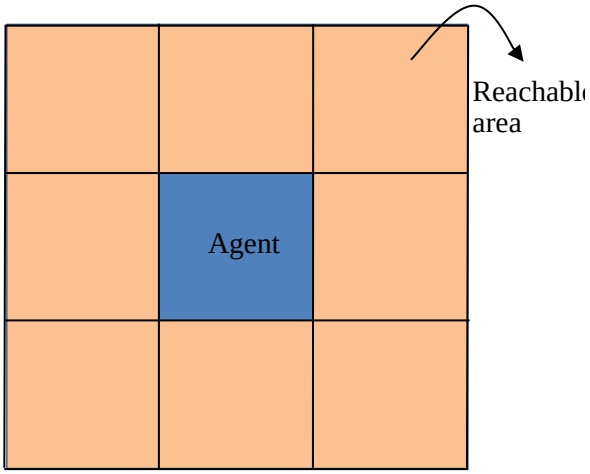


Figure 5. The active area of the agent.

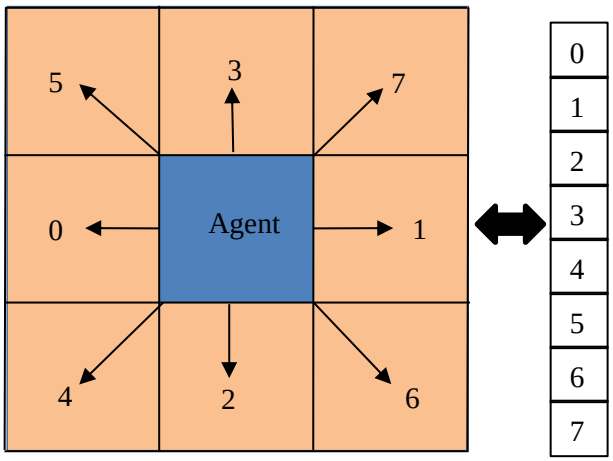


Figure 6. Correspondence diagram of vector and action.

The images of the two channels together constitute the input image of the neural network. After entering the value of the coordinates of the starting position of the agent, the complete input data of the neural network is formed.

Another feature of VIN is that it introduces an attention mechanism. In fact, the size of the experimental map is larger. For the agent to make a decision in a certain state, it only needs to pay attention to the local information near the current state, and it is not essential to focus on all the information of the whole image. Therefore, using the attention mechanism, only the value of the state position of the agent is passed in, which can exclude secondary irrelevant information as much as possible, and improve the accuracy of the agent's local decision-making.

After the data is subjected to the attention mechanism, the output results are transformed through the fully connected layer to obtain the last 8 outputs. These 8 outputs are to measure the value of the 8 actions that the agent may take in the unobstructed area, as shown in Figure 6. The agent takes the action represented by the highest value number in the output, moving from one state to the next. By taking a step-by-step action, and after a multi-step serialized decision, the entire path planning can be completed.

2.2. Dataset

In this experiment, expert samples are used for supervised learning. The loss function is defined using the more traditional cross-entropy loss. Expert samples are used to speed up training. For maps of every size, the number of samples in the training set and test set is presented in Table 1.

Table 1. Number of dataset samples

Map size	*Training set*	*Test set*
8×8	77760	12960
16×16	776440	129440
28×28	4510695	751905

2.3. Algorithm

2.2.1. Markov decision process (MDP). Most problems in reinforcement learning can be described using MDP. There are three basic elements in MDP: agent, environment and state. The agent interacts with the environment, the agent takes action in the environment, reaches the next state, as well as receives the reward fed back from the environment. The environment gives the corresponding reward in accordance with the action made by the agent. The whole goal of reinforcement learning aims to

maximize reward accumulation. In fact, the reinforcement learning process can be characterized using MDP. MDP is a Markov process with reward and decision, which can be represented by a quintuple $\langle S, A, P, R, \gamma \rangle$ [4].

2.2.2. Deep reinforcement learning algorithm. Deep learning algorithms are closely related to artificial neural networks. The construction of an artificial neural network is constructed by the basic unit - neuron. A deep reinforcement learning algorithm is an algorithm that combines deep learning and reinforcement learning. Additionally, deep neural network has a strong fitting ability. Therefore, this feature can be used to approximate the value function in reinforcement learning using a deep neural network, so that it outputs a value that approximates the value function for decision-making.

On the basis of this idea, the Deep Q-Network (DQN) was born, which can make the computer reach the level of human beings at Atari games [5]. DQN enables deep neural networks to approximate the optimal action-value function, and then make decisions about what actions an agent in a certain state should take.$Q(s, a)$ is used to represent approximate functions. In Q-Learning of reinforcement learning, the iterative update is performed using equation (1) [5].

$$Q(s, a) = (1 - \alpha)Q(s, a) + \alpha(R_{t+1} + \gamma \max_{a'} Q(s', a')) \quad (1)$$

In Equation (1), s and a stand for the current state and action taken, respectively; s' and a' represent the agent's subsequent state and subsequent action, respectively; the α represents the update rate.

For DQN, the weights of the neural network are derivable. Therefore, it is necessary to define the loss function, and then calculate the gradient of the loss function to the network weights, and update the weights of the neural network accordingly. Repeating the above process will eventually lead to the convergence of the model. θ is used to denote the weights of the network; the experience of the agent is the training data set, denoted by D. The loss function of the *i*-th iteration of the network is expressed in formula (2) [5]:

$$L_i(\theta_i) = E_{(s,a,r,s') \sim U(D)}[R_{t+1} + \gamma \, maxQ(s', a'; \theta_i) - Q(s, a; \theta_i)^2] \quad (2)$$

The network weights are updated by the method of error back propagation until the model reaches convergence, which completes the training of the network. In fact, this model has been trained to play Atari games and has successfully achieved human-level control.

2.2.3. A planning-based policy model. Value iteration is on the basis of the Markov decision process (MDP) formulation. The aim of the MDP is to discover a policy that obtains maximum expected return. A policy π can prescribe an action distribution for each state. Using the value iteration algorithm, Vn (state value function at iteration n) converges to V* (optimal state value function) as n tends to infinity [3]. The aim is to learn a policy which can maximize the expected return.
The planning-based policy model is shown in Figure 7 [3].

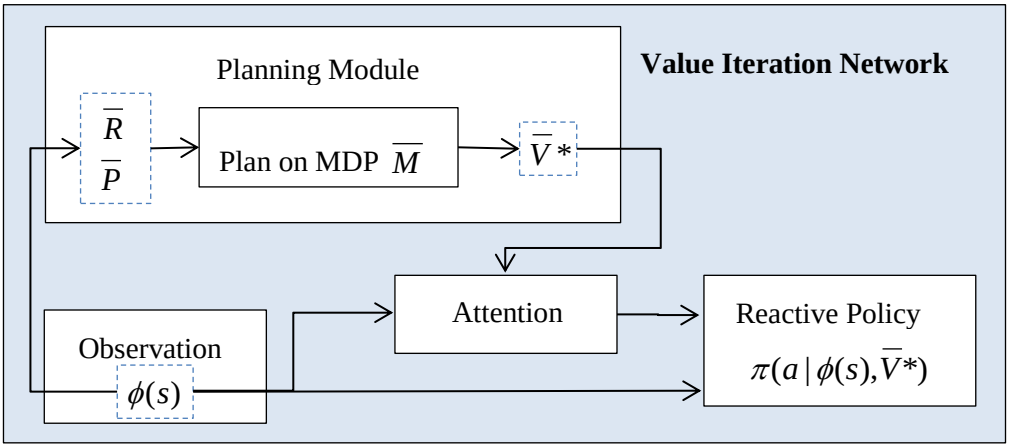


Figure 7. The planning-based policy model

Notice that an explicit planning computation is added and then map the observation to planning MDP $\overline{M}$. Neural Networks map observation to reward and transitions. Both of them ($\overline{R}$ and $\overline{P}$) are trainable. In fact, action prediction can require only subset of $\overline{V}^*$, Q values were chosen for current state.

2.4. Evaluation indicators

There are two evaluation indicators of the algorithm. The first is the average loss for training. It hopes that the network model can perform well on the training set, and the training average loss is as low as possible. The second is the prediction accuracy rate. The prediction accuracy rate is the core indicator of the algorithm. This metric can be adopted for comparing the performance of different algorithms and the generalization ability of the algorithms. The average loss is calculated on the training set, and the accuracy of the network prediction is obtained by testing on the test set. If the average loss during training is high, the network may not be able to converge. If the prediction accuracy rate is low during testing, it indicates that the generalization performance of the algorithm is poor.

3. Results

3.1. VIN and Implementation

3.1.1. Implementation process. In this section, experiments were carried out following the experimental environment described in 2.1. Expert samples are employed to train the network, and the cross-entropy loss is defined as the loss function for training. Moreover, the classic supervised learning approach is adopted. Error backpropagation and gradient descent methods are used to optimize network model parameters. Finally, the model is tested.

When optimizing the neural network, the RMSProp optimizer is used [6]. It is a commonly used optimizer that has been implemented in the deep learning framework PyTorch. In the experiment, the parameter learning rate lr of the RMSProp optimizer and the accuracy requirement eps are manually set. The lr set in the experiment is 0.002, and the eps value is 1×. What makes that special is that the value iteration of the mathematical formula is corresponding to the convolutional network structure.

Each channel in the convolution layer is consistent with the Q-function for a specific action, and convolution kernel weights are in consistence with the discounted transition probabilities. As a result, by recurrently employing a convolution layer K time, K iterations of VI are efficiently carried out. Each channel in this layer corresponds to Q for a particular action a. Subsequently, channel-wise max-pooling is performed to generate the value function layer V for the next iteration. Additionally, the whole Value Iteration Network (VIN) [3] is shown in Figure 8.

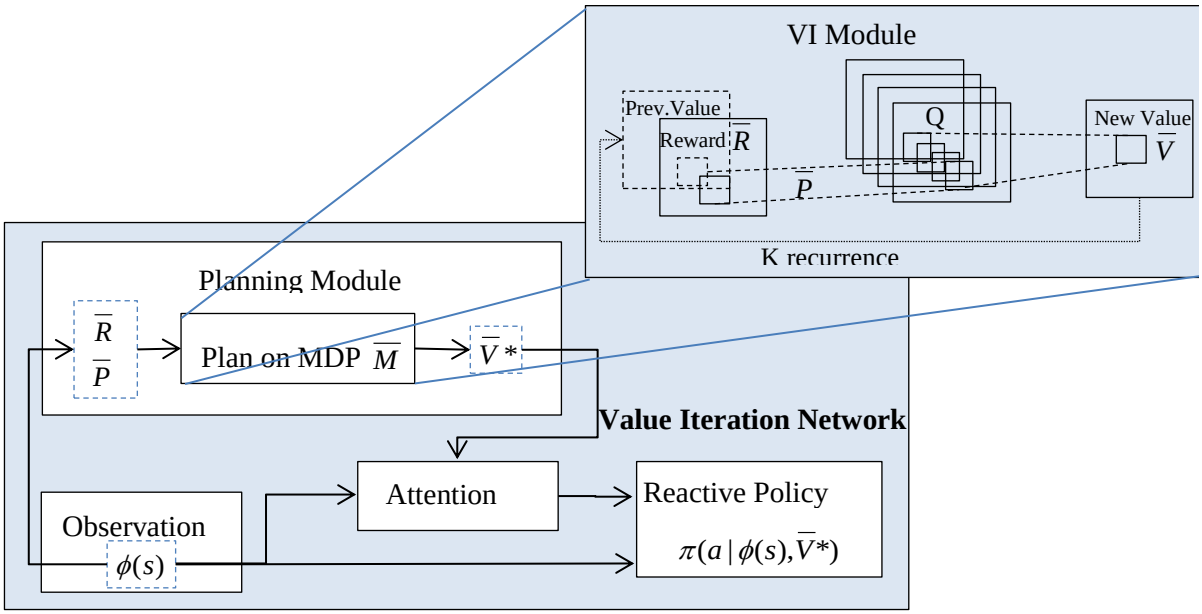
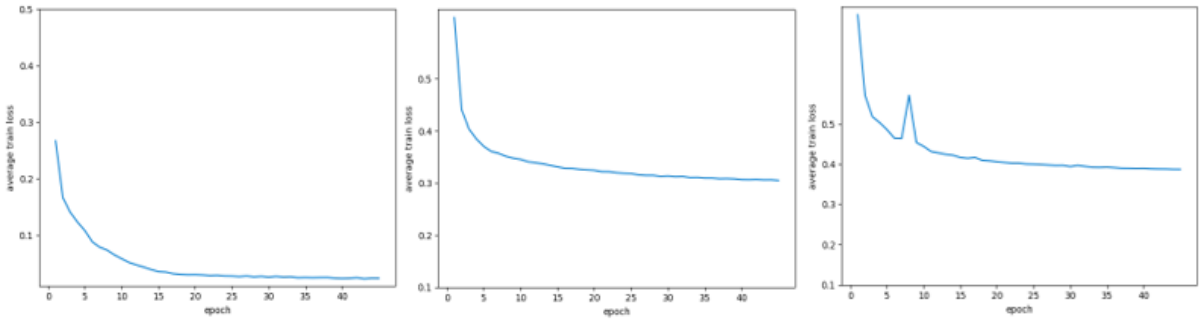


Figure 8. The whole VIN

3.1.2. Experiment result. The supervised learning from expert (shortest path) is used. And the input observation is image of obstacles + goal, current state. Compare VINs with reactive policies, and results are presented in Figure 9.



(a) 8×8 (b) 16×16 (c) 28×28

Figure 9. Average training loss of VIN

The performance of the model on the training set is only one aspect, and what is more important is the performance of the trained model on the test set. When dealing with deep learning problems, a common problem is that the average training loss accuracy is very low, but the accuracy of the network running on the test set is mediocre. This is due to the phenomenon of overfitting during training. At the same time, the prediction accuracy of the model can also be a good indicator of the generalization ability of the network. Therefore, the prediction accuracy of the model is the core indicator of detecting algorithm. Prediction accuracy on test set is shown in Figure 10.

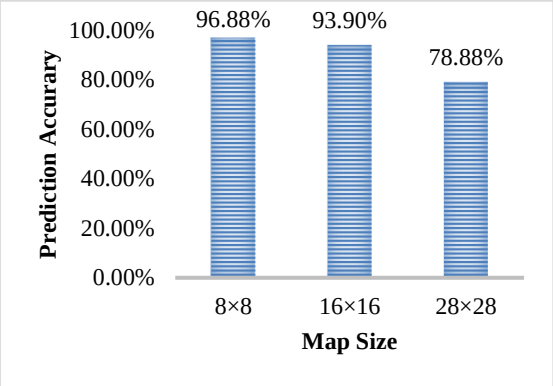


Figure 10. Prediction accuracy on test set of VIN.

The model performed well on small size of map. For example, the prediction accuracy rate is high on a map of size 8×8. However, as the map size increases, the complexity of the path planning problem increases rapidly, and the prediction accuracy rate of the network decreases rapidly. It can be seen that the prediction accuracy still maintains a high level on a map with a size of 16×16. But on a map of size 28×28, the prediction accuracy drops drastically. A map of size 28×28 is the largest map in this experiment and the most difficult map for the path planning problem.

The sharp drop in the prediction accuracy of the network model on such maps indicates that the original VIN still lacks the ability of long-distance multi-step planning on large maps, and the generalization ability of the network is still not good enough. Therefore, the model needs further improvement.

Besides, as for natural image inputs, VIN is shown to learn planning from natural image inputs [3]. Figure 11 is the overhead images of Mars terrain, showing a grid- world with natural image observations. The obstacles in the figure are represented as slope of 10◦ or more. It should be noted that the elevation data is not part of input [3].

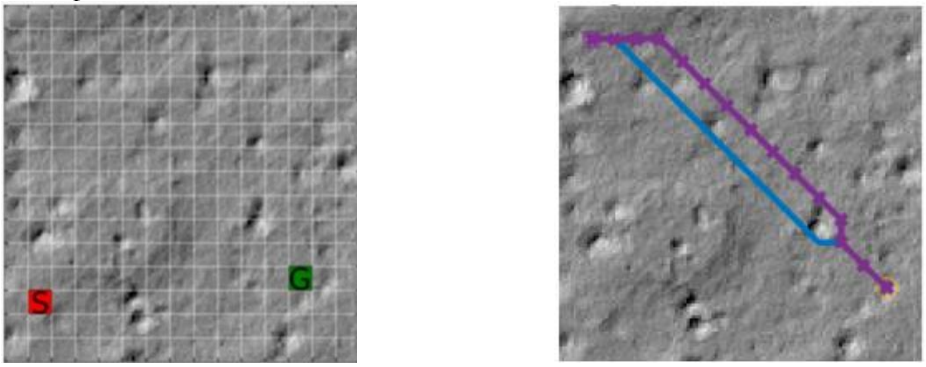


Figure 11. Path planning results for Mars terrain [3]

Table 2 shows the result of prediction loss and success rate of VIN and best achievable. After training, VIN could achieve a success rate of 84.8% [3].

Table 2. The prediction accuracy of each network

	Pred. loss	*Succ. rate*
VIN	0.089	84.8%
Best achievable		90.3%

3.1.3. Visualization. Here are the path planning results on different size of map on the test set.

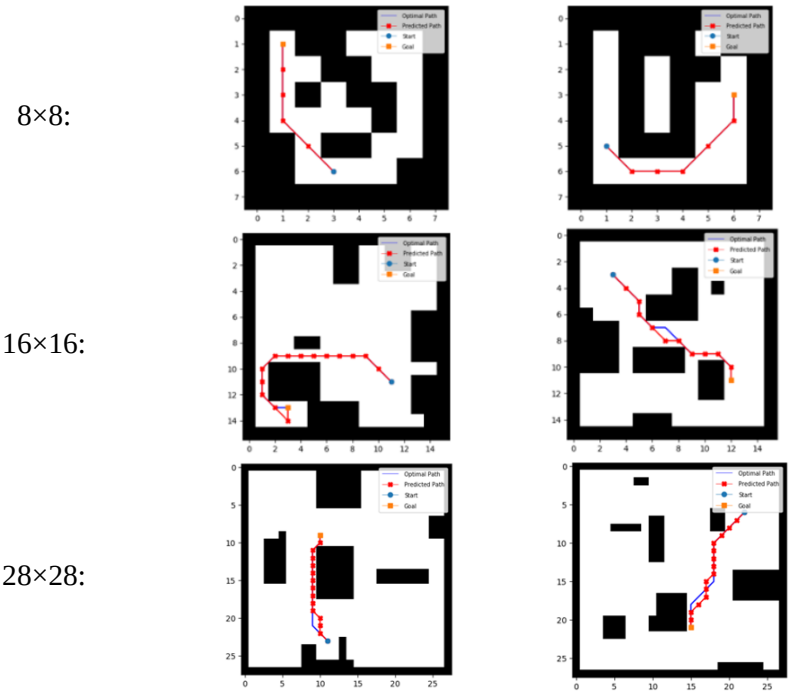


Figure 12. Path planning results for maps of different sizes on the test set.

And here is value iteration process is shown in Figure 13.

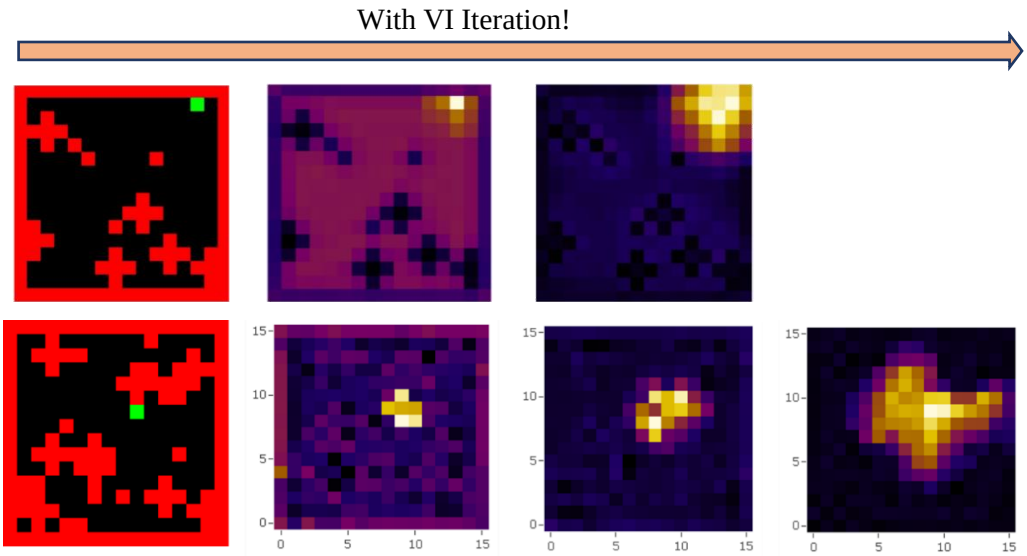


Figure 13. Visualized value image

As you can see, the place around the target position gradually becomes bright, indicating that the corresponding position value is high and the agent should be there as close as possible. The area of the obstacle is turning black, indicating that the corresponding position value is low and the agent should avoid there as far as possible. Through value iteration, the network can seek advantages and avoid disadvantages, and has the planning ability.

3.2. Improvement of VIN

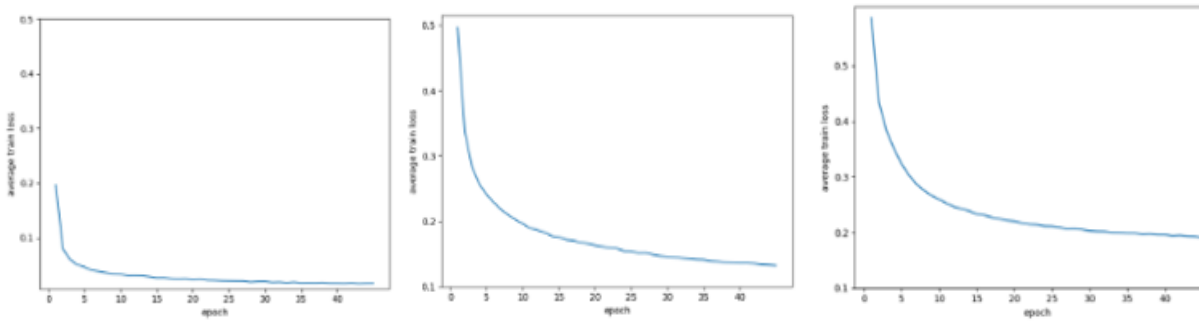
Although VIN has better mathematical interpretability and planning ability than the deep Q network, from the experimental results, the prediction accuracy of VIN on the large map is still low. That is to

say, VIN is not suitable for complex situations as the planning ability and generalization ability are still insufficient.

Therefore, a series of improvements on the basis of the original VIN this part is made. Compared with the original VIN, the prediction accuracy of improved VINs has been greatly improved on the large map.

3.2.1. Deeper VIN. In VIN, the weights in all VI models (weight sharing) are bound. If the weights in the VI model are unwrapped, this means that there is a different weight for each VI module. Also, there are skip-connections between VI modules [7].

In the experiment, the network structure obtained by the above improvement is reproduced, and training and testing are carried out. Deeper VIN is trained on three different sizes of maps, and then tested on the test set. Figure 14 presents the results.



(a) 8×8 (b) 16×16 (c) 28×28

Figure 14. Average training loss of deeper VIN.

The prediction accuracy of Deeper VIN is shown in Figure 15.

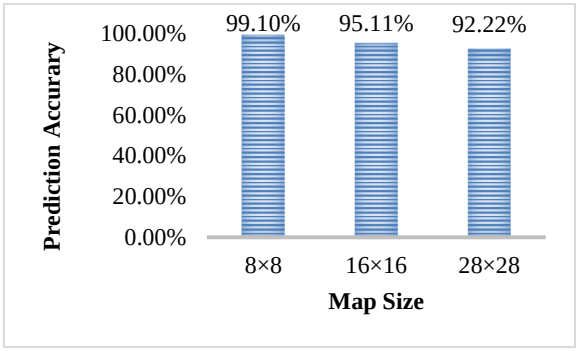


Figure 15. The prediction accuracy of Deeper VIN.

Figure 15 shows that there is a breakthrough has been made in prediction accuracy on maps of all sizes. The average training loss of the network is obviously reduced, and it performs well on the training set. On a large-size map (28 × 28), the prediction accuracy of deeper VIN reaches 92.2152%, far exceeding that of the original VIN (78.8834%). It can be said that compared with origin VIN, Deeper VIN has greatly improved its long-term planning ability and generalization ability.

3.2.2. Dueling Architecture (ICML2016) plus Batch Normalization (BN_Layers). It can be noticed that the Q function can be written into two parts [8].

$$Q_\pi(s,a) = V_\pi(s) + A_\pi(s,a) \tag{3}$$

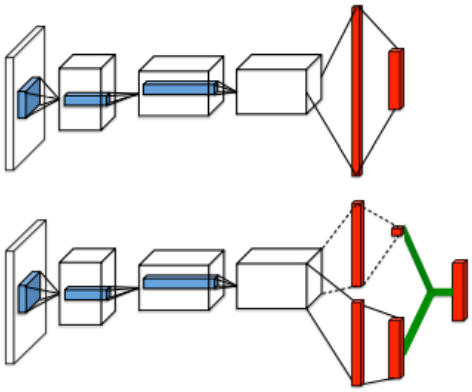
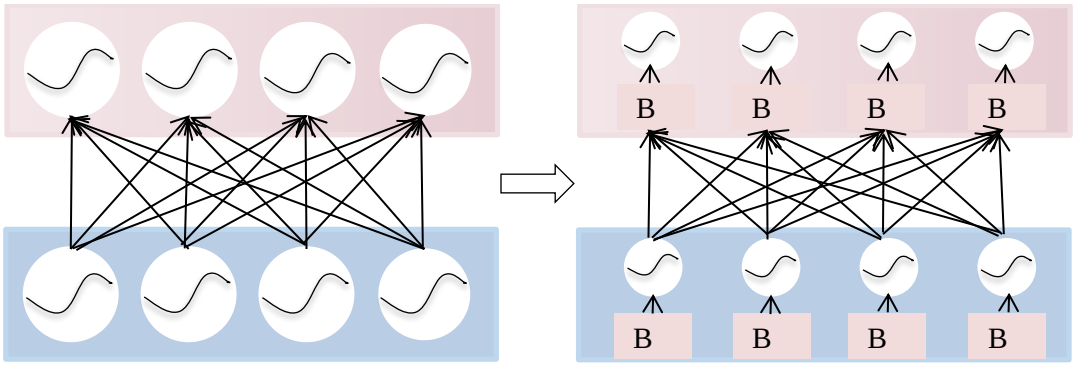


Figure 16. Competitive Structures in Neural Networks [8]

In competitive structures in neural networks, the upper part has no competitive structure, and the lower part replaces the fully connected layer in the original network with a competitive structure, which improves the data output [8]. Meanwhile, because the network has been greatly deepened, the complexity of the model has also increased significantly. This slows down the rate of decline in the average training loss of the network significantly. The average training loss remains at a high value at the end of the neural network training process. For this, just changing the last layers (FC layers, softmax) by using Batch Normalization will do the trick [9]. There is:

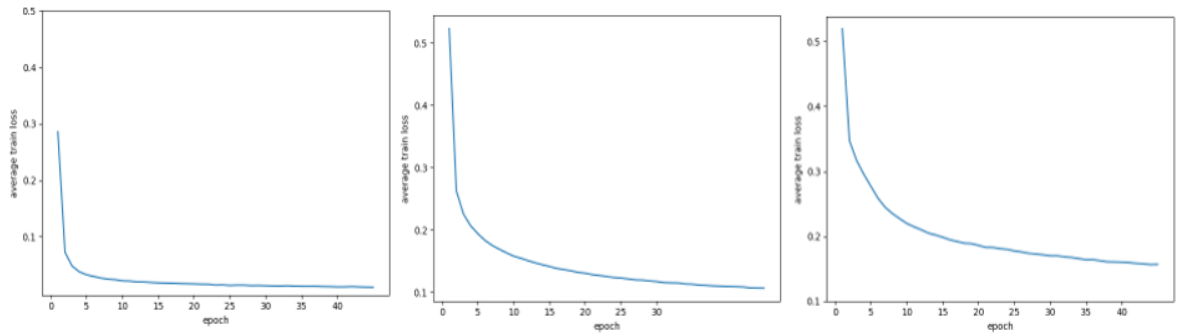
$$y = \frac{x - E[x]}{\sqrt{Var[x] + \varepsilon}} \cdot \gamma + \beta \tag{4}$$



Without With BNLayer

Figure 17. BN layers in a neural network

Batch Normalization (BN) is already a common operation in current deep learning algorithms. It has been implemented in the deep learning framework PyTorch and can be called directly. By using BN layers, the Network will converge faster. Therefore, the model can be trained with a larger learning rate, reducing the loss function and avoiding exploding gradients. After adding the BN layer, the experimental results of the network test are shown in Figure 18:



(a) 8×8 (b) 16×16 (c) 28×28

Figure 18. Average training loss of further improved VIN.

It can be seen that the average training loss of the neural network after further improvement drops significantly faster. The training speed of deep neural networks has been accelerated to reach convergence faster.

3.2.3. Hierarchical Structure VIN. Path planning on the big map is the core of this research. When dealing with large map problems, the agent often needs to go through multiple steps from the starting position to the target position.

Assuming that it takes steps to reach from the starting position to the target position, the network will need at least one step to reach the value iteration. With the size of the map being large, the number may be large, which causes the network to perform many value iterations. In the meanwhile, because of the large size of the map, it is challenging for the network to use global information when planning, and it is easy to plan according to local conditions, ignoring the global information.

In order to convey reward information faster in VI, and lower the effective K, multiple levels of resolution are used based on all the previous improvements. Two levels of value iteration are performed in the hierarchical structure, including Deeper VIN and VIN with dueling architecture (ICML2016) plus BN_Layers. Then, the Hierarchical Structure VIN is obtained, and its structure is shown in Figure 19 [3]:

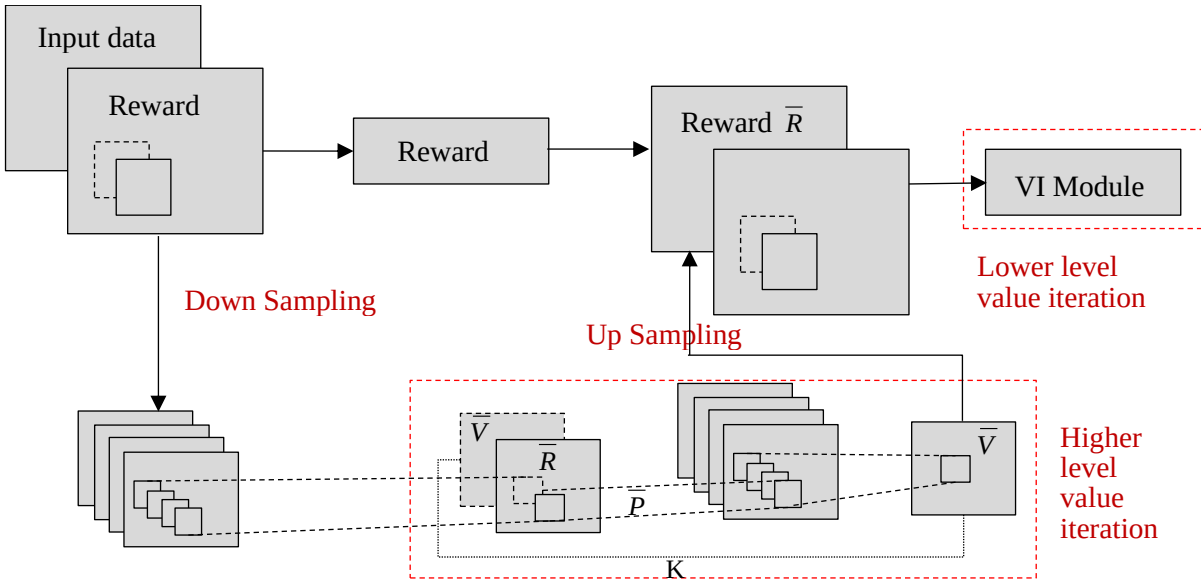
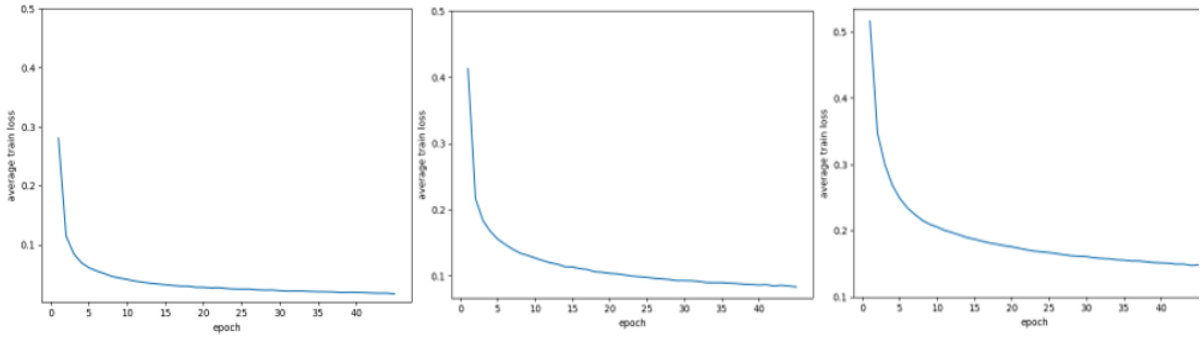


Figure 19. The Structure of Hierarchical Structure VIN.

A copy of the input is fed into a convolution layer and subsequently down-sampled. Then it is fed into a VI module and later up-sampled, and supplemented as an additional channel in the reward layer. Average training loss of Hierarchical Structure VIN is shown in Figure 20.



(a) 8×8 (b) 16×16 (c) 28×28

Figure 20. Average training loss of Hierarchical Structure VIN.

The prediction accuracy of Hierarchical Structure VIN is shown in Figure 21.

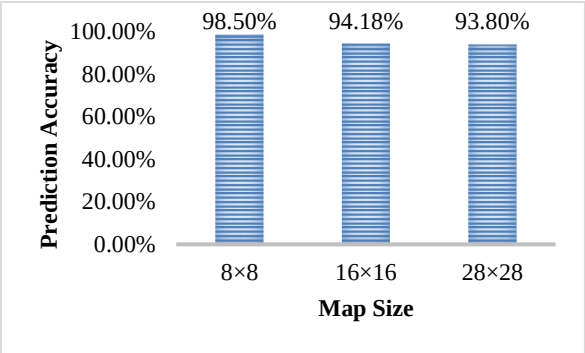


Figure 21. The prediction accuracy of Hierarchical Structure VIN.

The Hierarchical Structure VIN obtained after a series of improvements, the convergence speed of the network is accelerated and the average training loss is further reduced. The model has the best performance with 93.7989% accuracy on the big map (28 × 28) of test sets. It far exceeds the experimental result of the original value iteration network (78.8834%). In contrast, Hierarchical Structure VIN has better long-term planning capabilities and network generalization capabilities, and truly has the ability to solve large-scale and complex path planning problems.

3.3. Comparative analysis

The experimental results of each network tested in the above experiments are summarized as presented in Table 3.

Table 3. The prediction accuracy of each network

Network Structure	8×8	16×16	28×28
Original VIN	96.8750%	93.9019%	78.8834%
Deeper VIN	99.1027%	95.1141%	92.2152%
Dueling + BN_Layer	98.0924%	95.0924%	92.6279%
Hierarchical VIN	98.5027%	94.1778%	93.7989%

It can be seen that the accuracy of Hierarchical VIN is higher than 93% on different size maps, especially on the large map (28×28), it still shows a high planning accuracy rate and strong applicability. The performance of the network on the big map has improved significantly, from less than 79% to more than 93%. It shows that a series of improvements are effective and the network generalization ability is improved.

In order to further illustrate the experimental performance of Hierarchical VIN, the results obtained by other deep reinforcement learning algorithms in the same experimental environment are compared horizontally. According to the experimental results in the research of Aviv, et al. (2016), the results of using Deep Q-Network (DQN) network for deep reinforcement learning and using Fully Convolutional Network (FCN) are compared with the results of Hierarchical VIN in this experiment [10]. Experimental results of these three different algorithms as illustrated in Table 4.

Table 4. The prediction accuracy of each algorithm

Map Size	Deep Q-Network (DQN)	Fully Convolutional Network (FCN)	Hierarchical VIN	Accuracy Improved (Compared with DQN)
8×8	97.9%	97.3%	98.5%	+0.6%
16×16	87.6%	88.3%	94.2%	+6.6%
28×28	74.2%	76.6%	93.8%	+19.6%

Table 3 clearly shows that Hierarchical VIN has high accuracy (over 93%) on any size map. In particular, the performance on the big map is much better than the other two algorithms. Compared with DQN, its accuracy is increased by nearly 20%. This shows that Hierarchical VIN has a much better generalization ability than DQN and FCN, has the ability to solve complex, large-scale, and difficult path planning problems for long-term planning, and is suitable for maps of different sizes.

4. Conclusion

Value Iteration Network (VIN) has good planning ability and strong mathematical interpretability. The VIN structure ingeniously corresponds to the mathematical Markov decision process, so that when the data is propagated forward, it is a Markov decision on the data. In view of the remaining problems in the original value iteration network model, such as the lack of planning ability for complex situations, the network generalization ability is not strong enough, the prediction accuracy on the big map, etc., a series of improvement measures are used, including using deeper value iteration network, dueling architecture (ICML2016) plus batch normalization layers, and multiple levels of resolution. Finally, the Hierarchical Structure VIN was obtained. In the Hierarchical Structure VIN, down-sampling and up-sampling are performed value iteration operations at two levels, respectively, and the best results are obtained. Hierarchical Structure VIN achieves over 93% planning accuracy on large-scale maps (28×28), far exceeding the results of the original value iteration network. It shows that Hierarchical Structure VIN has strong generalization ability and long-term planning ability.

References

[1] Cherkassky, B. V., Goldberg, A. V., & Radzik, T. (1996). Shortest paths algorithms: Theory and experimental evaluation. Mathematical programming, 73(2), 129-174.

[2] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.

[3] Tamar, A., Wu, Y., Thomas, G., Levine, S., & Abbeel, P. (2016). Value iteration networks. Advances in neural information processing systems, 29.

[4] Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. 2nd. 59-69

[5] Mnih, V., Kavukcuoglu, K., Silver, D., et al., (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533.

[6] Tieleman, Tijmen, and Geoffrey Hinton. (2012), Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning 4(2): 26-31

[7] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, 4700-4708.

[8] Wang, Z., Schaul, T., Hessel, M., Hasselt, H., et al. (2016). Dueling network architectures for deep reinforcement learning. In International conference on machine learning, 1995-2003.

[9] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning, 448-456.

[10] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, 3431-3440.