A Reinforcement Learning System with Multi-Layered Fuzzy Neural Network

Takashi Kurmeoto^{a,*}, Hiroki Matsusaka^a, Masanao Obayashi^a, Shingo Mabu^a, Kunikazu Kobayashi^b

^aYamaguchi University, Tokiwadai 2-16-1, Ube, 755-8611, Japan

^bAichi Prefectural University, Ibaragabasama, Nagakute, 480-1198, Japan

*Corresponding Author: wu@yamaguchi-u.ac.jp

Abstract

To define the state of unknown environment for intelligent agent or autonomous robot, many methods such as neural networks, linear models, decision trees, etc have been proposed in the study of reinforcement learning field. In this paper, a self-organized fuzzy neural network with multiple fuzzy inference layers (MLFNN) is proposed. The deeper fuzzy inference layer includes fuzzy membership functions and fuzzy rules as same as the normal fuzzy neural networks, however, it extracts abstract states of the input data. Goal-navigated exploration problem was used in the experiment to confirm the effectiveness of the proposed reinforcement learning system.

Keywords: fuzzy neural networks, reinforcement learning, intelligent agent, autonomous robot.

1. Introduction

The intelligent agent or autonomous robot are defined as that these artifacts have the self-learning abilities to adapt to the unknown or unstable environments. To design these artificial intelligent models, reinforcement learning (RL) method ⁽¹⁾, which is a kind of machine learning field, is a powerful tool and showed its attractive achievements recently ⁽²⁾⁽³⁾. In May, 2017, an artificial intelligence (AI) software AlphaGo ⁽³⁾ used RL and decision tree (DT) methods (developed by Google's DeepMind Ltd.) won the strongest human player and professional teams of Go game at China completely.

RL realize adaptive action acquisition according to the exploration and exploitation trials. There are 4 principle elements in RL: state, action, policy, and reward. In RL, a learner observes the state of environment, outputs its action

according to the policy (generally stochastic), and obtain the reward/punishment from the environment (the transition process of the states). In this paper, we concentrate the study of state identification. The original information observed by the learner need to be categorized as states of RL, and there have been many proposals such as neural networks ⁽¹⁾⁻⁽³⁾, linear models ⁽¹⁾, decision trees ⁽³⁾, fuzzy inference systems ⁽¹⁾⁻⁽¹⁰⁾, random tiling ⁽¹¹⁾, etc. In this paper, we propose a novel multi-layered fuzzy neural network (MLFNN) based on the conventional self-organized fuzzy neural network (SOFNN) proposed in our previous works ^{(5) (6)-(8)-(10)}. The main aim of the proposed MLFNN is to reduce the dimensionality of the states inference by the SOFNN. In MLFNN, some abstract states, which categorized the states given by SOFNN, are designed by designer. The experiment of goal-navigated exploration problem was performed and the effectiveness of the proposed MLFNN was confirmed.

2. The Proposed System: MLFNN

2.1 SOFNN

To identify the state of the environment, a self-organizing fuzzy neural network (SOFNN) was proposed ^{(5) (6)-(8)-(10)}.

For an n -dimension input data $\mathbf{x}(x_1(t), x_2(t), \dots, x_n(t))$ at time t , a Gaussian membership function $B_i^k(x_i(t))$ is defined by Eq. (1).

$$B_i^k(x_i(t)) = \exp \left\{ -\frac{(x_i(t) - c_i^k)^2}{2\sigma_i^{k2}} \right\} \quad (1)$$

Fig. 1. A RL system with FNN ⁽¹⁰⁾

Here c_i^k , σ_i^k denotes the mean and the deviation of i th membership function, which can be interpreted as the center and the width in the RBF nodes, respectively.

Let $K(t)$ be the largest number of fuzzy rules R^k (k

$= 1, 2, \dots, K(t)$), then we have Eq. (2) for R^k :

If ($x_1(t)$ is $B_1^k(x_1(t))$, ..., $x_n(t)$ is $B_n^k(x_n(t))$) then

$$\phi_k(x_i(t)) = \prod_{i=1}^n B_i^k(x_i(t)) \quad (2)$$

where $\phi_k(x(t))$ is output of the rule R^k for the input $x(t)$, $k = 1, 2, \dots, K(t)$.

To decide the number of membership functions and fuzzy rules, a self-organizing algorithm was proposed as follows ⁽⁵⁾.

Step 1. For each dimension in the input space, only one membership function is generated by the first input data, the value of its membership's mean equals to the value of input data, and the value of deviation of all Gaussian function units is fixed to an empirical value.

Step 2. For the next input $\mathbf{x}(x_1(t), x_2(t), \dots, x_n(t))$,

a new membership function is generated if Eq. (3) is satisfied.

$$\max_s B_{i,s}(x_i(t)) < F \quad (3)$$

Here $B_{i,s}(x_i(t))$ denotes the value of existed membership functions calculated by (1) and $s = 1, 2, \dots, L_i(t)$ indicates the s th membership function, the maximum number is $L_i(t)$. F denotes a threshold

value to judge whether the input state is identified precisely by existing membership functions.

Step 3. A new rule is generated automatically when a new membership function is added according to Eq. (2). The connections between all membership functions and the new rule are added.

Iteratively, the Fuzzy net is completed to adapt to all input states and can be reconstructed when the new input appears.

2.2 MLFNN

In MLFNN, here let the output of fuzzy rule $\phi_k(x(t))$ be a part of input to the second fuzzy net. Additionally, a set of membership function for the abstracted states is designed. Similar to Eq. (1), the Gaussian membership functions $B_k^p(x_k(t))$, $p = 1, 2, \dots, P$ are

designed and parameters c_k^p , σ_k^p are set by prior knowledge.

$$B_k^p(x_k(t)) = \exp \left\{ - \frac{(d_k(x_k(t)) - c_k^p)^2}{2\sigma_k^{p2}} \right\} \quad (4)$$

where $d_k(x_k(t))$ indicates the distance between the goal and the current state, $p = 1, 2, \dots, P$ is the number of fuzzy membership function in the second fuzzy net.

For example, in the goal exploration problem, c_k^p can be the distance of positions in areas nearby the goal, middle

Fig. 2. The structure of RL with MLFNN

range to the goal, or far from the goal, and σ_k^p can be a constant.

The fuzzy rules of the second fuzzy net are defined as follows:

If $(x_1(t)$ is $B_1^p(x_1(t))$, ..., $x_n(t)$ is $B_K^p(x_K(t))$) then

$$\Phi_p(x_k(t), \phi_k(t)) = \sum_{k=1}^K \phi_k(t) B_k^p(x_k(t)) \quad (5)$$

where $\phi_k(t)$ is output of the rule R^k for the output of the first fuzzy net $\phi_k(t)$.

2.3 A RL system with MLFNN

As same as the reinforcement learning (RL) system with SOFNN (See Fig. 1), a RL system with MLFNN (See Fig. 2) can be composed by the fuzzy net and Temporal Difference (TD) learning algorithm, e.g., Q-learning and Sarsa learning.

The output of the fuzzy net are connected to the state-action value function $Q_t(s_t, a_t)$.

$$Q_t(s_t, a_t) = \frac{\sum_{p=1}^P w_{pq}^{Q_t} \Phi_p}{\sum_{p=1}^P \Phi_p} \quad (6)$$

where $q=1, 2, \dots, Q$ is the number of available actions.

The action selection policy is a Boltzmann distribution function:

$$p(a_t = a_j | s_t) = \frac{\exp(Q_t(s_t, a_j)/T)}{\sum_m \exp(Q_t(s_t, a_m)/T)} \quad (7)$$

where T is a constant effecting the entropy of the stochastic policy. The learning rule of MLFNN in the case of Sarsa learning is given as follows (proved in reference (10)):

$$w_{pq}^{Q_t}(t+1) = w_{pq}^{Q_t}(t) + \begin{cases} \alpha_w \varepsilon_{TD} \Phi_p(x(t)) & a_t = a_j \\ 0 & o t h e r \end{cases} \quad (8)$$

$$\varepsilon_{TD} = r_{t+1} + \gamma \cdot Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \quad (9)$$

where $0 < \alpha_w < 1$, r_{t+1} , γ are the learning rate, the reward after action a_{t+1} , and the discount, respectively.

In reference (8)-(10), the suitable distance between multiple agents was used as a kind of swarm intelligence⁽¹²⁾ to improve the learning performance. It is realized by giving positive/negative reward into Eq. (9), when the distance between multiple independent learners is suitable/not suitable. When the suitable/not suitable distance is considered into the reward, the learning method is called “swarm learning”, and oppositely “no swarm learning”. We also confirmed the efficiency of swarm learning in the case of RL with MLFNN by the experiment results. Furthermore, in reference (7) and (10), learning rate in Eq. (8) was adjusted adaptively according to the visited times of the states. More visited state uses smaller value of learning rate.

Fig. 3. Environment used in the goal-navigated problem experiment

3. Experiments

A goal-navigated problem (Fig. 3) was used in the computer simulation experiment. In Fig. 3 (a), there are 2 agents a, and b at the start positions. The exploration 2-D space has a size of 16x16, and a 4x4 obstacle area in the center of the exploration square, a 4x4 goal area in the opposite location of the start position. The observed data (input to the RL system) were coordinates positions of agents, and the action of agents were the 4 direction movements with 1 step 1 distance (i.e., the output of the RL system with discrete space). In Fig. 3 (b), there are 3 middle goal areas designed for MLFNN's second fuzzy layer. The distances from the centers of R^1 , R^2 , R^3 to the goal are $c_k^1=4.95$, $c_k^2=13.52$, $c_k^3=17.68$, respectively. There were

3 kinds of RL methods were compared in the experiments: Q-table ⁽¹⁾, FNN ⁽¹⁰⁾, and MLFNN (proposed here).

Parameters used in the experiments are shown in Table 1.

In Fig. 4, the last explored paths of agents learned by RL with MLFNN were shown. The trajectories of 2 agents kept nearly in the case of swarm learning as shown in Fig. 4 (a) and separately in the case of no swarm learning (Fig. 4(b)). In Fig. 5, the change of the lengths of agents in trials was shown. According to the iterations (horizontal axis indicates trials or iterations), the lengths were shorten in both cases of swarm learning and no swarm learning, however, swarm learning showed better convergence.

Table 1. Parameters used in the experiment.

Parameter name	Symbol	Value
Number of agents	-	2
Input dimensionality	n	2
Discount rate	γ	0.8
Temperature constant	T	0.8→0.2
Variance in Fuzzy MF	σ_i^k, σ_k^p	0.4
Reward of swarm	r_{swarm}	1.0
Reward of no swarm	$r_{no\ swarm}$	-Distance
Minimum distance limitation between agents for swarm learning	Min_Dis	1.0
Maximum distance limitation between agents for swarm learning	Max_Dis	5.0
Reward for Goal	r_{goal} (deepest)	100.0 (200.0)
Reward for Middle Goals	r_{mid_goal}	100.0
Reward for obstacle and wall crash	r_{wall}	-1.0

(a) swarm learning result

(b) swarm learning result

Fig. 4. Paths after learning by the proposed method.

Fig. 5. The learning performance comparison between swarm learning and independent (no swarm) learning.

The change of the number of membership functions of the first layer of MLFNN is shown in Fig. 6, and the number of fuzzy rules of the first layer is shown in Fig. 7. As shown in Fig. 6, both agents explored the maximum height and the maximum width of the environment (16x16) after 400 trials. And from Fig. 7, it can be found that agent 2 experienced fewer positions than agent 1. The learning performance comparison between Q-table, RL with FNN, RL with MLFNN was given by Fig. 8. The proposed method, RL with MLFNN, showed its priorities in both cases of swarm learning and no swarm learning. The average learning costs of 10 experiments (1000 trials per experiment) of these methods are shown in Fig. 9. Swarm learning showed its priority in all cases of methods, and swarm learning of RL with MLFNN had the best performance.

Fig. 6. The change of the number of membership functions during learning.

Fig. 7. The change of the number of fuzzy rules in the first layer of MLFNN during learning.

4. Conclusions

To solve the problem of state explosion (the curse of dimensionality) in the field of reinforcement learning (RL) for multi-agent system (MAS), a multi-layered fuzzy neural network (MLFNN) was proposed in this paper. Based on the self-organizing fuzzy neural network (SOFNN), MLFNN constructs its first fuzzy inference layer to categorize input data to kinds of states, and by inserting a fuzzy inference layer for abstract states (e.g., middle goals), MLFNN outputs fewer states to the state-action value function (Q function), and it results the improvement of the learning performance. Goal-navigated problem experiments showed the outstanding efficiency of the proposed method comparing to the conventional Sarsa learning with Q-table or SOFNN. The future works of this study is to design an automatic mechanism to find the abstract states, and it may also belong to a RL way.

(a) no swarm learning

(b) swarm learning

Fig. 8. The learning performance comparison between different methods

Fig. 9. Average learning costs (steps) of different methods.

Acknowledgment

This work was supported by JSPS KAKENHI Grant No. 26330254 and No.25330287.

References

- (1) R.S. Sutton, and A.G. Barto., "Reinforcement learning: An introduction", The MIT Press, 1998
- (2) V. Mnih, et al., "Human-level control through deep reinforcement learning", Nature, Vol. 518, pp. 529-533, 2015
- (3) D. Silver, et al., "Mastering the game of Go with deep neural networks and tree search", Nature, Vol. 529, pp. 484-489, 2016
- (4) L. Jouffe, "Fuzzy inference system learning by reinforcement learning," IEEE Transaction on System, Man and Cybernetics-B, Vol.28, No.3, pp.338-355, 1998
- (5) M. Obayashi, A. Iseki, and K. Umesako, "Self-organized reinforcement learning using fuzzy inference for stochastic gradient ascent method", Proceedings of the International Conference on Control, Automation and Systems (ICCAS2011), pp.735-738, 2001
- (6) T. Kuremoto, M. Obayashi, and K. Kobayashi, "Nonlinear prediction by reinforcement learning", Lecture Note in Computer Science, Vol. 3644, pp.1085-1094, 2005
- (7) X. S. Wang, Y. H. Cheng, and J. Q. Yi, "A fuzzy actor-critic reinforcement learning network," Information Sciences, Vol.177, pp.3764-3781, 2007
- (8) T. Kuremoto, M. Obayashi, K. Kobayashi, H. Adachi, and K. Yoneda, "A reinforcement learning system for swarm behaviors," Proceedings of IEEE World Congress on Computational Intelligence (WCCI/IJCNN), pp.3710-3715, 2008
- (9) T. Kuremoto, M. Obayashi, K. Kobayashi, "Adaptive swarm behavior acquisition by a neuro-fuzzy system and reinforcement learning algorithm," International Journal of Intelligent Computing and Cybernetics, Vol.2, No.4, pp.724-744, 2009.
- (10) T. Kuremoto, Y. Yamagno, L.B. Feng, K. Kobayashi, M. Obayashi, "Adaptive swarm behavior acquisition using a neuro-fuzzy reinforcement learning system," (in Japanese), IEEJ Transactions on EIS, Vol.131-C, No.6, pp.1076-1085, 2013
- (11) H. Kimura, "Reinforcement learning in multi-dimensional state-action space using random tiling and Gibbs sampling", Transactions of the Society of Instrument and Control Engineers, Vol. 42, No. 12, pp. 1336-1343, 2006
- (12) C.W. Reynolds, "Boids background and update," <http://www.red3d.com/cwr/boids/>