Influence of the Chaotic Property on Reinforcement Learning using a Chaotic Neural Network

Yuki Goto and Katsunari Shibata

Dept. of Electrical and Electronic Engineering, Oita University,
700 Dannoharu, Oita 870-1192, Japan
iwishdayss@gmail.com,shibata@oita-u.ac.jp

Abstract. Aiming for the emergence of higher complicated dynamic function such as "thinking", our group has set up a hypothesis that internal chaotic dynamics in an agent's chaotic neural network grows from "exploration" to "thinking" through reinforcement learning, and proposed a new learning method for that. However, even after learning in a simple obstacle avoidance task, the agent sometimes moved irregularly and collided with the obstacle. By reducing the scale of the recurrent connection weights, which is expected to have a deep relation to the chaotic property, the problem was reduced. Then in this paper, the learning performance depending on the recurrent weight scale is observed. The scale has an appropriate value as can be seen in FORCE learning in reservoir computing.

Keywords: reinforcement learning, chaotic neural network, emergence of intelligence, obstacle avoidance, chaotic property

1 Introduction

Aiming for artificial general intelligence (AGI), our group has proposed the end-to-end reinforcement learning approach in which a neural network (NN) is responsible for the entire process from sensors to motors and various functions emerge in it through reinforcement learning (RL)[1–3]. Recently, in the same approach, the DeepMind group has shown the impressive result in TV games[4] or game of "Go"[5]. This supports the significance of our approach.

From the viewpoint of higher functions, it is obvious that not only static mapping from sensor signals to motor commands but also internal dynamics should be acquired through learning. As a kind of internal dynamics, we have shown that memory-required functions emerge through RL by using a recurrent neural network (RNN)[6, 7], but the acquired dynamics are limited mainly in fixed-point convergence dynamics. However, a typical higher function such as "thinking" needs autonomous but rational transition in the internal state.

On the other hand, exploration that is essential for autonomous learning should be random-like, but is similar to thinking with respect to the dynamics with autonomous state transitions. When we stand up before a fork on a road, we explore to choose one of the ways while considering many things such as the road condition or traffic sign, and such exploration is not completely random on the motor-level, but past learning is reflected on it. That suggests us that "exploration" and "thinking" cannot be separated explicitly. According to the consideration, we have set up a hypothesis that "exploration" grows into "thinking" by forming rational non-converging attractors through learning on the chaotic internal dynamics in a chaotic neural network (ChNN). In this framework, "inspiration" or "discovery" can be expected to emerge as unexpected state transitions like "chaotic itinerancy" observed in associative memory[8]. It is also expected that exploratory behavior is autonomously resumed in unknown situations. To realize the learning using a ChNN, we have proposed a new RL method in which external random noises are not used anymore[2, 9, 10].

To show that the new RL method works, we have shown that an agent could learn in several goal-directed tasks before challenging the learning of "thinking"[2, 9, 10]. In an obstacle avoidance task using a wheel-type robot and visual sensors, the agent could learn to reach the goal while avoiding the obstacle[10]. However, even after learning, it was observed that the agent sometimes made irregular motions suddenly or collided with the obstacle and trapped there for a while. To solve the problem, the scale of the recurrent connection weights in the ChNN, which is expected to have a deep relation to the chaotic property, was reduced, and the problem was actually reduced. Then in this paper, the learning performance depending on the recurrent weight scale is observed.

On the other hand, FORCE learning that is a kind of supervised learning using a reservoir can learn to generate complex dynamics easily and rapidly[11]. It was reported that the scale of recurrent connection weights influenced the learning performance, and the relation to the chaotic property in the reservoir network was discussed. It has been also shown that the reservoir can be learned from reward-like signals[12, 13]. Then, referring to the result in FORCE learning, we discuss our results in relation to the chaotic property.

2 Reinforcement Learning (RL) using a Chaotic Neural Network (ChNN)

RL is autonomous and purposive learning to get more reward and less punishment. In general RL, an agent explores stochastically using external random noises, but here, it explores by chaotic dynamics generated inside its ChNN without adding random noises.

To deal with continuous motions, Actor-Critic is used as a RL architecture. To isolate the critic from the chaotic dynamics, the ChNN is used for the actor and a regular layered NN is used for the critic as shown in Fig.1. The Actor ChNN outputs $\mathbf{A}(\mathbf{S}_t)$ are used as motion signals, and the Critic NN output $V(\mathbf{S}_t)$ is used as state value where $\mathbf{S}_t$ is the sensor input vector at time t. The neuron model used in both NNs is static as

$$u_{j,t}^{l} = \sum_{i=1}^{N^{l-1}} w_{j,i}^{l} x_{i,t}^{l-1} \left(+ \sum_{i=1}^{N^{l}} w_{j,i}^{FB} x_{i,t-1}^{l} \right) \tag{1}$$

where $u_{j,t}^{l}$ and $x_{j,t}^{l}$ are the internal state and the output of the j-th neuron in the l-th layer at time t, and $w_{j,i}^{l}$ is the connection weight from the i-th neuron in the $(l-1)$-th layer to the j-th neuron in the l-th layer. The second term in the right-hand side is only for the hidden layer in the actor ChNN, and $w_{j,i}^{FB}$ is the recurrent connection weight from the i-th neuron to j-th neuron in the hidden layer. All the weights are decided randomly. In this paper, the scale of $\mathbf{w}^{FB}$, which is the size of the symmetric uniform random number, is varied and the difference in learning performance is observed. The activation function is the sigmoid function $f()$ whose value ranges from -0.5 to 0.5, and the output is $x_{j,t}^{l} = f(u_{j,t}^{l})$.

For learning, TD-error $\hat{r}_t$ is represented as

$$\hat{r}_t = r_{t+1} + \gamma V(\mathbf{S}_{t+1}) - V(\mathbf{S}_t) \tag{2}$$

where r_{t+1} is the reward given at time $t+1$, γ is a discount factor. T_{V_t}, which is the training signal for the critic output at time t, is computed as

$$T_{V_t} = V(\mathbf{S}_t) + \hat{r}_t = r_{t+1} + \gamma V(\mathbf{S}_{t+1}). \tag{3}$$

The critic NN is trained once according to Error Back Propagation using T_{V_t}.

In the proposed method, there is no external random noises added to the actor outputs. Only the connection weights $w_{j,i}^{l}$ from inputs to hidden neurons or from hidden neurons to output neurons are trained. The weight $w_{j,i}^{l}$ in the ChNN is modified using the causality trace $c_{j,i,t}^{l}$ and a learning rate η as

$$\Delta w_{j,i,t}^{l} = \eta \hat{r}_t c_{j,i,t}^{l} \tag{4}$$

where $\Delta w_{j,i,t}^{l}$ is the update of the weight $w_{j,i}^{l}$ at time t. The trace $c_{j,i,t}^{l}$ is put on each connection, and takes in and maintains the input through the connection according to the change in its output $\Delta x_{j,t}^{l} = x_{j,t}^{l} - x_{j,t-1}^{l}$ as

$$c_{j,i,t}^{l} = \left(1 - |\Delta x_{j,t}^{l}|\right) c_{j,i,t-1}^{l} + \Delta x_{j,t}^{l} x_{i,t}^{l-1}. \tag{5}$$

3 Simulation

In this paper, the same obstacle avoidance task as in [2, 10] is simulated. In this simulation, as shown in Fig.1, there is a 20 × 20 field, and a goal is fixed at the upper center area (0,5) with radius $r = 1.0$. An agent ($r = 0.5$) and an obstacle ($r = 1.5$) are located randomly at the beginning of every episode. Each of the 2 omni-directional visual sensor catches only goal or obstacle and has 72 cells, each of which has 5° receptive field. Additionally, the other 2 sensor signals indicate distance to the wall in front of or behind the agent. Total of 146 sensor signals are the inputs of both actor and critic networks ($= \mathbf{S}_t$). The right and left wheels of the agent rotate according to the 2 actor outputs ($= \mathbf{A}(\mathbf{S}_t)$) respectively.

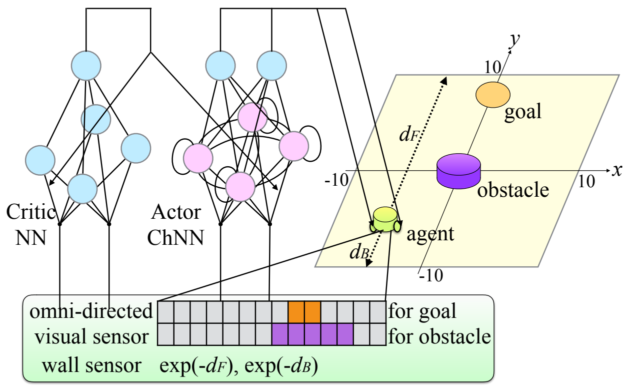


Fig. 1. Reinforcement learning system and the obstacle avoidance task in this paper

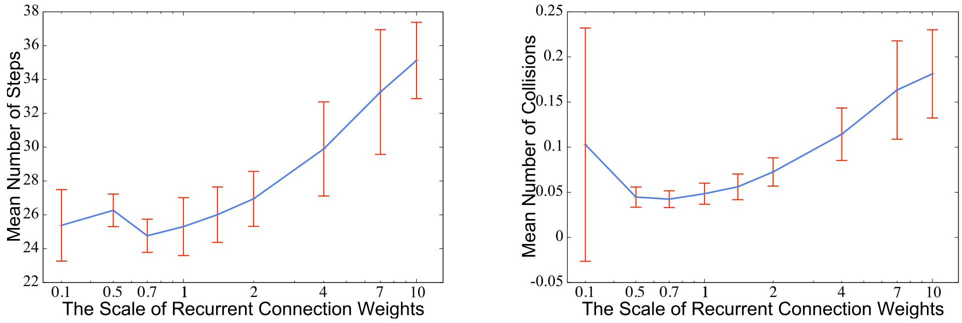
When the agent reaches the goal area, $r_t = 0.4$ is given as a reward. When it collides with the obstacle or a wall at the boundary of the field, $r_t = -0.1$ is given as a penalty. The episode is terminated when the agent either reaches the goal or fails to do so in 1,000 steps from start. The parameters used in the simulation are shown in Table 1.

Table 1. The parameters used in the simulation

Name		Actor	Critic
Number of Episodes		1,000,000	
Number of Layers		3	
Number of Hidden Neurons		100	
Gain of Sigmoid Function: g	Output	1	
	Hidden	2	1
Learning Rate: η		0.001	1.0
Range of Initial Weights $w^l_{j,i}$		[-1,1]	
Discount Factor: γ		—	0.99

The scale of $\mathbf{w}^{FB}$ was changed in 8 cases from 0.3 to 10.0, and 10 simulations were done with a different random sequence used for connection weights and the initial arrangement of agent and obstacle at each episode. In Fig.2, (a) shows the number of steps from the start of the agent to the goal and (b) shows the number of collisions with the obstacle until the agent reach the goal. Mean and standard deviation during 900,000 to 1,000,000 episodes are shown for each scale of $\mathbf{w}^{FB}$ in each graph in Fig.2. As the scale is smaller, the numbers of steps and collisions tend to decrease, and the both becomes minimum when the scale is 0.7. However, when the scale is less than 0.7, they are larger than the case of 0.7, and the standard deviations for them also become larger. Furthermore, although not shown in Fig.2, when the scales is 0.1, the agent could not sufficiently explore the field and finally stopped moving.

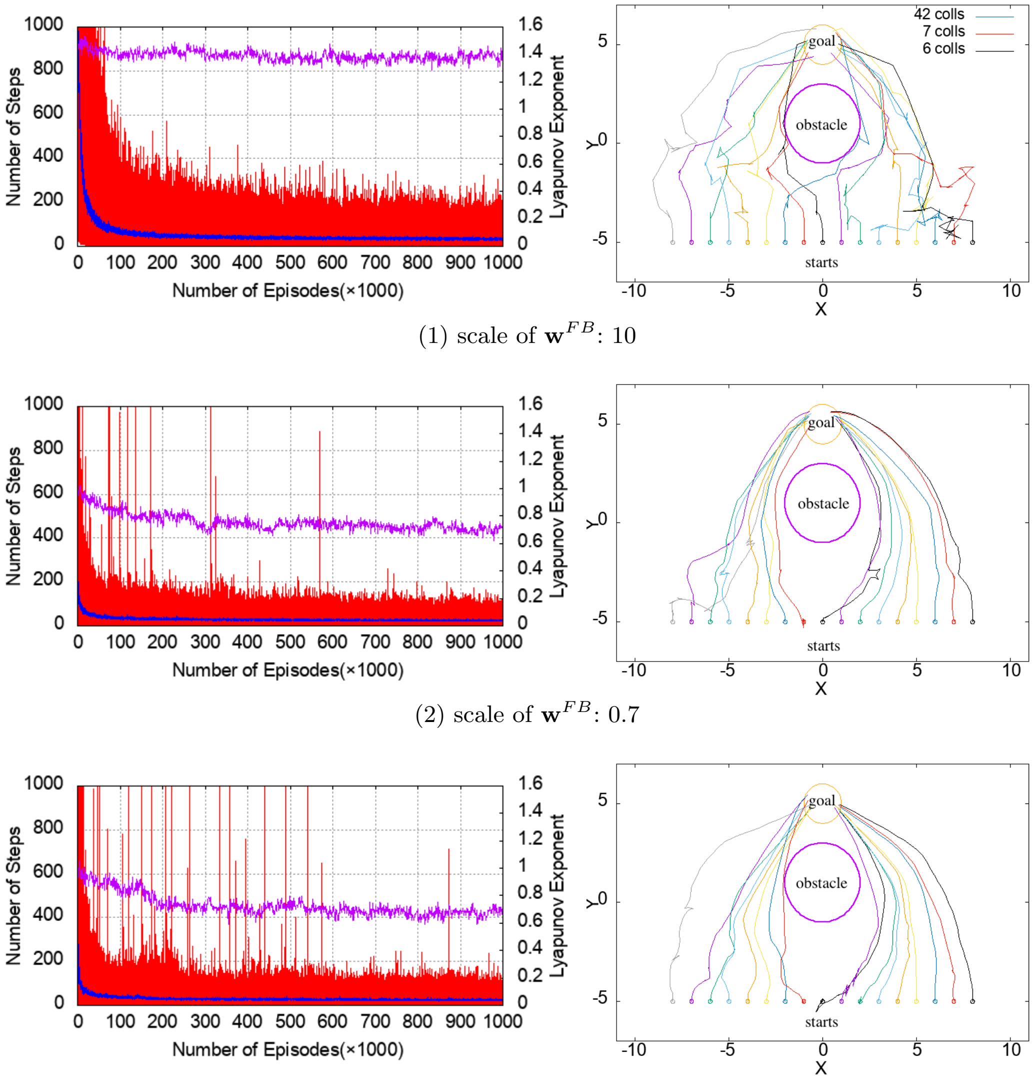
Then, three cases, in which the scale is 10, 0.7 or 0.5, are picked up and the details are shown as follows. In Fig.3, (a) shows learning curve and (b) shows sample trajectories. In (a), the red and blue traces show the number of steps to reach the goal at every episode and average steps for every 100 episodes respec-



(a) steps from start to the goal

(b) collisions with the obstacle or wall

Fig. 2. Change in the learning performance according to the scale of recurrent connection weights $\mathbf{w}^{FB}$



(1) scale of $\mathbf{w}^{FB}$: 10

(2) scale of $\mathbf{w}^{FB}$: 0.7

(3) scale of $\mathbf{w}^{FB}$: 0.5

(a) learning curve (red, blue: average) and pseudo-Lyapunov exponent (magenta)

(b) sample agent trajectories after learning

Fig. 3. Comparison of learning performance among 3 scales of $\mathbf{w}^{FB}$

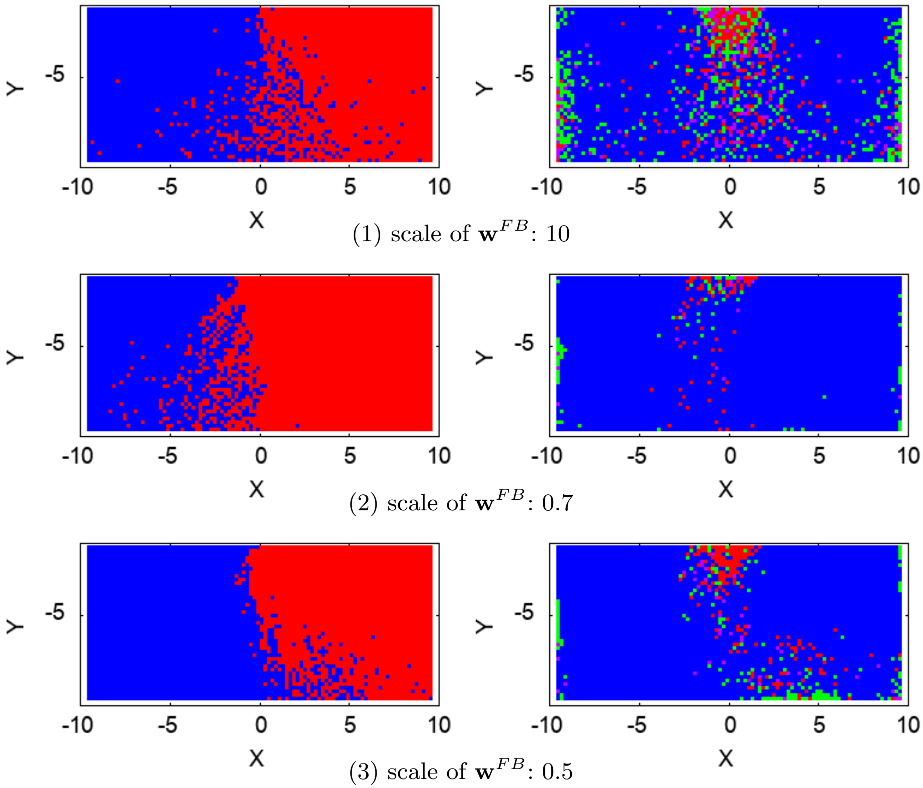
tively. The magenta trace shows the change in the pseudo Lyapunov exponent which is an index of chaotic property of the system including the loop with the environment, for every 1,000 episodes. The exponent shows the sensitivity to small perturbations. When it is positive, the dynamics is likely to be chaotic. In this paper, every 1,000 episodes, a random vector whose size is normalized to 0.001 is added to internal state of the hidden neurons in the ChNN. After five-step action according to $\mathbf{A}(\mathbf{S}_t)$, the Euclidean distance d of the hidden states from the case when no perturbation is added was compared between before and after the action. The above is performed in 51 situations in which the agent location varies as $x = -8, -7, \cdots, 8, y = -5$ as shown in Fig.3(b) and the obstacle location varies $x = -5, 0, 5, y = 1$. Pseudo Lyapunov exponent λ is calculated as

$$\lambda = \frac{1}{51}\sum_{p=1}^{51}\frac{1}{5}\sum_{t=1}^{5} ln\frac{d_t^{(p)}}{d_{t-1}^{(p)}}. \tag{6}$$

To observe the agent behavior after learning, the goal and obstacle are located at (0,5) and (0,1) respectively, and the initial agent location varies as $x = -8, -7, \cdots, 8, y = -5$. The trajectories of the agent are shown in (b). Fig.3(a-1) shows that the number of steps is larger in the latter stage of learning than the cases of the other two scales. As shown in Fig.3(b-1), the agent often moves irregularly at a whole. Especially three trajectories when starting from (-1,-5), (0,-5) and (1,-5), it collided with the obstacle many times. Additionally in (a-1), the pseudo-Lyapunov exponent is as large as around 1.4 during learning. It is thought that the hidden neurn outputs were in the saturation area of sigmoid functions and they changed suddenly by the chaotic property that could not be suppressed during learning. When the scale is 0.7 (in (b-2)), the agent moves smoothly at a whole. As shown in Fig.3(a-3) when the scale is 0.5, the agent sometimes could not reach the goal and the episode failed. The change in the exponent in (a-2) and (a-3), are similar and they decreases slowly as the progress of learning.

Fig.4 shows how the agent behavior varies depending on the initial agent location in the area $y < -1$ where the agent is located father than the obstacle from the goal. (a) shows distribution of the initial agent location from which the agent passed the left side or the right side of the obstacle to reach the goal. (b) shows frequency distribution of collisions with the obstacle or the wall for each initial agent location. In (a), the agent is likely to pass through the left side of the obstacle when the initial location is in the left side part of the field, and vice versa. In (a-1) and (a-2), the boundary of the two areas appears in the front of the obstacle. As shown in (b-2), the agent reaches the goal without any collision with the obstacle in most of the entire field.

In [11], in the FORCE learning with a reservoir network, which is a kind of RNN with fixed recurrent connection weights, the scale of weights ($= g$) was also varied and it was shown that the scalse shoulde be in a range for appropriate learning. When the scale is smaller than the range, the network is not chaotic and fails to learn. By making the scale large, the network is chaotic, however, learning did not converge and failed to suppress chaotic activity when the scale



(1) scale of $\mathbf{w}^{FB}$: 10

(2) scale of $\mathbf{w}^{FB}$: 0.7

(3) scale of $\mathbf{w}^{FB}$: 0.5

(a) the side of the obstacle that the agent passed through for each initial agent location (red:right side, blue:left side)

(b) the number of collisions for each initial agent location (blue: 0, green: 1∼5, magenta: 6∼10, red: more than 10)

Fig. 4. Comparison of difference in agent behavior depending on the initial agent location for 3 scales of $\mathbf{w}^{FB}$

is too strong. There are many differences from ours; output feedback, number of neurons, sparsity of feedback connection neuron model and so on. However, in both learning, the trend for the scale of recurrent connection weights is very similar, and the significance of talented ”Edge of Chaos”[14] is suggested.

4 Conclusion

In this paper, our new reinforcement learning using a chaotic neural network was applied to an obstacle avoidance task, and with varying the scale of the recurrent connection weights, the learning performance was observed. When the scale is larger, the frequency that the agent makes irregular motions or collides with the obstacle increases even after learning. When the scale is smaller, the agent trajectories after learning become smooth, but if it is too small, the agent could not explore appropriately and failed to learn. Therefore, it is important to set the scale appropriately. That trend is very similar to the case of FORCE learning. It is suggested that the learning performance is deeply related to the chaotic property of the ChNN. It can be thought that too small scale causes lack of exploration and too large scale causes saturation of hidden neurons and irregular change due to remaining chaotic property. Even in the best result, the agent sometimes still collides with the obstacle or wall. Some further improvement and step up to learning of complicated dynamics should be done in the future.

Acknowledgement. This work was supported by JSPS KAKENHI Grant Number 15K00360.

References

1. Shibata, K.: Emergence of Intelligence through Reinforcement Learning with a Neural Network, A. Mellouk (Ed.), “Advances in Reinforcement Learning” InTech, pp.99-120 (2011)
2. Shibata, K., Goto, Y.: Significance of Function Emergence Approach based on End-to-end Reinforcement Learning as suggested by Deep Learning, and Novel Reinforcement Learning Using a Chaotic Neural Network toward Emergence of Thinking, Cognitive Studies, 24(1), pp.96-117 (2017) (in Japanese)
3. Shibata, K.: Functions that Emerge through End-to-end Reinforcement Learning – The Direction for Artificial General Intelligence –, arXiv:1703.02239v2(RLDM17)
4. Volodymyr, M., et al.: Playing Atari with Deep Reinforcement Learning, NIPS Deep Learning Workshop 2013 (2013)
5. David, S., et al.: Mastering the game of Go with deep neural networks and tree search, Nature 529, pp.484-489 (2016)
6. Shibata, K., Utsunomiya, H.: Discovery of Pattern Meaning from Delayed Rewards by Reinforcement Learning with a Recurrent Neural Network, Proc. of IJCNN 2011, pp. 1445-1452 (2011)
7. Shibata, K., Goto, K.: Emergence of Flexible Prediction-Based Discrete Decision Making and Continuous Motion Generation through Actor-Q-Learning, Proc. of ICDL-Epirob 2013, ID 15 (2013)
8. Kaneko, K., Tsuda, I.: Chaotic itinerancy, Chaos, 13(3), pp.926-936 (2003)
9. Shibata, K., Sakashita, Y.: Reinforcement Learning with Internal-Dynamics-based Exploration Using a Chaotic Neural Network, Proc. of IJCNN 2015, #15231 (2015)
10. Shibata, K., Goto, Y.: New Reinforcement Learning Using a Chaotic Neural Network for Emergence of ”Thinking” – ”Exploration” Grows into ”Thinking” through Learning –, arXiv:1705.05551(RLDM17)
11. David, C.S.: Learning in Chaotic Recurrent Neural Networks, Ph.D.Thesis, Columbia University (2009)
12. Hoerzer, G.M., Legenstein, R., Maass, W.: Emergence of complex computational structures from chaotic neural networks through reward-modulated Hebbian learning, Cerebral Cortex, Vol.24, No.3, pp. 677-690 (2014)
13. Matsuki, T., Shibata, K.: Reward-Based Learning of a Memory-Required Task Based on the Internal Dynamics of a Chaotic Neural Network, Proc. of ICONIP 2016, pp.376-383 (2016)
14. Chris, G.L.: Computation at the edge of chaos: Phase transitions and emergent computation, Physica D: Nonlinear Phenomena (Elsevier), 42(1-3), pp. 12-37 (1990)