An Adaptive State Space Segmentation Method Based on ART Neural Network with Two Learning Phases for Reinforcement Learning

Taisuke Nakamura†, Takeshi Kamio†, Kunihiko Mitsubori‡, and Hisato Fujisaka†

†Hiroshima City University, 3-4-1, Ozuka-higashi, Asaminami-ku, Hiroshima-shi, Hiroshima, 731-3194, Japan
‡ Japan Coast Guard Academy, 5-1, Wakaba-cho, Kure-shi, Hiroshima, 737-8512, Japan
Email: tsk_n@imc.im.hiroshima-cu.ac.jp, kamio@im.hiroshima-cu.ac.jp

Abstract– The trade-off between exploration and exploitation has often been discussed in studies on reinforcement learning (RL). This is because exploration and exploitation influence the quality of solutions and the learning efficiency respectively. Previously, we have proposed an adaptive state space segmentation method based on ART neural network (ART) to execute RL effectively. However, if the exploration strength is too large, the learning efficiency degreases rapidly. Since the appropriate strength is generally unknown, this problem must be solved. In this paper, we propose a new segmentation method based on ART with two learning phases to improve our conventional method in the tolerance of exploration strength.

1. Introduction

Reinforcement learning (RL) [1] is a goal-directed learning method based on the agent-environment interaction. The agent senses the state of the environment from the perceptual inputs and selects one of actions by the policy. The environment makes a transition to a new state and gives the agent a reward. Through the iterations of these processes, the agent completes the value function to maximize cumulative reward. As a result, the agent can achieve a given task.

RL has been applied to many purposive behavior tasks with continuous state variables and discrete-valued actions. However, since RL algorithms require the discrete state space, many researchers have proposed state space segmentation methods. We also have proposed a segmentation method based on ART neural network (ART) [8]. It has been confirmed by computer simulations that our method is much better than similar ones [4]-[7].

By the way, the trade-off between exploration and exploitation has often been discussed in studies on RL. This is because exploration and exploitation influence the quality of solutions and the learning efficiency respectively. Our conventional method is useful for not only the state space segmentation but also the balance between exploration and exploitation. However, if the exploration strength is too large, the learning efficiency degreases rapidly. The reason is that frequent failures in the task caused by exploration do serious damage to the state space which grows desirably. Although finding the appropriate exploration strength is one of the most efficient solutions to this problem, it is generally unknown. Therefore, it is valuable that our conventional method improves in the tolerance of exploration strength.

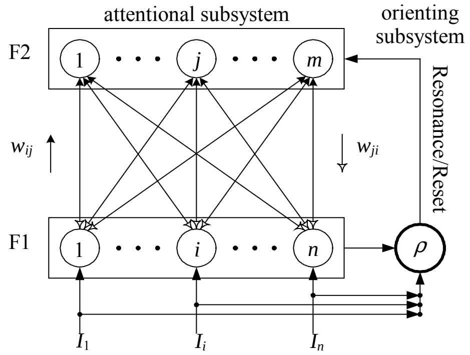


Fig.1 Structure of ART neural network.

In this paper, we propose a new state space segmentation method based on ART with two learning phases (LPs). If the first LP (LP1) is active, the proposed method is almost equal to the conventional one. On the other hand, if the second LP (LP2) is active, then the state space constructed in LP1 is completely preserved and finer segmentation is executed. Moreover, LP changes round as the need arises. Finally, it is confirmed by simulating Q-learning [2] for the acrobot swing-up task [8] that the proposed method is better than the conventional one in terms of the tolerance of exploration strength.

2. Fuzzy-ART Neural Network

ART [3] is shown in Fig.1. The attentional subsystem consists of an input layer (F1) and a category layer (F2). An F1 neuron i and an F2 neuron j are interconnected by a bottom-up weight w_{ij} and a top-down weight w_{ji}. In the case of fuzzy-ART, w_{ij} is equal to w_{ji}. A top-down weight vector $\mathbf{w}_j=[w_{j1},\cdots,w_{jn}]$ is a memory pattern of an F2 neuron j and the index j is the category number. On the other hand, the orienting subsystem has a classification precision called the vigilance parameter ρ.

Here, we explain the behavior of fuzzy-ART. After a normalized vector $\mathbf{I}\in[0,1]^n$ is input to F1, each F2 neuron j receives a choice strength T_j:

$$T_j = |\mathbf{I} \wedge \mathbf{w}_j|_1 / (\alpha + |\mathbf{w}_j|_1), \quad (j=1,\cdots,m), \qquad (1)$$

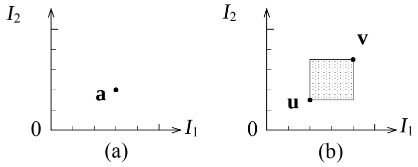


Fig.2 Category space in fuzzy-ART.

where $(\mathbf{p}\wedge\mathbf{q})_i$=min($p_i$,$q_i$), $|\mathbf{p}|_1=\sum|p_i|$, α is a positive value called the choice parameter, and m is the number of F2 neurons. At F2, the neuron J with the maximal choice strength is activated. If there are several neurons with the maximal choice strength, the neuron with the minimal index is selected from them. The activated F2 neuron J has the memory pattern $\mathbf{w}_J$ which is the most similar to the input $\mathbf{I}$. The F2 neuron J provides F1 with $\mathbf{w}_J$. Then the orienting subsystem calculates the matching degree A_J from the F1 activity $\mathbf{I}\wedge\mathbf{w}_J$ and the input $\mathbf{I}$:

$$A_J = |\mathbf{I}\wedge\mathbf{w}_J|_1 / |\mathbf{I}|_1 \,. \tag{2}$$

Also the orienting subsystem compares A_J with $\rho\in[0,1]$. If $A_J\geq\rho$, $\mathbf{w}_J$ is updated as follows:

$$\mathbf{w}_J^{new} = \beta(\mathbf{I}\wedge\mathbf{w}_J^{old}) + (1-\beta)\mathbf{w}_J^{old} \,, \tag{3}$$

where $\beta\in[0,1]$ is the learning rate and is often set to one. If $A_J<\rho$, the F2 neuron J is reset. The reset F2 neuron J is enduringly inactivated until the input $\mathbf{I}$ changes. The above processes are iterated unless ART finds an F2 neuron with $A_J\geq\rho$. However, if all the F2 neurons are reset, a new neuron m+1 is added to F2 and $\mathbf{w}_{m+1}$=$\mathbf{I}$.

Especially, in the case of fuzzy-ART, the input $\mathbf{I}$ is the complement code of the original input $\mathbf{a}\in[0,1]^n$:

$$\mathbf{I} = [a_1,\cdots,a_n,a_1^c,\cdots,a_n^c] \in [\mathbf{a},\mathbf{a}^c] \,, \tag{4}$$

where $a_i^c\equiv 1-a_i$. This is because $\mathbf{I}$=$\mathbf{a}$ makes all the memory patterns converge on the zero vector. We enumerate the characteristics of category spaces designed by fuzzy-ART whose inputs are given by (4).

- If $\mathbf{w}_j$ memorizes an only input $\mathbf{I}$, the category space is a point defined by the original input $\mathbf{a}$ (see Fig.2(a)).
- If $\mathbf{w}_j$ memorizes more than one input, the category space becomes a hyper rectangular. When $\mathbf{w}_j$=[$\mathbf{u}$,$\mathbf{v}^c$], the hyper rectangular is illustrated by Fig.2(b).
- For every inner point of the category space j, the matching degree A_j is given by $A_j^{\mathrm{IN}}=|\mathbf{w}_j|_1/n$.

3. Q-learning (QL)

We use Q-learning (QL) [2], which is representative of RL algorithm. QL has the value function called the Q-value, which is defined for each state-action pair. The purpose of QL is getting Q-value to achieve a given task. QL is executed as follows. It is assumed that the agent senses the state $s_t\in S$ from the perceptual inputs $\mathbf{P}_t$ and selects the action $a_t\in A$ by the policy at the time t. As a result, the environment makes a transition to a new state $s_{t+1}\in S$ and gives the agent a reward r_{t+1}. In this case, Q-value $Q(s_t,a_t)$ is updated by

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha_{QL}[r_{t+1} + \max_{a\in A} Q(s_{t+1},a) - Q(s_t,a_t)] \,, \tag{5}$$

where α_{QL} is the learning rate. These processes are iterated until the agent completes Q-value to maximize cumulative reward under given learning conditions. If the desired Q-value is obtained, the agent can achieve a given task. The policy used here is based on the ε-greedy policy. That is to say, while the agent basically selects the action with the largest Q-value in the current state, the action is randomly selected with a small probability ε. The probability ε is adjusted in order to converge on the greedy policy:

$$\varepsilon = \begin{cases} \varepsilon_0, & \text{if } P_\varepsilon < N_\varepsilon, \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

where $\varepsilon_0\neq 0$, P_ε is the counter to designate the policy, and N_ε is the duration of the ε-greedy policy. The unit of P_ε and N_ε is the episode of a given task.

4. State Space Segmentation Based on ART with Two Learning Phases

4.1. Conventional Method (CM)

If ART classifies the perceptual inputs and the category index is given to the agent, then ART can segment the continuous state space adaptively. This is the basic idea of the state space segmentation based on ART [4]-[8]. However, if ART is applied to QL according to only the basic idea, designers ought to discover following four problems. The first one is that the segmented state space is not necessarily suitable for the agent. This is because such a state space is constructed only by the criteria which ART has. The second one is that designers have to fix the segmentation precision by trial and error, since the vigilance parameter is a constant value. The third one is that ART has no sensitivity to the sign of each perceptual input. If the sign is important for the achievement of the task, the sensitivity helps to generate states appropriately. The fourth one is that all the states exist enduringly, even if some of them become unnecessary. As a result, the calculation cost of ART may increase rapidly. We have proposed the state space segmentation method to solve these problems [8].

4.2. Proposed Method (PM)

The conventional method (CM) detects the convergence of Q-value from the stability of the number of states (i.e., F2 neurons) N_S and the task achievement ratio R. If R is low at the time of convergence, CM judges that the segmentation is insufficient and increases ρ. Moreover, if the F2 neuron J is necessarily reset by the increment of ρ, or $A_J^{\mathrm{IN}}<\rho^{new}$, then the category space is segmented into two subspaces and they succeed to Q-value of the F2 neuron J. After executing these processes, the policy becomes ε-greedy forcibly (i.e., P_ε=0). Therefore, CM is useful for the balance between exploration and exploitation.

However, if the exploration strength (i.e., ε) is too large, the learning efficiency decreases rapidly. The reason is that frequent failures in the task caused by exploration do serious damage to the state space which grows desirably. Designers using CM will hit on two direct ways to inhibit the damage. One is setting the learning rate β to a very small value. But, since the state space is constructed very slowly, the learning efficiency may worsen. The other is finding an appropriate ε. But, since the value is generally unknown, ε must be decided by trial and error. Therefore, it is valuable that CM improves in the tolerance of ε.

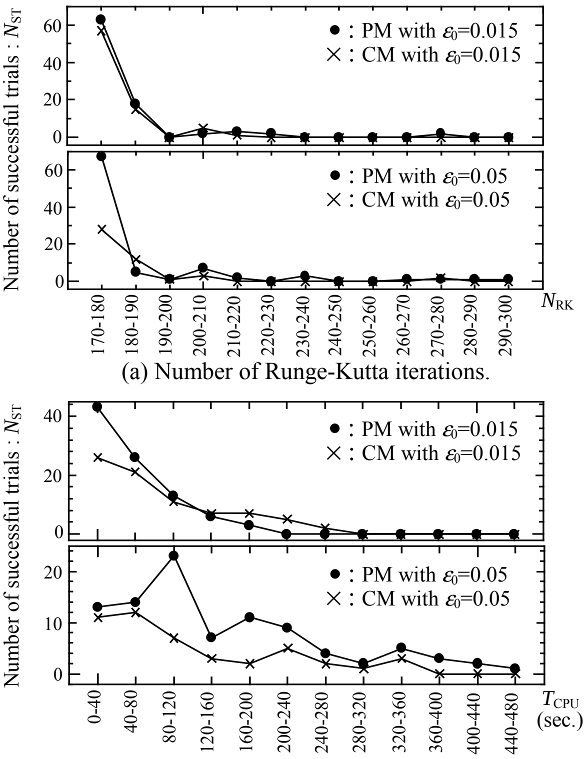
Here, we propose a new state space segmentation method based on ART with two learning phases (LPs). At the beginning of QL, the first LP (LP1) is active. If ART works in LP1, the proposed method (PM) is almost equal to CM. $E_{\max}$ is set to E_1 ($<E_{\text{fin}}$). E_1 and E_{fin} are the first and final goals of R respectively. If it is satisfied that $R \geq E_{\max} = E_1$ during N_{ph1} successive episodes, then PM judges that ART has constructed good state space in LP1 and makes the second LP (LP2) active. However, if $R \geq E_{\text{fin}}$, LP1 remains active. If ART works in LP2, then ART completely preserve the state space constructed in LP1 and executes finer segmentation. Concrete processes in LP2 are as follows. β_1 is set to zero. After preserving ρ_1 (i.e., $\rho_1^{prev}=\rho_1$), ρ_1 is set to $A_{\min}$. β_1 and ρ_1 are the learning rate and vigilance parameter for F2 neurons generated in LP1. $A_{\min}$ is the minimum value of A_j^{IN}. If a new F2 neuron is generated in LP2, it succeeds to Q-value of the existing F2 neuron which is activated under the condition that $\rho_1=\rho_1^{prev}$ in order to keep up the learning efficiency. F2 neurons generated in LP2 are preserved, only if the present episode is successful. Also, PM continues adding $\Delta\rho_2$ to ρ_1 every successful episode in LP2, until at least one new F2 neuron is preserved. The learning rate and vigilance parameter for F2 neurons generated in LP2 are given by β_2 and ρ_2 respectively. β_2 is set to zero and ρ_2 is set to a large value in order to inhibit the damage to the state space constructed in LP1. Thus, PM can obtain the tolerance of ε by the processes executed in LP2. However, since the state space constructed in LP2 is not necessarily suitable for the agent, LP1 must be active again. Therefore, after N_{ph2} episodes are passed in LP2, PM compares R with E_{fin} every episode. If $R<E_{\text{fin}}$ is observed, then PM makes LP1 active and parameters are set as follows: $\beta_1=1.0$, $\rho_1=\rho_1^{prev}$, $E_{\min}=E_1$, $E_{\max}=E_{\text{fin}}$. As a result, PM can reconstruct the state space based on F2 neurons generated in LP2 drastically. But, if the category segmentation caused by the increment ρ_1 is occurred, then all the F2 neurons generated in LP2 are deleted and parameters are set as follows: $E_{\min}=E_0$, $E_{\max}=E_1$.

As mentioned above, PM can obtain the desirable state space by the changeover between LP1 and LP2. PM is listed as follows.

1) Initialize the condition of QL and ART.
2) Each of counters (N_S, N_{ep}, P_ε, C, U_j, C_{ph}, T_{D1}, T_{D2}) is set to zero. N_S is the number of states (i.e., F2 neurons). N_{ep} is the total number of episodes. P_ε is the counter to designate the policy. C is the number of successive episodes with a non-increasing N_S and $\varepsilon=0$. U_j measures the number of times when F2 neuron j is used. C_{ph} is the counter to control the changeover between LP1 and LP2. T_{D1} and T_{D2} check the timing to delete F2 neurons generated in LP1 and LP2 respectively. Moreover, parameters are initialized as follows: $L_{ph}=1$, $\rho_1=\rho_0$, $\beta_1=1.0$, $E_{\min}=E_0$, $E_{\max}=E_1$. L_{ph} shows the present LP (e.g., $L_{ph}=1$ means LP1).
3) Execute QL through an episode which is defined by a given purposive behavior task. If a new F2 neuron (i.e., state) j is generated, then $U_j=D$. If an existing F2 neuron j is used as the state, then U_j increases by one. The perceptual inputs $\mathbf{P}\in\Re^n$ are given to ART in the following form:
$$\begin{cases}\mathbf{I}=[\mathbf{a},\mathbf{a}^c]\in[0,1]^{4n}\ , \\ a_i = f_{\text{N}}(p_i)\in[0,1]\ , \\ \mathbf{p}=[|P_1|,\cdots,|P_n|,\text{sgn}(P_1),\cdots,\text{sgn}(P_n)]\in\Re^{2n},\end{cases} \tag{7}$$
where $\mathbf{I}$ is the input vector of ART and f_{N} is the function to normalize p_i to [0,1]. In the case of $L_{ph}=2$, the following processes are added. A new F2 neuron succeeds to Q-value of the existing F2 neuron which is activated under the condition that $\rho_1=\rho_1^{prev}$. F2 neurons are preserved, only if the present episode is successful. Also, PM adds $\Delta\rho_2$ to ρ_1 every successful episode, until at least one new F2 neuron is preserved in the present LP2.
4) At the end of each episode, observe N_S and R. If N_S is larger than its previous value or $\varepsilon\neq 0$, C is set to zero; otherwise, C increases by one. Moreover, N_{ep}, P_ε, T_{D1} and T_{D2} increase by one. In the case of $L_{ph}=1$, if $R\geq E_{\max}$, C_{ph} increases by one; otherwise C_{ph} is set to zero. In the case of $L_{ph}=2$, C_{ph} increases by one.
5) Execute the changeover between LP1 and LP2 as follows. If $L_{ph}=1$, $C_{ph}\geq N_{\text{ph1}}$, $R<E_{\text{fin}}$, then $L_{ph}=2$, $C_{ph}=0$, $\beta_1=0.0$, $\beta_2=0.0$, $P_\varepsilon=N_\varepsilon$. Moreover, after $\rho_1^{prev}=\rho_1$, $\rho_1=A_{\min}$. If $L_{ph}=2$, $C_{ph}\geq N_{\text{ph2}}$, $R<E_{\text{fin}}$, then $L_{ph}=1$, $C_{ph}=0$, $\beta_1=1.0$, $\beta_2=0.0$, $E_{\min}=E_1$, $E_{\max}=E_{\text{fin}}$, $\rho_1=\rho_1^{prev}$.
6) If $L_{ph}=1$ and $T_{D1}\geq N_{D1}$, then delete F2 neuron j generated in LP1 which satisfies $U_j<D$; otherwise, $U_j=0$. After that, $T_{D1}=0$. If $T_{D2}\geq N_{D2}$, then delete F2 neuron j generated in LP2 which satisfies $U_j<D$; otherwise, $U_j=0$. After that, $T_{D2}=0$. But, if $R\geq E_{\text{fin}}$, then these processes are not executed.
7) If $L_{ph}=1$, compare R with E given by (8). If $R<E$, add $\Delta\rho_1$ to ρ_1. The maximum ρ_1 and E are $\rho_{\max}$ and $E_{\max}$ respectively.
$$E=\begin{cases}0, & \text{if}\ \ C<N_E, \\ E_{\min}+(C-N_E)\cdot\Delta E, & \text{if}\ \ C\geq N_E.\end{cases} \tag{8}$$
8) If $L_{ph}=1$ and F2 neuron j satisfies $A_j^{\text{IN}}<\rho_1$, segment the category space j into two subspaces ($j1$ and $j2$) as follows. If F2 neuron j has the weight $\mathbf{w}_j=[\mathbf{u},\mathbf{v}^c]$, F2 neurons $j1$ and $j2$ have $\mathbf{w}_{j1}$ and $\mathbf{w}_{j2}$ respectively:

Table.1 Success rates of CM and PM

Exploration strength	CM	PM
ε_0=0.015	79%	91%
ε_0=0.05	46%	94%



(a) Number of Runge-Kutta iterations.

(b) CPU time.

Fig.3 Comparison of PM with CM.

$$\mathbf{w}_{j1}=[\mathbf{u},\mathbf{u}^c],\quad \mathbf{w}_{j2}=[\mathbf{v},\mathbf{v}^c]. \tag{9}$$

After copying Q-value of the F2 neuron *j* into ones of the F2 neurons *j*1 and *j*2, delete F2 neuron *j*. U_{j1} and U_{j2} are set to *D*. P_ε is set to zero. Moreover, if there are F2 neurons generated in LP2, delete all of them. The parameters are set as follows: $E_{\min}$=E_0, $E_{\max}$=E_1.

9) Judge whether QL is finished or continued. If QL is continued, go to step 3.

5. Experiments

The acrobot is a robot with two links and two joints [8]. Since the second joint only has an actuator, it can exert torque. One objective for controlling the acrobot is to swing the tip above the first joint by an amount equal to one of links. If the agent achieves the task during 1000 successive episodes, we judge that QL has succeeded in the present learning. On the other hand, if N_{ep} is over 20000, we judge that QL has failed in the present learning. The initial position of acrobot is randomly selected within [−2.5, +2.5] (deg.). The behavior of the acrobot is analyzed by the fourth order Runge-Kutta method. For the acrobot swing-up task, PM is compared with CM. The number of learning trials is 100. The computer simulations are executed on Sun Ultra SPARC IIIi (CPU: 1.28GHz, Memory: 4GB). The parameters are set as follows: α_{QL}=0.1, ε_0=0.015, 0.05, N_ε=100, E_0=0.7, E_1=0.7, $E_{\rm fin}$=1.0, ΔE=0.001, N_E=100, α=1.0, ρ_0=0.6, $\rho_{\max}$=0.8, $\Delta\rho_1$=0.0001, $\Delta\rho_2$=0.01, ρ_2=0.95, *D*=1, N_{D1}=400, N_{D2}=50, $N_{\rm ph1}$=200, $N_{\rm ph2}$=100.

First, we show the learning success rates (*SR*s) of CM and PM in Table 1. Table 1 indicates PM keeps *SR* high even if exploration strength is large. Next, we show the quality of solutions and the learning time in Fig.3. The former can be estimated by the number of Runge-Kutta iterations $N_{\rm RK}$. The latter is evaluated by CPU time $T_{\rm CPU}$ which the successful learning consumes. Fig.3 means PM is much better than CM in terms of $N_{\rm RK}$, although PM and CM have similar distribution of $T_{\rm CPU}$.

6. Conclusions

We have proposed a state space segmentation method based on ART with two learning phases. Simulation results have shown that this method improves the conventional one in the tolerance of exploration strength.

References

[1] R. S. Sutton and A.G. Barto, Reinforcement Learning: An Introduction, The MIT Press, MA, 1998.

[2] C. J. C. H. Watkins and P. Dayan, "Q-learning", Machine Learning, 8, pp.279-292, 1992.

[3] R.R. Yagar and L.A. Zadeh, ed., Fuzzy Sets, Neural Networks, and Soft Computing, Thomson Publishing, NY, 1994.

[4] A. Dubrawski and J. L. Crowley, "Self-supervised neural system for reactive navigation", Proc. IEEE ICRA-94, pp2076-2082, 1994.

[5] A. Dubrawski and P. Reingnier, "Learning to categorize perceptual space of a mobile robot using Fuzzy-ART neural network", Proc. IEEE/RSJ Int. Conf. on IROS-94, pp.1272-1277, 1994.

[6] H. Handa, A. Ninomiya, T. Horiuchi, T. Konishi and M. Baba, "An incremental state-segmentation method for reinforcement learning using ART neural network", Proc. IEEE IECON 2000, pp.2732-2737, 2000.

[7] H. Handa, A. Ninomiya, T. Horiuchi, T. Konishi, and M. Baba, "An incremental state-space construction based on the notion of contradiction for reinforcement learning", Trans. of the Society of Instrument and Control Engineers, vol.38, no.5, pp.469-476, 2002 (in Japanese).

[8] T. Kamio, S. Soga, H. Fujisaka, and K. Mitsubori, "An adaptive state space segmentation for reinforcement learning using fuzzy-ART neural network," Proc. IEEE MWSCAS 2004, vol.3, pp.117-120, 2004.