ATLANTIS PRESS

Journal of Robotics, Networking and Artificial Life
Vol. 6(1); June (2019), *pp.* 42–47
DOI: https://doi.org/10.2991/jrnal.k.190531.009; ISSN 2405-9021; eISSN 2352-6386
https://www.atlantis-press.com/journals/jrnal

JOURNAL OF ROBOTICS, NETWORKING AND ARTIFICIAL LIFE

Research Article

Possible Mechanism of Internal Visual Perception: Context-dependent Processing by Predictive Coding and Reservoir Computing Network

Hiroto Tamura[1,*], Yuichi Katori[2,3], Kazuyuki Aihara[3]

[1]*Graduate School of Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan*
[2]*Future University Hakodate, 116-2 Kamedanakano-cho, Hakodate, Hokkaido 041-8655, Japan*
[3]*Institute of Industrial Science, The University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan*

ARTICLE INFO

Article History

Received 12 October 2018
Accepted 28 November 2018

Keywords

Visual system
perception
predictive coding
reservoir computing
context
nonlinear dynamics

ABSTRACT

The predictive coding is a widely accepted hypothesis on how our internal visual perceptions are generated. Dynamical predictive coding with reservoir computing (PCRC) models have been proposed, but how they work remains to be clarified. Therefore, we first construct a simple PCRC network and analyze the nonlinear dynamics underlying it. Since the influence of contexts is another important factor on the visual perception, we also construct PCRC networks for the context-dependent task, and observe their attractor-landscapes on each context.

1. INTRODUCTION

It is widely known that what we see is not the visual sensory input as it is. Instead, our brains integrate the sensory inputs and reconstruct the internal image in the manner we can easily understand. For example, although the actual visual input is 2D and received by both eyes, what we see is the 3D vision as one image. However, exactly how the internal visual perceptions are generated in the visual cortex has provoked much debate.

The predictive coding is one of the most accepted hypotheses on the internal perception. In the predictive coding framework, a perceived image is not merely the integrated visual sensory input, but the result of the prediction made by the internal generative model. The predictive coding also assumes that the generative model is optimized to minimize the residual error between the prediction and the actual sensory input. In particular, the hierarchical predictive coding model [1] postulates that the top-down signals from the higher-order area carry the predictions of lower-level neural activities, whereas the bottom-up signals from the lower-order area carry the residual errors between the predictions and the actual lower-level activities, so that the ascending signals have much less redundancy.

However, this model is only suitable for static visual inputs and cannot deal with temporally changing visual images, or movies. Then Fukino et al. [2] proposed the Predictive Coding with Reservoir Computing (PCRC) model, which can predict the temporally changing auditory inputs, implementing the generative model by the dynamical reservoir. Furthermore, the hierarchical PCRC models for more complex auditory inputs were proposed by Ara and Katori [3,4].

Here, the reservoir computing [5] refers to a type of the Recurrent Neural Network (RNN) approach with a simple learning strategy. When the reservoir computing networks are trained, only the output connections are modified, and the recurrent and feedback connections are fixed with randomly given values.

However, precisely how these PCRC models [2–4] work largely remains to be clarified. Moreover, these conventional models cannot perceive unlearned inputs. In addition, they are not exactly driven by the prediction error but by the sum of the error and their own prediction, which is equal to the original sensory input.

In this study, therefore, we first modify them and construct a simple one-layer PCRC network exactly driven by the prediction errors, which can perceive even unlearned inputs. Then we analyze the nonlinear dynamics underlying the trained network, in order to clarify the mechanism of the behavior.

The influence of contexts, which refers to situations, goals, and relevant past experiences, is another important factor on the visual perception. For example, even identical sensory stimuli can result in very different perceptions depending on contexts. Indeed, RNN models for context-dependent tasks have been proposed [6].

[*] *Corresponding author. Email: hiroto0324@sat.t.u-tokyo.ac.jp*

Therefore, we also construct a PCRC network for a simple context-dependent perception task. We analyze the trained network again, in order to reveal how the network perceives the sensory stimuli on each context. We further construct a PCRC network which can perceive more high-dimensional visual inputs, in order to show that the proposed network can be a possible mechanism of the visual perception. We observe that the mismatch between the context and the type of sensory stimuli induces the perceptual error, which exhibits complex visual features.

2. SIMPLE PCRC

In this section, we construct a simple one-layer PCRC network exactly driven by the prediction errors. We also analyze the nonlinear dynamics underlying the trained network to elucidate how it works.

2.1. Network Architecture and Dynamics

We use a leaky-integrator RNN, defined by the equation:

$$\tau\dot{\boldsymbol{x}} = -\boldsymbol{x} + W^{\mathrm{REC}}\boldsymbol{r} + W^{\mathrm{FB}}\boldsymbol{z} + W^{\mathrm{IN}}(\boldsymbol{d}-\boldsymbol{z}), \qquad (1)$$

where τ is the membrane time constant, N is the number of neurons, $\boldsymbol{x}(t) := (x_1(t), ..., x_N(t))^T \in \mathbb{R}^N$ represents the membrane potentials or activities of the neurons at time $t \in \mathbb{R}$, and $\boldsymbol{r} := (\phi(\boldsymbol{x}_1), ..., \phi(\boldsymbol{x}_N))^T \in \mathbb{R}^N$ represents the firing rates of the neurons with $\phi(\boldsymbol{x}) :=$ tan $h(\boldsymbol{x})$. $W^{\mathrm{REC}} \in \mathbb{R}^{N\times N}$ is a random recurrent connectivity matrix, whose elements are sampled i.i.d. from $\mathcal{N}(0, g^2/N)$ with the parameter $g \in \mathbb{R}$. M is the dimension of the input and output. The output of the network $\boldsymbol{z} := W^{\mathrm{OUT}}\,\boldsymbol{r} \in \mathbb{R}^M$ represents the prediction and is fed back through weights $W^{\mathrm{FB}} \in \mathbb{R}^{N\times M}$, whose elements are independently and uniformly sampled from $[-1, 1]$. The residual error between the target (or sensory input) $\boldsymbol{d}(t) \in \mathbb{R}^M$ and the prediction $\boldsymbol{z}(t)$ is fed through weights $W^{\mathrm{IN}} \in \mathbb{R}^{N\times M}$, whose elements are independently and uniformly sampled from $[-1, 1]$. The output weights $W^{\mathrm{OUT}} \in \mathbb{R}^{M\times N}$ are initially set to all zero, and modified during training. Note that the last term is unique to the PCRC network. This network architecture is illustrated in Figure 1, though contexts are ignored in this section.

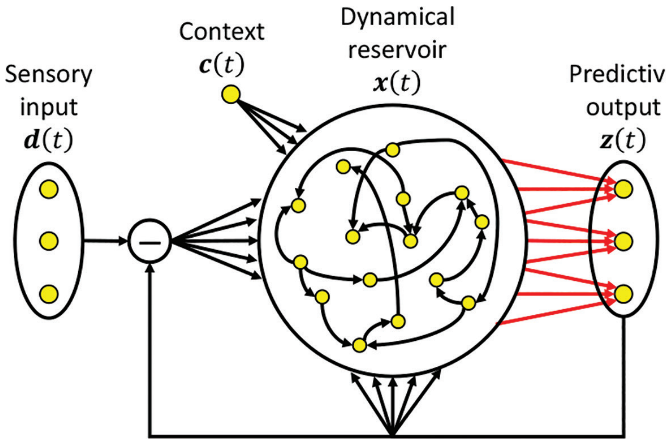


Figure 1 | Schematic chart of the one-layer PCRC network architecture. Only the output weights (red) are modified during training.

In order to simulate this dynamics numerically, we introduce the discrete-time version of Equation (1), derived by Euler method:

$$\boldsymbol{x}(n+1) = \boldsymbol{x}(n) + \frac{\delta}{\tau}\left[-\boldsymbol{x}(n) + W^{\mathrm{REC}}\boldsymbol{r}(n) + W^{\mathrm{FB}}\boldsymbol{z}(n) + W^{\mathrm{IN}}(\boldsymbol{d}(n)-\boldsymbol{z}(n))\right], \qquad (2)$$

where $n \in \mathbb{Z}$ is the discrete time step, δ is the small time interval, and other notations follow Equation (1).

Throughout this paper, we use $N = 1000$, $g = 1.2$, $\tau = 100$ ms, and $\delta = 10$. In this section, we use $M = 2$ for visibility of the dynamics.

2.2. Task

We present the constant vectors $\boldsymbol{d}^1, ..., \boldsymbol{d}^{N_D} \in \mathbb{R}^M$ in turn as the sensory inputs to the network, where N_D is the number of trials. The network is trained to keep outputting the target $\boldsymbol{d}^i$ until the next target $\boldsymbol{d}^{i+1}$ is presented, at each i^{th} trial. Each sensory input $\boldsymbol{d}^i$ is presented for 0.2 s, and its elements are independently and uniformly sampled from $[1, 2]$.

Since the network actually receive the prediction error $\boldsymbol{d}^i - \boldsymbol{z}(t)$ as the input, it is required to decode this error into the original sensory input $\boldsymbol{d}^i$. This corresponds to the framework of the predictive coding.

2.3. Learning Rule

We train W^{OUT} by Fast Order Reduced and Controlled Error (FORCE) learning algorithm [7], which is based on the recursive least square filter. Its update rule is:

$$\boldsymbol{e}(t) := W^{\mathrm{OUT}}(t-\Delta t)\,\boldsymbol{r}(t) - \boldsymbol{d}(t), \qquad (3)$$

$$\boldsymbol{s}(t) := P(t-\Delta t)\,\boldsymbol{r}(t)\,(1+\boldsymbol{r}^T(t)\,P(t-\Delta t)\,\boldsymbol{r}(t))^{-1}, \qquad (4)$$

$$P(t) = P(t-\Delta t) - \boldsymbol{s}(t)\,\boldsymbol{r}^{\mathrm{T}}(t)\,P(t-\Delta t), \qquad (5)$$

$$W^{\mathrm{OUT}}(t) = W^{\mathrm{OUT}}(t-\Delta t) - \boldsymbol{e}(t)\boldsymbol{s}^{\mathrm{T}}(t), \qquad (6)$$

where the initial value for $P(t) \in \mathbb{R}^{N\times N}$ is given by

$$P(0) = \frac{1}{\alpha}I, \ (\alpha \in \mathbb{R}). \qquad (7)$$

In this algorithm, the inverse of $P(t)$ is a running estimate of the autocorrelation matrix of the firing rates $\boldsymbol{r}(t)$ plus a regularization term:

$$P^{-1}(t) = \int \boldsymbol{r}(t')\,\boldsymbol{r}^T(t')\,dt' + \alpha I. \qquad (8)$$

Throughout this paper, we use $\alpha = 0.02$ and $\Delta t = \delta$.

2.4. Results and Analysis

We trained the network for 1000 trials. (i.e., $N_D = 1000$). At each trial in the test phase, the sensory input $\boldsymbol{d}^i$ is presented for 5.0 s. As shown in Figure 2, the training resulted in almost perfect performance. Figure 2 also shows that at the beginning of each i^{th} trial, the prediction error $\boldsymbol{d}^i - \boldsymbol{z}(t)$ is fed to the network as a sharp pulse, but it immediately decays to zero and the network settles into a fixed point $\overline{\boldsymbol{x}^i}$ where $\boldsymbol{z}(t) \equiv \boldsymbol{d}^i$.

In order to reveal the underling mechanism of this behavior, we analyze the nonlinear dynamics of the trained network. In what follows, we regard the term $W^{\mathrm{IN}}(\boldsymbol{d} - \boldsymbol{z})$ as the external force and separate it from the network's own dynamics, because of its pulse-like behavior, i.e., we here analyze the dynamics:

$$\tau\dot{\boldsymbol{x}} = -\boldsymbol{x} + W^{\mathrm{REC}}\boldsymbol{r} + W^{\mathrm{FB}}\boldsymbol{z}. \tag{9}$$

Following the approach of Sussillo and Barak [8], we define the scalar function $q(\boldsymbol{x}) := |\dot{\boldsymbol{x}}|^2/2$, which is near to zero if $\boldsymbol{x}$ is an approximate fixed point, or a slow point. Figure 3a shows that almost all the q values at the end of trials are very low, and the corresponding slow points are located on a 2D-manifold in the phase space. Figure 3a also shows that at the beginning of each i^{th} trial, the pulse-like prediction error $\boldsymbol{d}^i - \boldsymbol{z}(t)$ drives the trajectory out of the 2D-manifold, but in the subsequent relaxation phase, the trajectory is attracted by the 2D-manifold, and the projection of the trajectory onto the manifold corresponds to the total movement $\boldsymbol{d}^i - \boldsymbol{d}^{i-1}$.

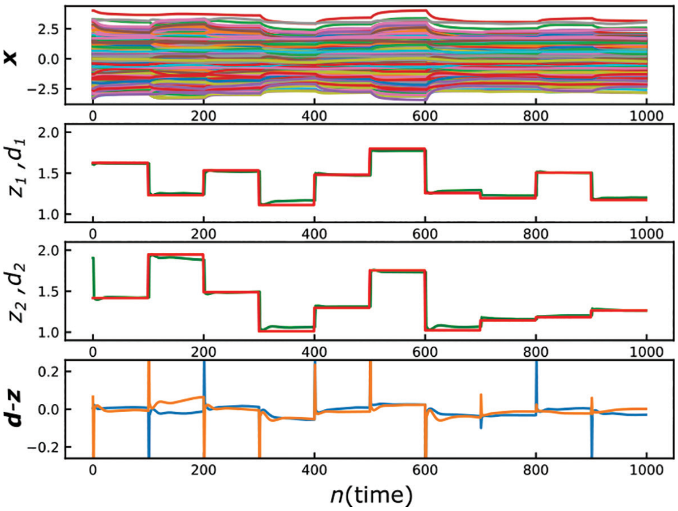


Figure 2 | The response of the trained network in the test phase. The 1st row represents the activities of the reservoir $\boldsymbol{x}(t)$, red plots in the 2nd and 3rd row represent the target $\boldsymbol{d}(t)$, green plots in the 2nd and 3rd row represent the output $\boldsymbol{z}(t)$, and the 4th row represents the prediction error $\boldsymbol{d}(t) - \boldsymbol{z}(t)$.

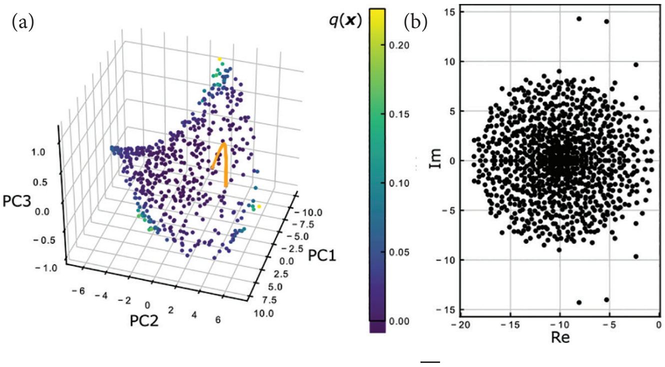


Figure 3 | (a) The locations of the slow points $\overline{\boldsymbol{x}^i}$ for the entire target range $\boldsymbol{d}^i \in [1, 2]^2$ in 3D principal component analysis space. The color scale of each slow point represents the value of $q(\overline{\boldsymbol{x}^i})$. The orange points represent an example of the trajectory of $\boldsymbol{x}(t)$ during a trial. (b) A typical example of the eigenvalue spectrum of the Jacobian at slow points on the 2D-manifold.

Furthermore, by analyzing the linearized system around each slow point on the 2D-manifold, we uncover the stability of this manifold. Linearizing Equation (9) around the slow point $\bar{\boldsymbol{x}}$ $(q(\bar{\boldsymbol{x}}) \simeq 0)$, we obtain the dynamics about the perturbation $\delta\boldsymbol{x} := \boldsymbol{x} - \bar{\boldsymbol{x}}$:

$$\delta\dot{\boldsymbol{x}} = \frac{1}{\tau}\left[-I + \left(W^{\mathrm{REC}} + W^{\mathrm{FB}}W^{\mathrm{OUT}}\right)R'(\bar{\boldsymbol{x}})\right]\delta\boldsymbol{x}, \tag{10}$$

where $R'_{ij}(\bar{\boldsymbol{x}}) := \delta_{ij}\phi'(\overline{x_i})$.

As for almost all the slow points, the linearized systems around them have only eigenvalues with the negative real part, as shown in Figure 3b. This suggests that almost all the slow points are locally stable, and the 2D-manifold composed of them attracts any trajectories in the vicinity of it. Nevertheless, this manifold attractor is not fully continuous and there is a slow flow on it. Then the trajectory on the manifold is attracted by the specific slow point on the manifold where the output $\boldsymbol{z}(t)$ is near to but not equal to the target $\boldsymbol{d}^i$, which leads to the little prediction error shown in Figure 2.

Up to this point, the trained network has exhibited the performance only on the discontinuously changing sensory inputs. Here we show that the same network can also perceive the continuously changing sensory inputs. For example, Figure 4a shows that the network output succeeded in following the sinusoidal input. In this case the network trajectory $\boldsymbol{x}(t)$ keeps travelling around the 2D-manifold in the phase space, as shown in Figure 4b. This behavior results from the balance between the attracting force from the 2D-manifold and the driving force by the prediction error $\boldsymbol{d}(t) - \boldsymbol{z}(t)$. Even in the general case, the same mechanism enables the network to perceive the continuously changing input.

Throughout this section, we have shown the case of $M = 2$ for simplicity, but the same scenario holds for the case of general M.

3. CONTEXT-DEPENDENT PCRC

In this section, we construct a simple PCRC network for the context-dependent perception task. We also analyze the trained network to elucidate how it switches the processing depending on contexts.

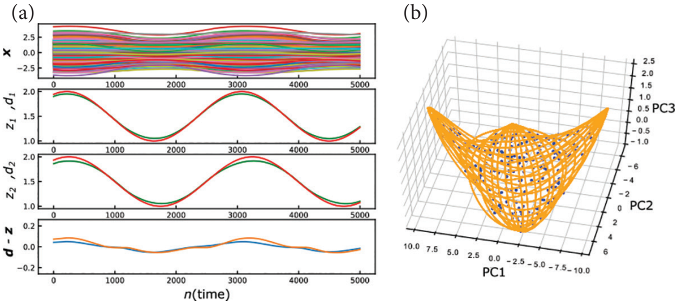


Figure 4 | The response of the trained network to a continuously changing sensory input. (a) The activities of the reservoir $\boldsymbol{x}(t)$ (the 1st row), the target $\boldsymbol{d}(t)$ (red plots in the 2nd and 3rd row), the output $\boldsymbol{z}(t)$ (green plots in the 2nd and 3rd row), and the prediction error $\boldsymbol{d}(t) - \boldsymbol{z}(t)$ (the 4th row). (b) The trajectory of $\boldsymbol{x}(t)$ (orange) and the 2D-manifold composed of the stable slow points (blue) in 3D PCA space.

3.1. Network Architecture and Dynamics

We add to Equation (1) the term of the context signal from external modules:

$$\tau\dot{\boldsymbol{x}} = -\boldsymbol{x} + W^{\mathrm{REC}}\boldsymbol{r} + W^{\mathrm{FB}}\boldsymbol{z} + W^{\mathrm{IN}}(\boldsymbol{d} - \boldsymbol{z}) + W^{\mathrm{CON}}\boldsymbol{c} \tag{11}$$

where the context $\boldsymbol{c}(t) \in \mathbb{R}^{L}$ is fed through weights $W^{\mathrm{CON}} \in \mathbb{R}^{N\times L}$, whose elements are independently and uniformly sampled from $[-1,1]$. In this section we use $M = 4$ and $L = 2$ for simplicity, and the other settings follow Section 2.

3.2. Task and Learning Rule

We present the constant vectors $\boldsymbol{d}^{1},\ldots,\boldsymbol{d}^{N_D} \in \mathbb{R}^{M}$ in turn as the sensory inputs to the network, and train the network to keep outputting each given constant vector until the next target is presented. We also present the context $\boldsymbol{c}^{1} := (0,1)^{T}$ during the 1st to $N_D/2$th trials, and the context $\boldsymbol{c}^{2} := (1, 0)^{T}$ during the $N_D/2+1$th to N_Dth trials, respectively. Each sensory input d^{i} is presented for 0.2 s, and its elements are given as $\boldsymbol{d}^{i} = (d_1^i, 1/d_1^i, d_2^i, 1/d_2^i)^{T}$ on the context $\boldsymbol{c}^{1}$ and $\boldsymbol{d}^{i} = (d_1^i, d_2^i, d_2^i/2, d_1^i/2)^{T}$ on the context $\boldsymbol{c}^{2}$, respectively. At each trial d_1^i and d_2^i are independently and uniformly sampled from $[1, 2]$. Note that the essential dimension of the input is $M/2$ on each context, but the trained network is required to switch the type of processing depending on contexts.

We train W^{OUT} by FORCE learning algorithm used in Section 2.

3.3. Results and Analysis

We trained the network for 1000 trials on each context $\boldsymbol{c}^{1}$ and $\boldsymbol{c}^{2}$. (i.e., $N_D = 2000$). At each trial in the test phase, the sensory input $\boldsymbol{d}^{i}$ is presented for 1.0 s. As shown in Figure 5a, the training resulted in almost perfect performance, and the pulse-like prediction error drives the network from one slow point to another, as with the context-free case in Section 2. Figure 5b shows that the two different 2D-manifold attractors are formed for the contexts $\boldsymbol{c}^{1}$ and $\boldsymbol{c}^{2}$, respectively. This suggests that the network switches its attractor-landscape depending on contexts, and the same mechanism as Section 2 enables the network to perceive the sensory inputs on each context.

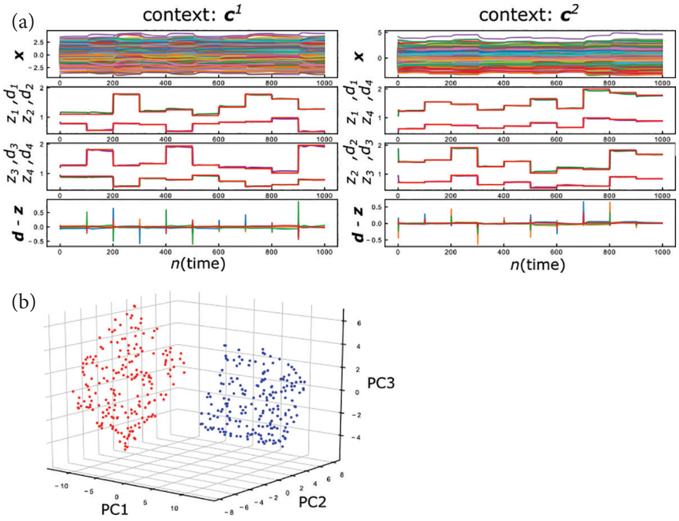


Figure 5 The context-dependent response of the trained network. (a) The activities of the reservoir $\boldsymbol{x}(t)$ (the 1st row), the target $\boldsymbol{d}(t)$ (red plots in the 2nd and 3rd row), the output $\boldsymbol{z}(t)$ (green and purple plots in the 2nd and 3rd row), and the prediction error $\boldsymbol{d}(t) - \boldsymbol{z}(t)$ (the 4th row). (Left: context c^1. Right: context c^2). (b) The locations of the slow points $\overline{\boldsymbol{x}^{i}}$ for the entire target range $(d_1^i, d_2^i) \in [1, 2]^2$ in 3D PCA space. (Blue: on the context c^1. Red: on the context c^2).

We next evaluate the performance of the trained network when the type of the sensory input ($\boldsymbol{d}^{i} = (d_1^i, 1/d_1^i, d_2^i, 1/d_2^i)^{T}$ or $(d_1^i, d_2^i, d_2^i/2, d_1^i/2)^{T}$) does not match the context. In this case, we present each sensory input $\boldsymbol{d}^{i}$ for 5.0 s. Figure 6 shows that the context mismatch keeps the prediction errors apart from zero, so that the network fails to perceive the sensory inputs, but nevertheless the network trajectories sometimes settle into slow points.

4. CONTEXT-DEPENDENT PCRC FOR VISUAL DATA

In this section, we construct a context-dependent PCRC network which can perceive more high-dimensional visual inputs, in order to demonstrate that the proposed network can be a possible mechanism of the visual perception. We also observe the complex features of the perceptual error induced by the context mismatch.

4.1. Network Architecture, Task, and Learning Rules

The network architecture and settings follow those of Section 3, except the dimension of the input and output: $M = 20$.

We use the Mixed National Institute of Standards and Technology (MNIST) data set, which is widely used for handwritten numeral recognition tasks, as the high-dimensional visual sensory stimuli. As the preprocessing, we first compress the MNIST data whose labels are "0" or "1" into 20 dimension, using the Non-negative Matrix Factorization (NMF). We next randomly choose one of the compressed MNIST data as the sensory input $\boldsymbol{d}^{i}$ and present it to the network for 0.2 s at each ith trial. At the same time, we present the context $\boldsymbol{c}^{1}$ if $\boldsymbol{d}^{i}$ has "0" label, and the context $\boldsymbol{c}^{2}$ if $\boldsymbol{d}^{i}$ has "1" label, respectively. (i.e., each context represents the category of the visual sensory input). We train the network to keep outputting the presented sensory input $\boldsymbol{d}^{i}$ during each trial. In the test phase, we present to the network unlearned compressed MNIST data as the

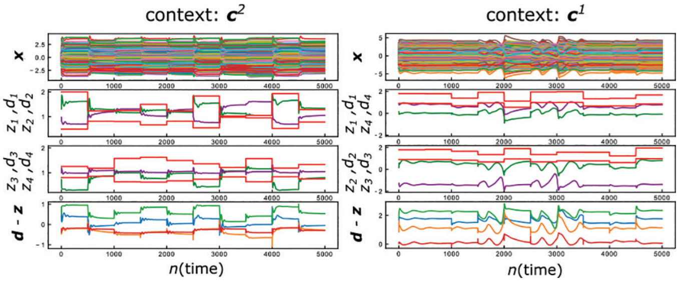


Figure 6 The perceptual errors induced by the context mismatch. (Left: $\boldsymbol{d}^{i} = (d_1^i, 1/d_1^i, d_2^i, 1/d_2^i)^{T}$ on the context $\boldsymbol{c}^{2}$. Right: $\boldsymbol{d}^{i} = (d_1^i, d_2^i, d_2^i/2, d_1^i/2)^{T}$ on the context $\boldsymbol{c}^{1}$).

sensory inputs. The trained network is required to form slow points that correspond to even unlearned MNIST data in its phase space.

We use the FORCE algorithm again during training.

4.2. Results and Analysis

We trained the network for 2000 trials on each context $\boldsymbol{c}^1$ and $\boldsymbol{c}^2$. (i.e., $N_D = 4000$). At each trial in the test phase, we present a randomly chosen unlearned MNIST data for 5.0 s as the sensory input $\boldsymbol{d}^i$. Figure 7a shows that the training network almost succeeded in perceiving unlearned MNIST inputs, and the pulse-like prediction error drives the network from one slow point to another, as with the case above. Figure 7b shows that the two different manifold attractors are formed for the "0" label and "1" label MNIST inputs respectively, but in the 3D PCA space we cannot observe the actual shapes of these manifolds.

We next evaluate the performance of the trained network when the label of the sensory input does not match the context. As shown in Figure 8, the context mismatch keeps the prediction errors apart from zero, but nevertheless the network trajectories settle into slow points.

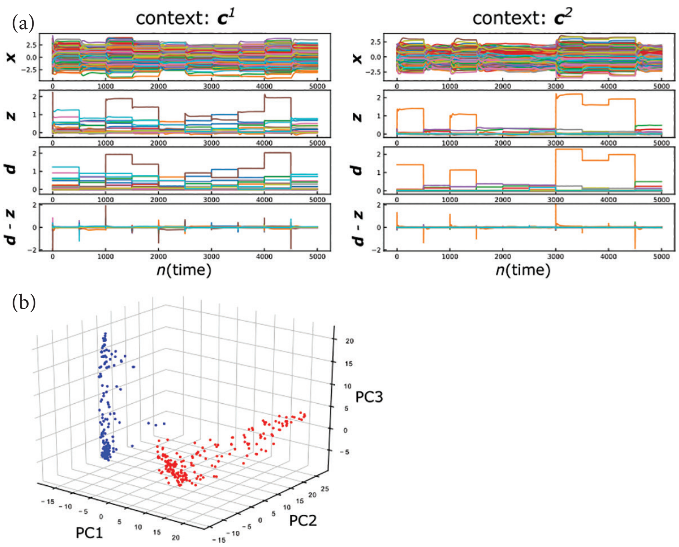


Figure 7 | The context-dependent response of the trained network to unlearned MNIST inputs. (a) The activity of the reservoir $\boldsymbol{x}(t)$, the output $\boldsymbol{z}(t)$, the target $\boldsymbol{d}(t)$ and the prediction error $\boldsymbol{d}(t) - \boldsymbol{z}(t)$. (Left: context $\boldsymbol{c}^1$. Right: context $\boldsymbol{c}^2$). (b) The locations of the slow points $\overline{\boldsymbol{x}}^i$ in 3D PCA space on the context $\boldsymbol{c}^1$ (red) and on the context $\boldsymbol{c}^2$ (blue), respectively.

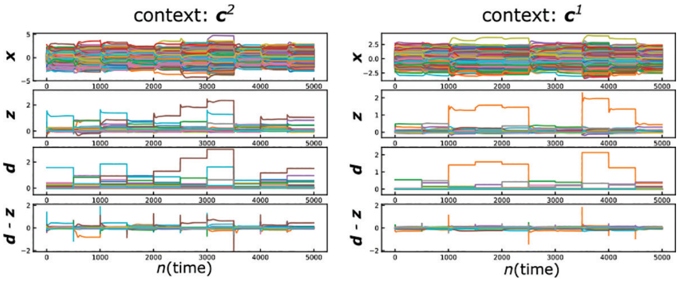


Figure 8 | The perceptual errors induced by the context mismatch. (Left: label "0" data are presented on the context $\boldsymbol{c}^2$. Right: label "1" data are presented on the context $\boldsymbol{c}^1$).

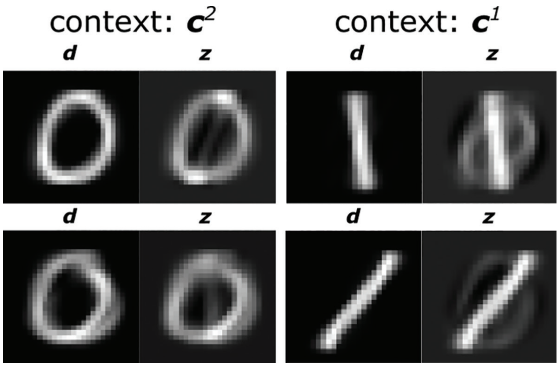


Figure 9 | The examples of the visualized comparison between the original inputs $\boldsymbol{d}$ and errored predictions $\boldsymbol{z}$ on the context mismatch. (Left: label "0" data are presented on the context $\boldsymbol{c}^2$. Right: label "1" data are presented on the context $\boldsymbol{c}^1$).

We further visualize these errored predictions $\boldsymbol{z}$ at slow points and compare them with the original inputs $\boldsymbol{d}$, by inversely transforming the output $\boldsymbol{z}$ into the dimension of the original MNIST data, using the matrix generated in NMF. As a result, in the errored predictions, the original sensory inputs and the predictions for the wrong label MNIST image overlap each other, as illustrated in Figure 9.

5. DISCUSSION

We first proposed the simple one-layer PCRC network driven by the prediction error, which can perceive even unlearned inputs. We analyzed the nonlinear dynamics underling the trained network, and revealed that the network perceives the sensory stimuli using the low-dimensional manifold attractor in its phase space. Since low-dimensional manifold attractors have also been observed in the trained RNNs in previous studies [6,8,9], it can be a natural strategy for RNNs to use them for the information processing.

Next, we constructed the simple PCRC network for the context-dependent task, and observed that the different attractor-landscape is formed on each context. Throughout this study, we used the PCRC networks with only one layer and assumed the context signals to be fed from the external module, for simplicity. However, the hierarchy plays a key role in the predictive coding framework [1], and how the context signals are generated remains to be clarified. Therefore, it is our future work to build the hierarchical PCRC model composed of the one-layer networks which are analyzed in this study, and incorporate the modules which generate the context signals inside the model.

Finally, we constructed the context-dependent PCRC network for the compressed MNIST data task, and demonstrated that the proposed network can be a possible mechanism of the visual perception. The perceptual errors induced by the context mismatch exhibited complex features, and interestingly, they share some common features with the symptoms of the hallucination in dementia with Lewy bodies [10], in which the patients see other people who are not there on the background which actually exists there. It is also our future work to study the relation between these perceptual errors.

CONFLICTS OF INTEREST

There is no conflicts of interest.

ACKNOWLEDGMENTS

The authors are grateful to T. Kohno for useful discussions. This paper is partially supported by AMED under Grant Number JP18dm0307009, NEC Corporation, and JSPS KAKENHI Grant Number JP16K00246, and also based on results obtained from a project subsidized by the New Energy and Industrial Technology Development Organization (NEDO).

REFERENCES

[1] R.P.N. Rao, D.H. Ballard, Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects, Nat. Neurosci. 2 (1999), 79–87.

[2] M. Fukino, Y. Katori, K. Aihara, A computational model for pitch pattern perception with the echo state network, 2016 International Symposium on Nonlinear Theory and its Applications, Yugawara, Japan, 2016, pp. 271–274.

[3] K. Ara, Y. Katori, Hierarchical network model of auditory information processing using dynamical predictive coding and non-negative matrix factorization, 23rd International Conference on Artificial Life and Robotics, Beppu, Japan, 2018, pp. 41–46.

[4] Y. Katori, Network model for dynamics of perception with reservoir computing and predictive coding, Advances in Cognitive Neurodynamics (VI), Chapter 11, Springer Nature Singapore Pvt., Ltd., Singapore, 2018, pp. 89–95.

[5] M. Lukoševičius, H. Jaeger, Reservoir computing approaches to recurrent neural network training, Comput. Sci. Rev. 3 (2009), 127–149.

[6] V. Mante, D. Sussillo, K.V. Shenoy, W.T. Newsome, Context-dependent computation by recurrent dynamics in prefrontal cortex, Nature 503 (2013), 78–84.

[7] D. Sussillo, L.F. Abbott, Generating coherent patterns of activity from chaotic neural networks, Neuron 63 (2009), 544–557.

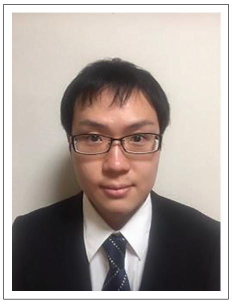
[8] D. Sussillo, O. Barak, Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks, Neural Comput. 25 (2013), 626–649.

[9] C.L. Beer, O. Barak, Dynamics of dynamics: following the formation of a line attractor, arXiv:1805.09603, 2018.

[10] D. Collerton, J.P. Taylor, I. Tsuda, H. Fujii, S. Nara, K. Aihara, et al., How can we see things that are not there? Current insights into complex visual hallucinations, J. Consciousness Stud. 23 (2016), 195–227.

Authors Introduction

Mr. Hiroto Tamura



He received the B.E. degree of applied mathematics in 2017 from Waseda University, Japan, and the M.E. degree of electronic engineering in 2019 from the University of Tokyo, Japan. Currently, he is PhD Candidate of Graduate School of Engineering, the University of Tokyo. His research interests include computational neuroscience and nonlinear dynamics.

Dr. Yuichi Katori

He received the PhD degree of science in 2007 from the University of Tokyo, Japan. Currently, he is Associate Professor of Future University Hakodate. His research interests include mathematical modeling of complex systems, computational neuroscience, and brain-like artificial intelligence systems.

Dr. Kazuyuki Aihara

He received the B.E. degree of electrical engineering in 1977 and the PhD degree of electronic engineering 1982 from the University of Tokyo, Japan. Currently, he is Professor of Institute of Industrial Science, Graduate School of Information Science and Technology, and Graduate School of Engineering, the University of Tokyo. His research interests include mathematical modeling of complex systems, parallel distributed processing with spatio-temporal neurodynamics, and time series analysis of complex data.