

Visual Predictive Coding Model with Reservoir Computing for Reinforcement Learning Tasks in 3D Environment

Tomohito Izumi[†] and Yuichi Katori^{†‡}

[†]School of Systems Information Science, Future University Hakodate
 116-2 Kamedanakano-cho, Hakodate, Hokkaido 041-8655, Japan

[‡]International Research Center for Neurointelligence (IRCN), The University of Tokyo,
 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan,
 Email: katori@fun.ac.jp

Abstract—Computational models are an indispensable tool for understanding the mechanism of the brain. Previous studies employ reservoir computing to construct a predictive coding model and have shown its ability to replicate the brain’s properties. However, reservoir computing models cannot be directly applied to broad brain functions due to performance limitations. Here, we propose a visual predictive coding model with reservoir computing that can handle the high-dimensional input to extend the scope of the application. We confirmed that our model could solve reinforcement learning tasks in a three-dimensional environment and that visual images can be reconstructed from the prediction by the reservoir. We believe that our approach presents a novel dynamical mechanism of visual processing in the brain and fundamental technology for a brain-like artificial intelligence system.

1. Introduction

Mammalian’s visual system has essential functions to understand the surrounding environment, such as object recognition and scene understanding. In addition, numerous brain functions involve visual systems as an indispensable component. Hence, understanding the underlying mechanism of the visual system is the key to implementing efficient artificial intelligence and curing neural diseases.

Predictive coding is a generally accepted theory in neuroscience that the different brain areas compose a hierarchical generative model. In this theory, the brain acquired a model of the external world since each brain area predicts the state of the lower area or sensory information, and areas learn to minimize the prediction error. The predictive coding with reservoir computing (PCRC) model employs the reservoir computing framework to implement that theory. Reservoir computing (RC) [5] is one of the recurrent neural networks’ architectures and has recently gained much attention for its computational efficiency. Several studies have extended the PCRC model and shown the capability to model the brain functions in its internal dynamics [10, 12].

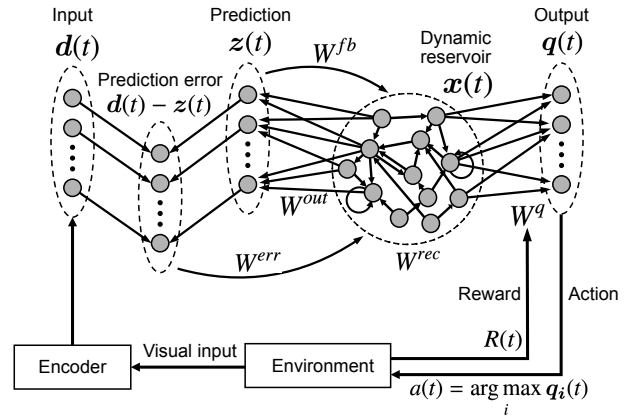




Figure 1: Schematic illustration of the model

Besides, deep learning models have been widely applied as brain models since they could replicate broad brain functions with superb performance. Yamins and DiCarlo suggested that the model’s objective is essential in building a deep learning model of the sensory cortex because the model must be effective as the brain at solving the tasks [11]. Indeed, the machine learning model for a definite purpose, such as classification, achieved great success in replicating sensory cortex properties [6, 3]. However, these models are not appropriate for revealing detailed computation of the system and processing time-varying information.

The PCRC model seems suitable for studying sensory systems since RC has a simple architecture and performs well on nonlinear dynamical modeling. Also, the implementation of the reservoir by spiking neurons suggests that idea [8]. However, stimuli in previous studies are confined to low-dimension inputs due to the reservoir’s limited performance. Thus, the range of visual systems’ characteristics that the PCRC model can handle is narrow compared to deep learning models.

In order to address this problem, this paper proposes an extension of the PCRC model capable of processing high-dimensional input by applying a variational autoencoder. We examine the model performance on behavioral tasks in a three-dimensional environment. As a relevant study,

ORCID iDs First Author:  0000-0002-9593-6930, Second Author:
 0000-0003-2773-0786

there is a convolutional reservoir computing model [2] that utilizes a fixed random-weight CNN to reduce the dimension of inputs. However, because of using CNN, the model cannot be used as a generative model.

2. Model

2.1. The model architecture

Figure 1 shows the schematic illustration of our model, which consists of an encoder and the reservoir. The encoder compresses raw visual input into low-dimensional features to make the reservoir capable of doing vision tasks in a three-dimensional environment. We took this encoder from a variational autoencoder pre-trained by the COCO Dataset [7], which includes over 10,000 natural images. The reservoir works in a predictive coding manner based on Fukino et al’s model [4]. The internal dynamics of the reservoir is governed by following equations.

$$\mathbf{x}(t+1) = (1-\tau)\mathbf{x}(t) + \frac{1}{\tau} \left(W^{rec} \mathbf{y}(t) + W^{fb} \mathbf{z}(t) + W^{err} (\mathbf{d}(t) - \mathbf{z}(t)) \right), \quad (1)$$

$$\mathbf{y}(t) = \tanh \mathbf{x}(t), \quad (2)$$

where $\mathbf{x}(t) \in \mathbb{R}^N$ is the internal state of dynamic reservoir, $\mathbf{y}(t) \in \mathbb{R}^N$ is the activity of dynamic reservoir, $\mathbf{z}(t) \in \mathbb{R}^M$ is the prediction, $\mathbf{d}(t) \in \mathbb{R}^M$ is the input from the encoder, $W^{rec} \in \mathbb{R}^{N \times N}$ is the recurrent weight matrix, $W^{fb} \in \mathbb{R}^{N \times M}$ is the prediction feedback weight matrix, and $W^{err} \in \mathbb{R}^{N \times M}$ is the prediction error feedback weight matrix. W^{fb} , W^{err} , and W^{rec} are fixed at sparse and random values.

The prediction is obtained by

$$\mathbf{z}(t) = W^{out} (t-1) \mathbf{x}(t), \quad (3)$$

where $W^{out} \in \mathbb{R}^{M \times N}$ is the prediction weight matrix that is updated by FORCE procedure [9].

2.2. Reinforcement learning

The model determines action as the index of maximum elements of $\mathbf{q}(t)$ with probability $1 - \varepsilon(t)$ and otherwise the model takes a random action.

$$a(t) = \arg \max_i \mathbf{q}_i(t), \quad (4)$$

where $\mathbf{q}(t) \in \mathbb{R}^A$ is the value of taking action, which is estimated by following equation.

$$\mathbf{q}(t) = W^q (t-1) \mathbf{y}(t), \quad (5)$$

$W^q(t) \in \mathbb{R}^{A \times N}$ is updated by online reinforcement learning algorithm [13] as follows.

$$W_a^q(t+1) = W_a^q(t) + \eta(t) (R(t) + \gamma q_a(t+1) - q_a(t)) \mathbf{y}(t) \quad (6)$$

where $R(t)$ is the reward obtained from the environment, $\eta(t)$ is the learning rate changed according to success rate of the last 100 episodes, and γ is the discount rate.

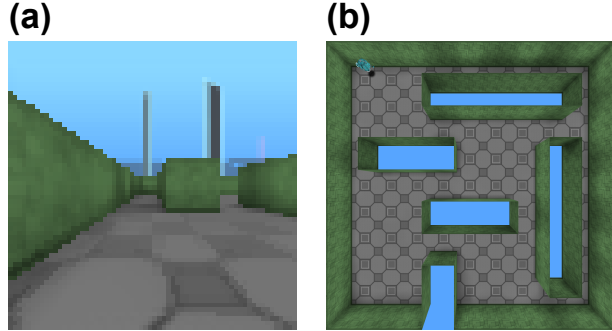


Figure 2: Example of the environment (a) Agent’s view (b) Top-down view of the environment.

3. Experiments

We evaluate the model’s performance on two tasks inspired by the function of the ventral and dorsal streams.

3.1. Location task

This task requires an agent to reach a specific location determined when the map is created. Contrary to the goal’s location, the agent spawn from a random location in each episode. The agent gets one reward if they reach the goal and otherwise zero. To behave efficiently, the agent should possess a map-like spatial representation of the environment.

3.2. Object task

This task requires an agent to fetch a target object. The target’s location and the agent’s spawn location are randomly determined at the onset of each episode. The agent gets one reward if they reach the object and otherwise zero. Having the representation of the target object is adequate for the agent to achieve the task.

3.3. Environment

We use DeepMind Lab platform [1] for generating environments in this paper. Figure 2 shows the example of the agent’s view and top-down view of the environment. The environment layout is randomly generated at each game depending on the seed, and specific constraints such as map size and maximum room number. Since an identical method generates the environment layout for both tasks, the difference between them is limited to the objective.

4. Results

Figure 3 shows the average reward the agent acquired during 5000 episodes. In both tasks, the agent learned to obtain a certain amount of reward. This result shows that the model can process the first-person view in a three-dimensional environment to select actions.

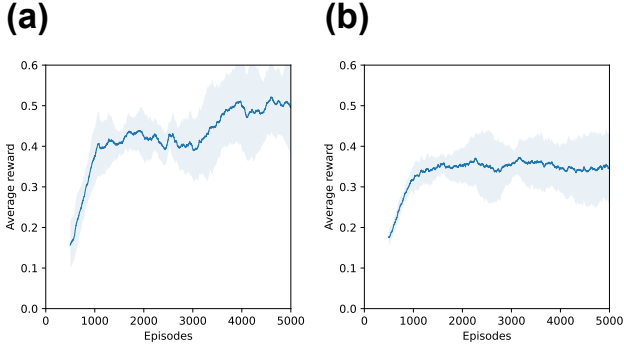


Figure 3: Average reward during 5000 episodes. Blue line represents the mean of 5 games and light blue area represents standard deviation. (a) In the location task (b) In the object task

Figure 4 shows the agent’s trajectory in 100 episodes after training. The upper figures are for the location task, and the lowers are for the object task. The pattern of behavior differs between the two tasks. In the location task, the agent focused on exploring the room and transitioned between rooms with low probability. In the object task, agents transitioned between rooms by moving along the walls and subsequently explored the room to obtain a reward.

Figure 5 shows visual input and model’s prediction. The left frame is the raw visual input, the center frame is the reconstructed raw visual input through the decoder, and the right frame is the reconstructed prediction of the model through the decoder. We can confirm that the model successfully predicts the input. Furthermore, the reconstructed prediction resembles raw visual input more than the reconstructed raw visual input.

5. Discussion

This paper proposed the extension of the PCRC model combined with a variational autoencoder. We examined the model’s performance on two tasks and confirmed model could learn behavior in the three-dimensional world.

However, the behavior the model learned was not optimized well. In the location task, the agent learned to explore all rooms randomly despite the most efficient way to obtain a reward is to memorize the goal location and pass the shortest pathway. The possible cause is that the model’s capacity is insufficient to possess a spatial map in internal dynamics. The model must behave as well as animals to be compared to the brain. Therefore, we need to improve the model’s performance in future work.

Acknowledgments

This paper is based on results obtained from a project, JPNP16007, commissioned by the New Energy and Industrial Technology Development Organization (NEDO),

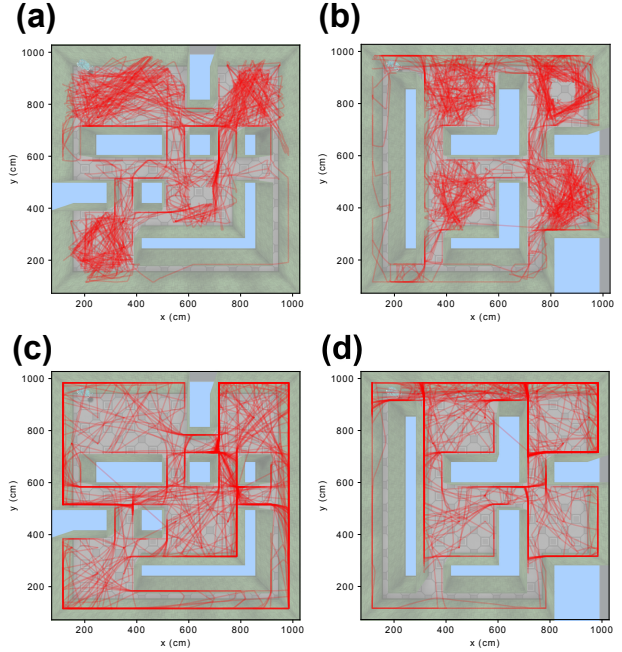


Figure 4: Agent’s trajectory in 100 episodes for (a) Location task in room A (b) Location task in room B (c) Object task in room A (d) Object task in room B



Figure 5: Visual input and model’s prediction. Left frame is the visual input. Center frame is the reconstructed visual input through the decoder. Right frame is the reconstructed prediction through the decoder.

and this work is also supported by JSPS KAKENHI (21H05163, 20H04258, 20H00596, 21H03512), and JST CREST(JPMJCR18K2), Moonshot R&D (JPMJMS2021).

References

- [1] Charles Beattie, Joel Z Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, Julian Schrittwieser, Keith Anderson, Sarah York, Max Cant, Adam Cain, Adrian Bolton, Stephen Gaffney, Helen King, Demis Hassabis, Shane Legg, and Stig Petersen. DeepMind lab. *arXiv*, cs.AI:1612.03801, December 2016.
- [2] Hantun Chang and Katsuya Futagami. Reinforcement learning with convolutional reservoir computing. *Applied Intelligence*, 50(8):2400–2410, August 2020.

- [3] Katharina Dobs, Julio Martinez, Alexander J E Kell, and Nancy Kanwisher. Brain-like functional specialization emerges spontaneously in deep neural networks. *Sci Adv*, 8(11):eabl8913, March 2022.
- [4] Miwa Fukino, Yuichi Katori, and Kazuyuki Aihara. A computational model for pitch pattern perception with the echo state network. <https://www.ieice.org/nolta/symposium/archive/2016/articles/1069.pdf>.
- [5] H. Jaeger. A tutorial on training recurrent neural networks, covering bppt, rtrl, ekf and the “echo state network” approach. *GMD Report*, 159(2):1–46, 2005.
- [6] Alexander J E Kell, Daniel L K Yamins, Erica N Shook, Sam V Norman-Haignere, and Josh H McDermott. A Task-Optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98(3):630–644.e16, May 2018.
- [7] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014.
- [8] Wilten Nicola and Claudia Clopath. Supervised learning in spiking neural networks with FORCE training. *Nat. Commun.*, 8(1):2208, December 2017.
- [9] David Sussillo and L F Abbott. Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–557, August 2009.
- [10] Hiroto Tamura, Yuichi Katori, and Kazuyuki Aihara. Possible mechanism of internal visual perception: Context-dependent processing by predictive coding and reservoir computing network. *J. Robot. Netw. Artif. Life*, 6(1):42, 2019.
- [11] Daniel L K Yamins and James J DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nat. Neurosci.*, 19(3):356–365, March 2016.
- [12] Yoshihiro Yonemura and Yuichi Katori. Network model of predictive coding based on reservoir computing for multi-modal processing of visual and auditory signals. *Nonlinear Theory and Its Applications, IEICE*, 12(2):143–156, 2021.
- [13] Yu Yoshino and Yuichi Katori. Short-term memory ability of reservoir-based temporal difference learning model. *Nonlinear Theory and Its Applications, IEICE*, 13(2):203–208, 2022.