Emergence of Hierarchical Structure mirroring Linguistic Composition in a Recurrent Neural Network
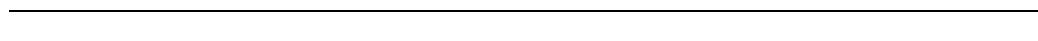
Wataru Hinoshita[a,1,*], Hiroaki Arie[b], Jun Tani[b], Hiroshi G. Okuno[a], Tetsuya Ogata[a]

[a]*Graduate School of Informatics, Kyoto Univ.*
Engineering Building #10, Sakyo, Kyoto, 606-8501, Japan
[b]*Brain Science Institute, RIKEN*
2-1 Hirosawa, Wako-shi, Saitama, 351-0198, Japan

[*]Corresponding author
 Email address: `hinosita@kuis.kyoto-u.ac.jp` (Wataru Hinoshita)
 [1]Tell: +81-75-753-5992, Fax: +81-75-753-5977

Emergence of Hierarchical Structure mirroring Linguistic Composition in a Recurrent Neural Network

Abstract

We show that a Multiple Timescale Recurrent Neural Network (MTRNN) can acquire the capabilities to recognize, generate, and correct sentences by self-organizing in a way that it mirrors the hierarchical structure of sentences: characters grouped into words, and words into sentences. In an experiment, we trained our model over a set of unannotated sentences from an artificial language, represented as sequences of characters. Once trained, the model could recognize and deterministically generate grammatical sentences, even if they were not learned. Moreover, we found that our model could correct a few substitution errors in a sentence, and the correction performance was improved by adding the errors to the training sentences in each training iteration with a certain probability. An analysis of the neural activations in our model revealed that the MTRNN had self-organized, reflecting the hierarchical linguistic structure by taking advantage of the differences in time scale among its neurons: in particular, neurons that change the fastest represented "characters," those that change more slowly, "words," and those that change the slowest, "sentences."

Keywords: Language acquisition, Multiple Timescale Recurrent Neural Network, Hierarchical linguistic structure, Self-organization

1. Introduction

Many theories have been proposed to explain how children can acquire a language system that generates diverse complex sentences, despite limited and often corrupted linguistic stimuli. As a result of the progress made in analyzing non-linear dynamical systems and chaos [6], it has been revealed that diverse, complex patterns can emerge from the continuous interaction between primitive components [20]. Therefore, connectionist approaches based on the viewpoint of language as a complex dynamical system have started to attract a great deal of attention [7, 1, 21].

Recurrent Neural Networks (RNN) [5, 12], which can be seen as non-linear dynamical systems [13], are often used as the basis of such connectionist models for language acquisition because of their representational power. In fact, some recurrent architectures have been shown to be computationally Turing equivalent [14, 11]. Another important feature of RNNs is "self-organization", which is a phenomena that a global coherent structure appears in a system not by a central authority but by local interaction among elements of the system. When we train an RNN to learn language, whole neurons of the RNN organize a structure that emulates some aspects of language function, through a process where each neuron changes its connection weights according to learning rules while interacting other neurons. We need not to design the linguistic structure because it spontaneously emerges in an RNN. Thus, examining what structure emerges helps us gain insight into how the human brain acquires language function.

1.1. Existing RNN Models for Language Acquisition and their Problems

Many researches have shown that RNNs can learn to classify sentences as either grammatical or ungrammatical [17, 14]. But these models can not generate any sentences and require sentences annotated as grammatical or not for a training set. Humans' ability to generate sentences is an essential part of language function. In addition, it is claimed that in reality there is virtually no evidence of what is ungrammatical in the linguistic stimuli exposed to children [2, 3]. In fact, children hardly encounter annotated sentences in daily life. Thus, it is significant to consider a model which can acquire a capability to generate sentences from only a set of unannotated sentences.

Elman [8, 9, 7] proved that a Simple Recurrent Network (SRN) could learn grammar using only a set of unannotated sentences by training it to predict the next word from those that had been input up to that step. The model provides step-by-step syntactic assessments of sentences. We can also sample various sentences from the trained SRN, as can formal grammars, by selecting words according to the probability output from it. The model, however, does not seem to be the best for us to gain insights into human's generative capability of language because its stochastic process for generating sentences is completely different from humans. Humans generate a sentence depending on their intention, but the SRN does not generate a certain sentence selectively and deterministically.

Sugita et al. [22] and Ogata et al. [16] modeled the process where a sentence was generated depending on an intention by using an RNN model with Parametric Bias (RNNPB) [23]. The model learns language in a similar

3

way as the SRN, but it deterministically generates sentences by selecting a word corresponding to the most highly-activated nodes at each step. Values of its parameter nodes determine which sentence the model generates and the parameter space is spontaneously organized through the training process of the model. We can see that the parameter vector corresponds to an intention which determines what to say. Their models enabled a robot to generate a sentence expressing its motion by mapping a robot's sensori-motor flow to the parameter space [16]. They, however, dealt only with very simple sentences composed of two or three words, because the model had difficulty learning long, complex sequences.

A common problem of existing RNN models is that they require a predetermined word set to learn language, because each of their input nodes corresponds to a word [10, 24, 14, 16]. If there is no previous knowledge about words, a model should acquire the representation of words themselves by composing more primitive elements, such as characters or phonemes. Thus, learning languages without a predetermined word set requires the capability of hierarchical composition, such as characters to words and words to sentences. Humans have this capability in reality and it plays an essential role in providing infinite expressions to natural language with a limited number of elements (characters or phonemes). Therefore, it is also important to reveal whether a neural system can self-organize reflecting such hierarchical structures.

In most studies on the language acquisition by RNNs, there seems to be an implicit assumption that all sentences given to the model are correct and clean, though linguistic stimuli exposed to children are not always correct

and clean. Thus, it is important to examine whether the RNN models can acquire the language function even from corrupted sentences. Moreover, the question of how corruptions in linguistic stimuli impact linguistic capabilities is also worth investigating.

1.2. Features of our Model

We applied a Multiple Timescale Recurrent Neural Network (MTRNN) [25] to a language acquisition task. The model was proposed by Yamashita and Tani and they originally used it for the task of learning robots' sensori-motor flow [25]. We found that the model enabled us to overcome the above mentioned problems on language acquisition in a connectionist way. We show the following points in this paper.

1. Our model self-organizes mirroring hierarchical linguistic composition (characters to words and words to sentences) when it is trained using a set of unannotated sentences, each of which is represented as a sequence of characters. Thus, our model can acquire the capabilities to recognize, generate and correct sentences, even if they are not learned, without any previous knowledge about lexicon nor grammar.

2. Our model can deterministically generate a certain sentence depending on its initial state. This state space is spontaneously organized through its training process in a way that holistic representation of sentences are embedded within it. Thus, we can model the process where humans generate a sentence depending on an intention. This capability can lead to a system which generates sentences depending on a situation by mapping the situation to the initial state space. Moreover, it can learn more complex sentences than RNNPBs [23].

5

3. The robustness of the linguistic structure that emerges in our model improves when we add some substitution errors to the training sentences with a certain probability in each training iteration. This can provide insight into the question of how corruptions in linguistic stimuli impact linguistic capabilities.

2. Model

Our language learning model is based on an MTRNN [25]. An MTRNN deals with sequences by calculating the next state $S(t + 1)$ from the current state $S(t)$ and the contextual information stored in their neurons. The model is composed of several neuron groups, each with an associated time constant. If the neurons have a larger time constant, their states change more slowly. The time scale difference causes information to be hierarchically coded. MTRNN can deterministically generate sequences depending on the initial states of certain context nodes. Moreover, given a sequence, the model can calculate the initial states from which it generates the target sequence. Therefore, this model can be used as the recognizer and generator of sequences. The initial state space self-organizes based on the dynamical structure among the training sequences. Thus, the model even deals with unknown sequences by generalizing the training sequences.

Figure 1 shows an overview of our model. Our language learning model has three neuron groups, consisting of the input-output (IO), Fast Context (Cf), and Slow Context (Cs) groups, in increasing order of time constant τ. The IO has 30 nodes and each of them corresponds to one of the characters from the 26 letters in the alphabet ('a' to 'z') and four other symbols (space,

6

period, comma, and question mark). Cf and Cs have 40 and 11 nodes, respectively. We choose six neurons from Cs to be used as the Controlling Slow Context (Csc), whose initial states determine the sequence. We determined the number of nodes for each neuron group on a trial basis. We set $\tau = 2, 5, 70$, for IO, Cf, Cs respectively, in reference to the original paper of the MTRNN [25] and we confirmed this worked well.

In our model, a sentence is represented as a sequence of IO activations corresponding to characters. The model learns to predict the next IO activation from the activations up to that point. Therefore, we use only a set of sentences to train our model. Figure 2 shows an example of the training sequence for this model. We used this representation method in order to reveal, as simply as possible, whether a neural dynamical system could acquire the capability of hierarchical composition, such as characters to words and words to sentences. Thus, we do not claim the neural plausibility for this representation method.

The activation value of the i-th neuron at step t ($y_{t,i}$) is calculated as follows.

$$
y_{t,i} =
\begin{cases}
\dfrac{\exp(u_{t,i} + b_i)}{\displaystyle\sum_{j \in I_{IO}} \exp(u_{t,j} + b_j)} & \cdots \ (i \in I_{IO}) \qquad \text{(1a)} \\[2em]
\dfrac{1}{1 + \exp(-(u_{t,i} + b_i))} & \cdots \ (i \notin I_{IO}) \qquad \text{(1b)}
\end{cases}
$$

$$
u_{t,i} = \begin{cases} 0 & \cdots (t = 0 \wedge i \notin I_{Csc}) \\ Csc_{0,i} & \cdots (t = 0 \wedge i \in I_{Csc}) \quad (2) \\ \left(1 - \dfrac{1}{\tau_i}\right) u_{t-1,i} + \dfrac{1}{\tau_i}\left[\displaystyle\sum_{j \in I_{all}} w_{ij} x_{t,j}\right] & \cdots (\text{otherwise}) \end{cases}
$$

$$
x_{t,j} = y_{t-1,j} \qquad\qquad\qquad\qquad\qquad\qquad \cdots (t \geq 1) \qquad (3)
$$

$$
\begin{aligned}
I_{IO}, I_{Cf}, I_{Cs}, I_{Csc} \;&:\; \text{neuron index set of each group } (I_{Csc} \subset I_{Cs}) \\
I_{all} \;&:\; I_{IO} \cup I_{Cf} \cup I_{Cs} \\
u_{t,i} \;&:\; \text{internal state of } i\text{-th neuron at step } t \\
b_i \;&:\; \text{bias of } i\text{-th neuron} \\
Csc_{0,i} \;&:\; \text{initial state controlling MTRNN} \\
\tau_i \;&:\; \text{time constant of } i\text{-th neuron} \\
w_{ij} \;&:\; \text{connection weight from } j\text{-th to } i\text{-th neuron} \\
& \quad\; w_{ij} = 0 \cdots (i \in I_{IO} \wedge j \in I_{Cs}) \vee (i \in I_{Cs} \wedge j \in I_{IO}) \\
x_{t,j} \;&:\; \text{input from } j\text{-th neuron at step } t
\end{aligned}
$$

The connection weights (w_{ij}), biases (b_i), and initial states ($Ccs_{0,i}$) are updated using the Back Propagation Through Time (BPTT) algorithm [19] as follows.

$$
w_{ij}^{(n+1)} = w_{ij}^{(n)} \;-\eta \frac{\partial E}{\partial w_{ij}} \;= w_{ij}^{(n)} \;-\frac{\eta}{\tau_i}\sum_t x_{t,j}\frac{\partial E}{\partial u_{t,i}} \qquad (4)
$$

$$
b_i^{(n+1)} = b_i^{(n)} \;-\beta \frac{\partial E}{\partial b_i} \;= b_i^{(n)} \;-\beta\sum_t \frac{\partial E}{\partial u_{t,i}} \qquad (5)
$$

$$
Csc_{0,i}^{(n+1)} = Csc_{0,i}^{(n)} - \alpha \frac{\partial E}{\partial Csc_{0,i}} = Csc_{0,i}^{(n)} - \alpha \frac{\partial E}{\partial u_{0,i}} \qquad \cdots (i \in I_{Csc}) \qquad (6)
$$

8

$$E = \sum_t \sum_{i \in I_{IO}} y_{t,i}^* \cdot \log \left(\frac{y_{t,i}^*}{y_{t,i}} \right) \qquad\qquad (7)$$

$$\frac{\partial E}{\partial u_{t,i}} = \begin{cases} y_{t,i} - y_{t,i}^* + \left(1 - \dfrac{1}{\tau_i}\right) \dfrac{\partial E}{\partial u_{t+1,i}} & \cdots \ (i \in I_{IO}) (8\text{a}) \\[3ex] y_{t,i}(1 - y_{t,i}) \displaystyle\sum_{k \in I_{all}} \dfrac{w_{ki}}{\tau_k} \dfrac{\partial E}{\partial u_{t+1,k}} + \left(1 - \dfrac{1}{\tau_i}\right) \dfrac{\partial E}{\partial u_{t+1,i}} & \cdots \ (\text{otherwise}) (8\text{b}) \end{cases}$$

n : iteration number in updating process

E : prediction error

$y_{t,i}^*$: ideal activation value of i-th neuron at step t for target sentence

η, β, α : learning rate constant

The derivations of these formulas are given in Appendix C. The form of error function E is called Kullback-Leibler divergence. When the output function for IO nodes is the softmax function (Eq. (1a)), this error function is used for the BPTT [25, 15]. When using the BPTT algorithm, the IO input values ($x_{t,j}$) are calculated along with the feedback from the training sequence using the following equation instead of Eq. (3).

$$x_{t,j} = (1 - r) \times y_{t-1,j} + r \times y_{t-1,j}^* \qquad \cdots (t \geq 1 \wedge j \in I_{IO}) \ (9)$$

r : feedback rate $(0 \leq r \leq 1)$

The initial Csc states determine the MTRNN's behavior. Thus, we define an initial Csc state vector ($\boldsymbol{Csc_0}$) as a six dimensional vector where each of the dimensions corresponds to the initial Csc state. $\boldsymbol{Csc_0}$ is independently prepared for each training sequence while the network weights (connection weights and biases) are shared by all the sequences. The initial state space

9

self-organizes based on the dynamical structure among the training sequences through a process where the network weights and $\boldsymbol{Csc_0}$ are simultaneously updated.

To recognize a sequence, the $\boldsymbol{Csc_0}$ representing the target sequence is calculated by the BPTT from Eq. (6), while the network weights are fixed to prevent the network from changing. The calculation of $\boldsymbol{Csc_0}$ by BPTT sometimes falls to a local minimum. Therefore, we calculate it 20 times while changing the initial value in the updating process $(Ccs_{0,i}^{(0)})$, and choose the result with the lowest error E (cf. Eq. (7)). It is difficult to calculate it in real time, but it does not take so long because the number of necessary iterative steps until convergence in a recognition phase is much less than that in the training phase. In this recognition phase, the IO input values are calculated using Eq. (9) if the value of the target sequence is given, otherwise they are calculated using Eq. (3). Thus, the MTRNN can recognize sequences even if only partial information is given.

A sequence is generated by recursively executing a forward calculation (Eqs. (1a), (1b), (2), and (3)) using $\boldsymbol{Csc_0}$ that represents the target sequence.

3. Representation of Sentences

In our model, a sentence is represented as a sequential activation pattern of IO neurons (Fig. 2). In this section, we explain in detail the process of translation between a sentence and an IO activation pattern.

We define C as a set of characters composed of the 26 letters in the alphabet ('a' to 'z') and four other symbols (space, period, comma, and
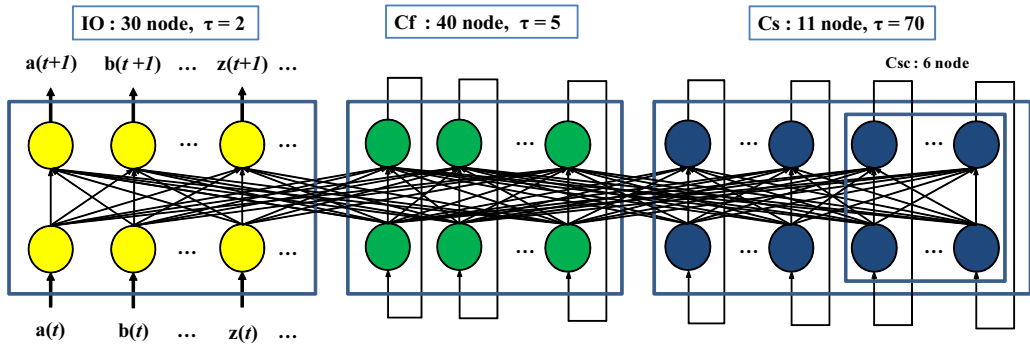
10

Figure 1: Overview of Language Learning MTRNN: $a(t)$ is activation value of neuron corresponding to 'a'. The others $(b(t), ..., z(t), ...)$ are defined in the same way. The sentences are represented by the successive activation of IO neurons.
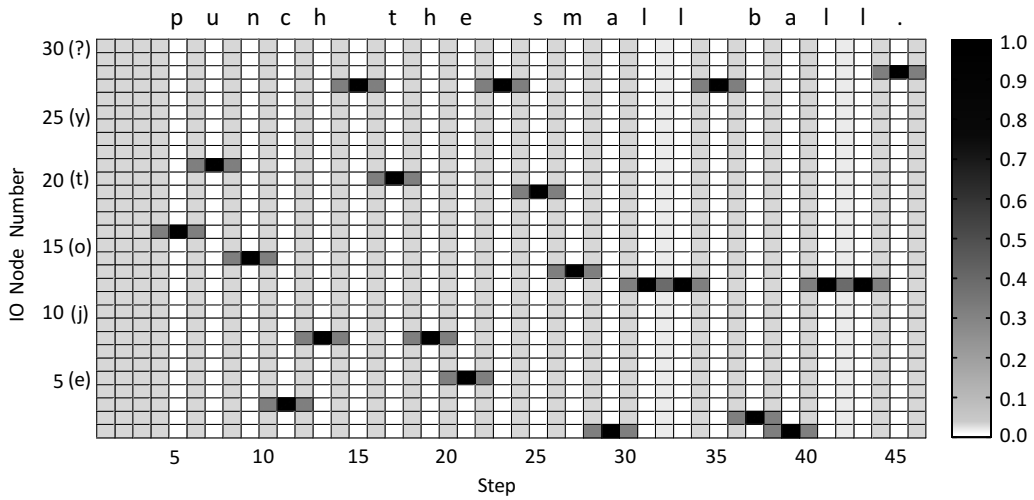

Figure 2: Example of training sequence: "punch the small ball."

question mark). We also define $c_i \in C$ as a character corresponding to the i-th neuron $(i \in I_{IO})$. We represent a sentence (S) whose length is L as a C^L vector : $S = \{s_1, s_2, ..., s_L\}$, $(s_k \in C)$.

3.1. Encoding Sentences into Neural Activation

An ideal activation value of the i-th neuron at step t for S, $y_{t,i}^*$, is calculated by applying a soft-max function to the ideal internal states of IO neurons. An ideal internal state of the i-th neuron at step t for S, $u_{t,i}^*$, is calculated by convoluting a gaussian $g(t)$ to a sequence of spikes $f_i(t)$ that are activated whenever c_i appears in S. The precise formulation is as follows.

$$y_{t,i}^* = \frac{\exp(u_{t,i}^*)}{\sum_{j \in I_{IO}} \exp(u_{t,j}^*)} \tag{10}$$

$$u_{t,i}^* = \lambda \cdot (g * f_i)(t) \tag{11}$$

$$f_i(t) = \begin{cases} 1 & \cdots (s_k = c_i \wedge t = \mu + (k-1)\nu) \\ 0 & \cdots (\text{otherwise}) \end{cases} \tag{12}$$

$$g(t) = \begin{cases} \exp\left(-\dfrac{t^2}{2\sigma^2}\right) & \cdots (-\dfrac{\omega}{2} \le t \le \dfrac{\omega}{2}) \\ 0 & \cdots (\text{otherwise}) \end{cases} \tag{13}$$

λ : scaling factor

μ : head margin

ν : interval between two characters

ω : filter width

σ : filter sharpness factor

12

The constant λ is used to match the activation scale between the IO and the other neurons whose activation functions are sigmoid (Eq. (1b)). We determined λ by using the following equation so that the maximum value of $y^*_{t,i}$ equals 0.9:

$$\lambda = \frac{1}{\max\limits_{t,i}((g * f_i)(t))} \cdot \ln\left(\frac{0.9}{1.0 - 0.9} \cdot (\text{size}(I_{IO}) - 1)\right). \tag{14}$$

In our experiment, we set the five constants as follows : $\lambda = 5.5645$, $\mu = 4$, $\nu = 2$, $\omega = 4$, and $\sigma^2 = 0.3$.

3.2. Decoding Neural Activation

We can obtain a sentence by decoding the IO activation pattern generated by an MTRNN. The k-th character (s_k) in the sentence is decided by using the following equations.

$$
\begin{aligned}
s_k &= c_m \\
m &= \operatorname*{argmax}_{i}(y_{(\mu+(k-1)\nu),i})
\end{aligned}
\tag{15}
$$

We can gain a whole sentence S by using these equations L times. Even if the sentence length L is not given, we can find the end of the sentence by checking whether s_k is a period or not.

4. Experiments

Our experiment aimed to reveal whether the MTRNN could acquire capabilities to recognize and generate sentences, even if they were not learned,

through self-organization of linguistic structure. Thus, we trained the MTRNN to learn language without any previous knowledge of words or grammar, and tested its capabilities to recognize and generate sentences.

The performance of the model was examined for three types of sentences: correct sentences, ungrammatical sentences, and sentences with a few misspellings. First, using both the correct and ungrammatical sentences, we examined whether our model could adequately acquire linguistic structure, such as the necessary grammar. In particular, we tested its capabilities to recognize and generate grammatical and ungrammatical sentences using a "sentence emulation task" where our model tries to recognize a sentence and generate the same sentence from the result of the recognition. Second, we examined the robustness of the linguistic structure that emerged in our model using a "sentence correction task" where our model tried to correct misspellings in the sentences.

We also aimed to investigate how corruption of the linguistic stimuli impacted the linguistic capabilities acquired from them. For this purpose, we added a few substitution errors to the training sentences with a fixed probability ρ in each training iteration. We trained our model 20 times for each $\rho = 0, 5, 10, 20, 30, 40 (\%)$, and each time changed the initial value of the network weights.

4.1. Target Language

In this experiment, we used a very small language set in order to make it possible to analyze the linguistic structure that emerged in the MTRNN. Our language set contained 17 words in 7 categories (Table 1) and a regular grammar consisting of 9 rules (Table 2). The number of different sentences

14

that can be generated from the language set is 441. (This language set was designed for robot tasks.)

Table 1: Lexicon

Category	Nonterminal symbol	Words
Verb (intransitive)	V_I	jump, run, walk
Verb (transitive)	V_T	kick, punch, touch
Noun	N	ball, box
Article	ART	a, the
Adverb	ADV	quickly, slowly
Adjective (size)	ADJ_S	big, small
Adjective (color)	ADJ_C	blue, red, yellow

Table 2: Regular grammar

S → V_I	NP → ART N	ADJ → ADJ_S
S → V_I ADV	NP → ART ADJ N	ADJ → ADJ_C
S → V_T NP		ADJ → ADJ_S ADJ_C
S → V_T NP ADV		

4.2. Experimental Procedure

1. Derive 100 different sentences from the regular grammar. We list all the sentences in Table A.5.
2. Train the MTRNN using the first 80 sentences (No.001 - 080). In each training iteration, we added substitution errors to each sentence with a fixed probability (ρ) as follows.

(i) Go to step (ii) with probability ρ, otherwise exit.

(ii) Replace a randomly chosen alphabet in a target sentence with another randomly chosen alphabet.

(iii) Return to step (i).

3. Sentence emulation task:

Test the trained MTRNN's capability using all of the 100 sentences, and another 20 sentences that are ungrammatical as a control experiment. The testing procedure was as follows.

(i) Recognition: Calculate $\boldsymbol{Csc_0}$ from a sentence.

(ii) Generation : Generate a sentence from the $\boldsymbol{Csc_0}$ gained in (i).

(iii) Comparison: Compare the original and generated sentence.

4. Sentence correction task:

Test the MTRNN's capability to correct the corrupted sentences, each of which includes one or two substitution errors. We used all of the 100 sentences (Table A.5) for this task.

(i) Corruption : Replace one or two randomly chosen alphabet(s) in a target sentence with randomly chosen one(s).

(ii) Recognition: Calculate $\boldsymbol{Csc_0}$ from the corrupted sentence.

(iii) Generation : Generate a sentence from the $\boldsymbol{Csc_0}$ gained in (ii).

(iv) Comparison: Compare the original and generated sentence.

5. Results

In our experiments, we trained 120 MTRNNs (for each 6 values of ρ, 20 individual simulations were done), and tested their performance for two tasks, a sentence emulation task and a sentence correction task. As a result,

16

we were able to find the best training result from the 120 done, which had the best score for both of these tasks. We present the best performance for sentence emulation in Sec. 5.1, and for error correction in Sec. 5.2.

The value of ρ for the best result was 30%. A detailed analysis for the correlation between ρ and the model's performance is given later in Sec. 6.2.

5.1. Performance of Sentence Emulation

We found that our model could correctly emulate 98 of 100 grammatical sentences. We list the sentences that the model failed to emulate in Table 3. For the correct emulation of a sentence, our model needs to acquire a dynamical structure representing the sentence. In other words, a stable trajectory representing the sentence should be formed in the dynamical system of the MTRNN, and its $\boldsymbol{Csc_0}$ should be properly embedded into the initial state space. We define a stable trajectory as a trajectory that has enough attractive power to recover from a given level of perturbation. If a trajectory representing the target sentence is not stable enough, our model can not re-generate the same trajectory from an initial state because its output easily deviates from the target trajectory through calculation errors in forward calculations of the RNN. The fact that the model could emulate 18 of 20 unknown sentences reveals that it had acquired the dynamical structure representing them by generalizing the training sentences.

We also found that none of the 20 ungrammatical sentences for the control experiment were emulated correctly because the recognition error (E (cf. (7)) in the recognition phase) did not adequately fall. Indeed, the average recognition error of the 20 ungrammatical sentences was about 22 times as large as that of the 20 unknown grammatical sentences. This implies that

17

our model can distinguish ungrammatical sentences by using the recognition error.

These results revealed that our model self-organized mirroring linguistic structure adequately only from the sentence set. A detailed analysis of the linguistic structure that emerged in our model is given later in Sec. 6.1.

Table 3: Errors in sentence emulation: input sentences whose number is less than 81 were included in the training set, otherwise not. The emphasized characters in a sentence are different from the original.

Number	Input sentence	Generated sentence
082	"kick a big yellow box."	"kick a <u>**silly**</u>lllow box."
100	"jump quickly."	"jump <u>**slowloxl**</u>"

5.2. Performance of Sentence Correction

We found that our model could correct 83 of 100 sentences, each with one or two substitution errors. The 83 sentences are comprised of 66 of the 80 input sentences whose original versions were included in the training set and 17 of the 20 inputs whose originals were not learned. We list the sentences that the model failed to correct in Table B.6.

A trajectory representing a corrupted sentence can be regarded as perturbed one from the original representing the correct sentence. The perturbation caused by the sentence corruption can be recovered when our model generates a trajectory from Csc_0, if the trajectory representing the correct sentence is robust enough and its attractive area is also sufficiently wide. This is the mechanism that the model uses to correct sentences. Thus, the

success of a sentence correction indicates that there is a robust trajectory representing the sentence.

In conclusion, the result of this experiment reveals that our model can acquire the capability to correct a few errors in a sentence with a high accuracy by self-organization of a robust linguistic structure.

5.3. Cross Validation

So far, we have shown the experimental results in the case where the training set consisted of sentences whose numbers were from 001 to 080. We also examined the generalization capabilities of our model in the same conditions as before, except for the training set. We listed the accuracy for both the emulation and correction task in each case in Table 4.

We found that our model demonstrated almost the same performance in every case. We also found that our model failed to correctly generate all sentences which included the word "run" in the case listed at the top of Table 4. This was because the word "run" never appeared in the training set in that case.

Sentences for Validation	Emulation Task	Correction Task
001 - 020	95/100	84/100
021 - 040	100/100	82/100
041 - 060	96/100	78/100
061 - 080	96/100	81/100
081 - 100 (above-mentioned)	98/100	83/100

Table 4: Results of Cross Validation:

6. Discussion

In this section, we deal with three topics. First, we analyze the hierarchy of the linguistic structure in Sec. 6.1. Second, we reveal the correlation between corruption in the training sentences and robustness of the linguistic structure acquired by our model in Sec. 6.2. Finally, we discuss language acquisition in more difficult condition in Sec. 6.3.

6.1. Linguistic Hierarchy in MTRNN

We claim that our model is self-organized, reflecting the hierarchical linguistic structure, and more precisely that IO neuron activation represents "characters," Cf represents "words," and Cs represents "sentences." In this section, we illustrate the basis of this argument by analyzing our model.

We analyzed the neural activation patterns when the MTRNN generated sentences to reveal the linguistic structures that emerged in the MTRNN. We used the principle component analysis (PCA) technique in our analysis. We summarize the analysis results for each neuron group below.

IO : Each IO neuron corresponds to a character. Thus, their activation patterns obviously represent the "characters" sequences.

Cf : We claim that Cf activations represent "words," including their grammatical information. Our claim is based on the following facts, which we found by analyzing Cf activation patterns.

   1. The same words are represented by almost the same trajectories in the state space of Cf activation.
     This can be confirmed with Fig. 3. Figure 3 shows transitions of

Cf activation for all of the words appearing in the data set (cf. Table A.5).

2. The words in the same category are represented in a similar way. Figure 3 shows that trajectories of words in the same category are located in the same region of the state space. We also show the Cf activations in the first step of each word in Fig. 4. This clearly illustrates that words are clustered by their grammatical roles.

3. The first and the last steps of the words are clustered by their grammatical roles, and the grammatical connection of the words is represented by the closeness of the clusters.

   We can regard this as the structure of the underlying grammar. For a better understanding of the structure, we present some example of the Cf activation transition which represents a whole sentence in Fig. 5.

4. The correspondence between characters and activations disappeared. This is easily confirmed since the activation patterns are different even if the characters are the same.

Cs : We claim that the Cs activation represents "sentences." The bases for our claim are as follows.

1. The initial states of Cs (Csc_0) are clustered mainly by the grammatical structure of the sentences (Fig. 6). The grammatical structure is featured by both the existence of an adverb and the complexity of the objectival phrase. The complexity of the objectival phrase is increasing in the following order.

   (i) sentence with a intransitive verb (e.g. "walk.")

21

(ii) sentence with a transitive verb and no adjectives

(e.g. "kick *a box*.")

(iii) sentence with a transitive verb and 1 adjective

(e.g. "kick *a red box*.")

(iv) sentence with a transitive verb and 2 adjectives

(e.g. "kick *a big red box*.")

2. The correspondence between words and activations disappeared. Even the same words in different sentences are represented in different ways. This can be confirmed with Fig. 7 (e.g. "yellow" in the center and to the right of the figure).
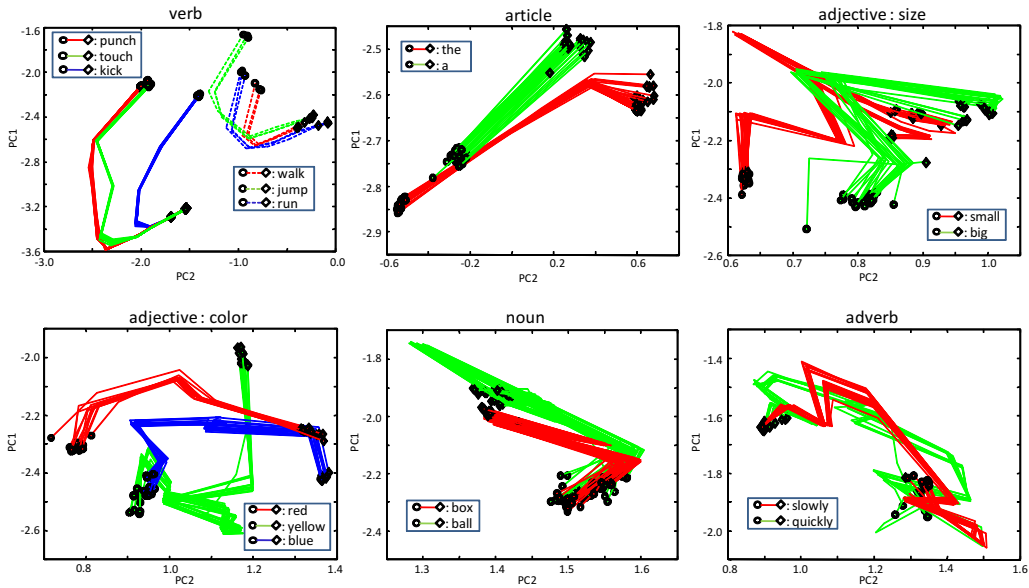

Figure 3: Cf activation transitions of each word: dimensions are reduced from 40 to 2 using PCA (total contribution rate of the 2 components is 77%). The same words have nearly identical trajectories, and words in the same categories have similar trajectories.
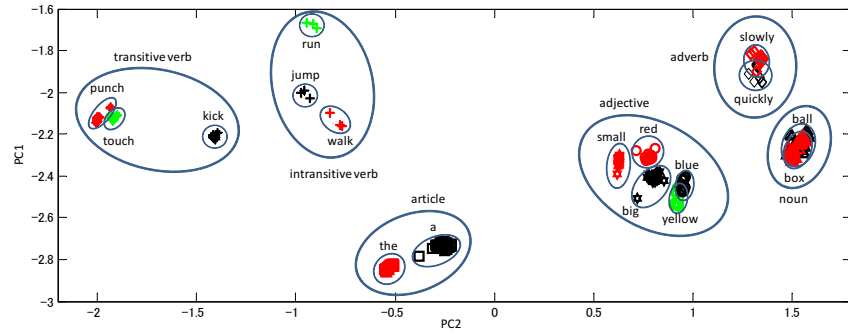
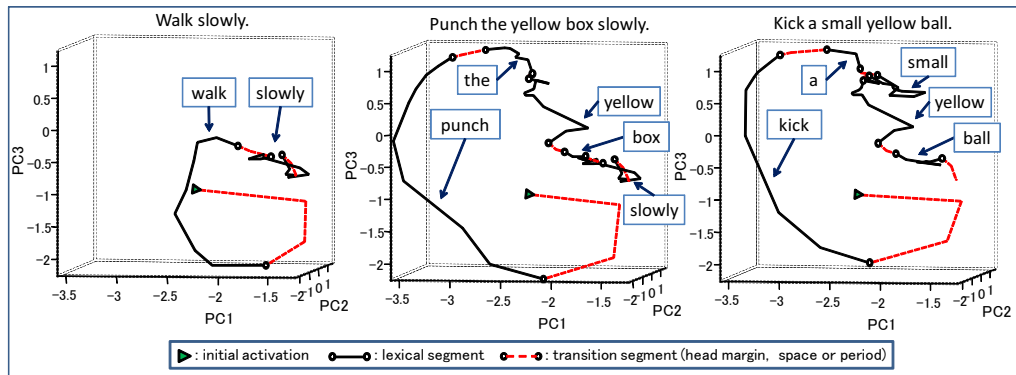Figure 4: Cf activation in the first step of each word: words are clustered based on their categories.

Figure 5: Cf activation transitions of sentence: dimensions are reduced from 40 to 3 using PCA (total contribution rate of the 3 components is 86%). The trajectory of each word ends near the initial states of words which belong to possible next word categories. The three activation patterns in this figure corresponds to the sentences, "walk slowly.," "punch the yellow box slowly.," and "kick a small yellow ball.."
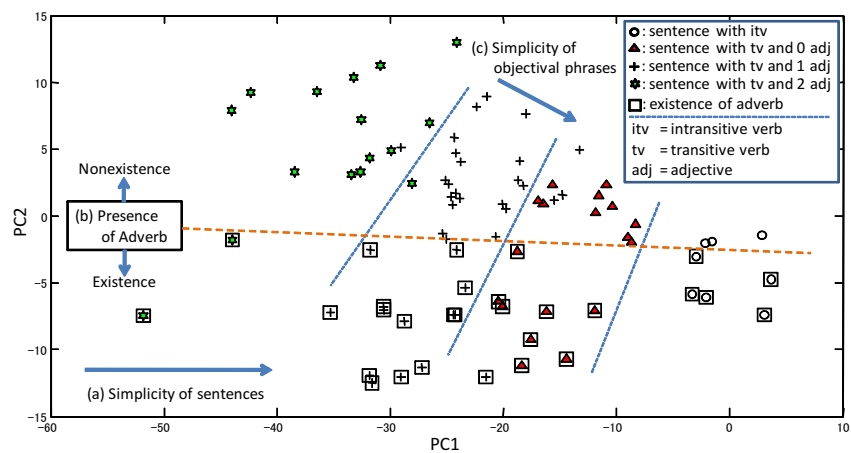
23

Figure 6: Initial state of Cs ($\boldsymbol{Csc_0}$): dimensions are reduced from 6 to 2 using PCA (total contribution rate is 90%). The sentences are clustered based on their grammatical structure. (a) The value of PC1 seems to be negatively correlated with the number of words in the sentence, namely the complexity of the sentence. (b) There seems to be a PC2 threshold that separates whether a sentence has an adverb or not. (c) Focusing on the number of words in an objectival phrase, there seems to be an axis that correlates with it.
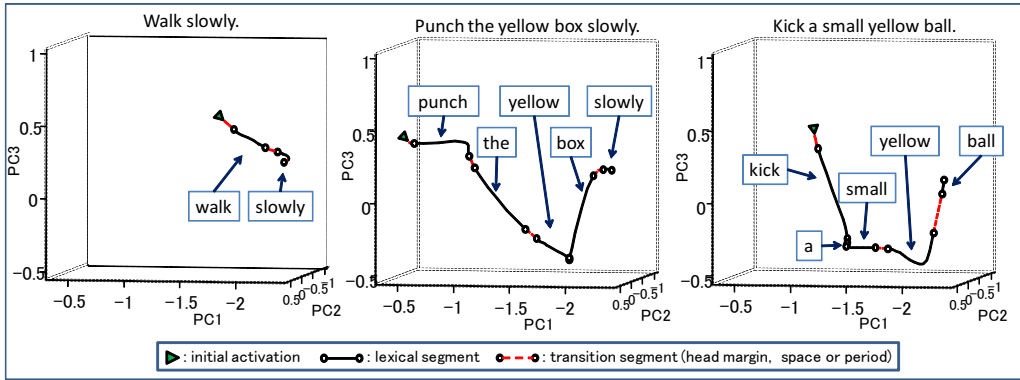
Figure 7: Cs activation transitions: dimensions are reduced from 11 to 3 by PCA (total contribution rate of the 3 components is 95%). In different sentences, even the same words are represented in different ways. The three activation patterns in this figure corresponds to the sentences, "walk slowly.," "punch the yellow box slowly.," and "kick a small yellow ball.."

6.2. Correlation between Corruption in Training Sentences and Robustness of Linguistic Structure

We analyzed the correlation between the error incidence probability ρ and the accuracy of the sentence emulation and correction task in order to investigate how corruption in the training sentences impacts the linguistic structure acquired by our model. For each $\rho = 0, 5, 10, 20, 30, 40(\%)$, we calculated the accuracy of the emulation and the correction task by averaging the top three accuracies of the 20 trials. Figure 8 shows the results.

From the results, we observe the following:

1. Adding errors to the training sentences considerably improves the capability to correct sentences.

2. Adding errors to the training sentences also improves the capability to emulate, or recognize and generate, sentences with no errors.

25

3. Adding too many errors to the training sentences reduces the capability to emulate and correct sentences.

In conclusion, we claim that the robustness of the linguistic structures that emerge in our model is improved by adding some substitution errors to the training sentences with a certain probability. We presume this is because attractive areas of a trajectory representing a sentence are broadened by training the model with perturbed trajectories.
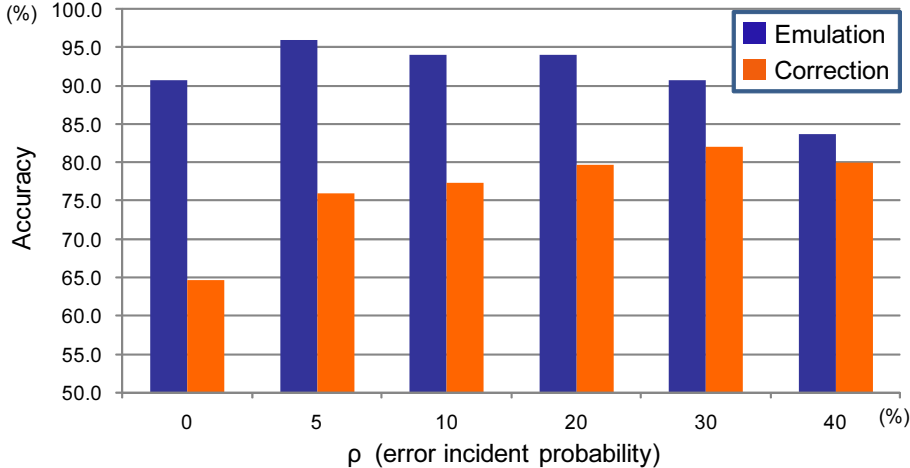

Figure 8: Accuracy against ρ (error incidence probability)

6.3. Language Acquisition in More Difficult Condition

We discuss language acquisition in more difficult condition. First, we show the result of an experiment where we trained our model without delimiters in Sec. 6.3.1. Second, we discuss possibilities to expand our model to learn a language with a more complex grammar in Sec. 6.3.2.

26

6.3.1. Acquisition of a Language without Delimiters

We found that a structure reflecting the lexicon emerged in the Cf. How did our model extract words from sentences represented as sequences of characters? Did it take advantage of white-space characters as delimiters? To answer this question, we trained and tested the MTRNN using a sentence set without white-space characters. Experimental conditions are the same as those described in Sec. 4, except all white-space characters are omitted from the sentences. As a result, we confirmed that the MTRNN performed well with an emulation accuracy of 96/100, a correction accuracy of 81/100 (those scores compares favorably with the best result presented in Sec. 5). Therefore, the linguistic structure can emerge regardless of whether or not there are word delimiters in sentences. We presume that the structure of words is learned through MTRNNs' tendency to reuse trajectories that represent a common substring in sentences in order to memorize them in the simplest way.

6.3.2. Acquisition of Language with More Complex Grammar

We used a simple language with a regular grammar for our experiment. That does not, however, imply an intrinsic limitation of RNNs' capability to acquire a grammar. Some recurrent architectures have been shown to be computationally Turing equivalent [14, 11]. Indeed, it has been confirmed that some RNN models can learn context-free language, and also learn context-sensitive and recursive language when their complexity is limited [7, 18, 4]. Thus, we presume it is possible to expand our model to learn a language with more complex grammars.

7. Conclusion

In this paper, we reported on language acquisition by the MTRNN. We trained the model to learn a language using only a sentence set without any previous knowledge of the words or grammar, but only of the character set. As a result of our experiments, we found that the model could acquire the capabilities to recognize and generate grammatical sentences even if they were not learned. Moreover, our model could correct one or two substitution errors in a sentence with high accuracy. Therefore, we showed that our model can self-organize reflecting linguistic structures by generalizing a sentence set.

To reveal the linguistic structure, we analyzed the neural activation patterns in each neuron group. As a result of the analysis, we found that our model self-organized, reflecting language hierarchy, taking advantage of the difference in time scales among neuron groups. More precisely, the IO neurons represented "characters," the Cf neurons represented "words," and the Cs neurons represented "sentences." The model recognizes and generates sentences through the interaction among these three levels.

We proved through our experimentation that a neural system such as an MTRNN can self-organize mirroring the hierarchical structure of language (e.g. characters $\rightarrow$ words $\rightarrow$ sentences) by generalizing a sentence set, and it can recognize and generate new sentences using the acquired structure. This result implies that the requirements for language acquisition are not innate faculties of language, but appropriate architectures of a neural system. We insist that multiplicity of time scale among neurons is a potential candidate for the requirements.

We also revealed that the robustness of the linguistic structure that

emerged in our model was improved when we added some errors to the training sentences with a certain probability. It is usually believed that the poor quality of linguistic stimuli makes it difficult for children to acquire a language. However, our result implies there is a possibility that certain levels of corruption in the linguistic stimuli can improve the robustness of the linguistic cognitive capabilities acquired from them.

For our future work, we intend to deal with language acquisition from the viewpoint of the interaction between linguistic cognition and other types of cognition (this is the viewpoint of cognitive linguists). In concrete terms, we are going to connect the language MTRNN with another MTRNN dealing with the sensori-motor system of a robot. We expect the robot to acquire a language grounded on its sensori-motor cognition using the dynamical interaction between the two MTRNNs.

Acknowledgement

Table A.5: The list of sentences

Number	sentence	Number	sentence
001	"jump slowly."	051	"punch the red box."
002	"punch the small ball."	052	"punch the big red box."
003	"run quickly."	053	"punch the small yellow ball."
004	"punch the ball quickly."	054	"touch a small ball."
005	"punch the small yellow box slowly."	055	"punch a blue box quickly."
006	"kick the big box."	056	"punch the yellow box slowly."
007	"touch a ball slowly."	057	"punch the box slowly."
008	"run slowly."	058	"punch the box quickly."
009	"touch the big box."	059	"touch a big ball."
010	"kick the box."	060	"touch a small red ball."
011	"punch a small box."	061	"kick a small yellow ball."
012	"punch the yellow box."	062	"touch the blue box."
013	"kick a box."	063	"punch the small ball slowly."
014	"touch a blue ball."	064	"touch a yellow box slowly."
015	"run."	065	"touch a ball quickly."
016	"punch a box."	066	"kick the big ball quickly."
017	"kick a ball."	067	"punch the small yellow box."
018	"punch the small blue box quickly."	068	"touch the big blue ball."
019	"touch the blue ball slowly."	069	"touch the box slowly."
020	"punch the box."	070	"kick the ball."
021	"punch a yellow box."	071	"kick the small yellow box."
022	"touch a ball."	072	"kick the blue ball."
023	"kick the big ball."	073	"punch a yellow ball."
024	"touch the small ball slowly."	074	"kick the small blue box."
025	"punch the yellow ball."	075	"punch the big box."
026	"punch the big ball quickly."	076	"punch the small blue ball."
027	"walk slowly."	077	"touch the big ball slowly."
028	"punch the blue box slowly."	078	"touch the small ball."
029	"kick the big blue box."	079	"walk quickly."
030	"punch the small ball quickly."	080	"kick the ball slowly."
031	"touch a yellow ball."	081	"touch the blue ball."
032	"punch the ball slowly."	082	"kick a big yellow box."
033	"touch the box."	083	"punch the big blue box."
034	"touch a small yellow ball."	084	"punch the big ball."
035	"punch the small box quickly."	085	"punch a big box."
036	"kick the red box."	086	"touch a blue box."
037	"touch a box."	087	"touch the small blue box."
038	"punch the big yellow ball."	088	"punch the small box."
039	"punch a ball."	089	"touch the small box."
040	"punch a big ball slowly."	090	"touch the big blue box."
041	"jump."	091	"kick the big red box."
042	"touch a small box."	092	"punch the small red ball."
043	"walk."	093	"kick the big red ball."
044	"kick a big red box."	094	"punch a box quickly."
045	"kick a red ball."	095	"kick the small ball slowly."
046	"touch a big box."	096	"punch a ball slowly."
047	"touch a small ball quickly."	097	"kick the small red box."
048	"touch a red box."	098	"touch a red ball."
049	"punch the big red ball."	099	"punch the small box slowly."
050	"touch the yellow ball."	100	"jump quickly."

Table B.6: Errors in Sentence Correction: the original version of corrupted sentence whose number is less than 81 was included in training set, otherwise not. The emphasized characters in a sentence are different from the original.

Number	Corrupted sentence	Generated sentence
001	"jump s**r**owly."	"jump slowly**x**"
002	"punch t**e**e small ball."	"punch the smal **bu**all."
003	"**o**un quickly."	"run **slowl sl**"
008	"run sl**q**wly."	"run slowly**x**"
015	"ru**j**."	"**jump**"
016	"**t**unch a box."	"**t**unch a box."
021	"punch a yel**k**ow box."	"punch a yellow b**x**_."
023	"kick t**u**e big ball."	"kick the big bal**.**."
026	"punch the big b**e**ll quickly."	"punch the big ball qui**lly.**."
038	"punch the big yel**p**ow ball."	"punch the big yel**ow**w bal**.**."
040	"punch a b**o**g ball slowly."	"punch a big**l**_all sl**l**wl**ly**"
042	"**p**ouch a small box."	"**p**ouch a small box."
045	"kick a red b**o**ll."	"kick a red b**ox**l."
060	"touch a small **q**ed ball."	"touch a small **b**ed ball."
082	"kick a b**o**g yellow box."	"kick a **y ll y**llow box."
094	"punch a **x**ox quickly."	"punch a box qu**l.l**ly."
098	"touch **d** **b**ed ball."	"touch a **b**ed ball."

Appendix A. The list of sentence

Appendix B. Errors in Sentence Correction

Appendix C. Derivation of the BPTT

Appendix C.1. Supplemental Explanation for Equation (4)

$$\frac{\partial E}{\partial w_{ij}} = \sum_t \frac{\partial E}{\partial u_{t,i}} \cdot \frac{\partial u_{t,i}}{\partial w_{ij}} = \sum_t \frac{\partial E}{\partial u_{t,i}} \cdot \frac{1}{\tau_i} \cdot x_{t,j} = \frac{1}{\tau_i} \sum_t x_{t,j} \frac{\partial E}{\partial u_{t,i}} \qquad \text{(C.1)}$$

Appendix C.2. Supplemental Explanation for Equation (5)

$$\begin{aligned}
\frac{\partial E}{\partial b_i} &= \sum_t \frac{\partial E}{\partial (u_{t,i} + b_i)} \cdot \frac{\partial (u_{t,i} + b_i)}{\partial b_i} = \sum_t \frac{\partial E}{\partial (u_{t,i} + b_i)} \cdot 1 \\
&= \sum_t \frac{\partial E}{\partial (u_{t,i} + b_i)} \cdot \frac{\partial (u_{t,i} + b_i)}{\partial u_{t,i}} = \sum_t \frac{\partial E}{\partial u_{t,i}} \qquad \text{(C.2)}
\end{aligned}$$

Appendix C.3. Derivation of Equation (8a)

Here, we use the "Kronecker delta" (δ_{ik}) defined as follows. $\delta_{ik} = \begin{cases} 1 & \cdots (i = k) \\ 0 & \cdots (i \neq k) \end{cases}$

$$\frac{\partial E}{\partial u_{t,i}} = \sum_{k \in I_{IO}} \frac{\partial E}{\partial y_{t,k}} \cdot \frac{\partial y_{t,k}}{\partial u_{t,i}} + \frac{\partial E}{\partial u_{t+1,i}} \cdot \frac{\partial u_{t+1,i}}{\partial u_{t,i}}$$

$$= \sum_{k \in I_{IO}} \frac{\partial E}{\partial y_{t,k}} \cdot \frac{\partial y_{t,k}}{\partial u_{t,i}} + \frac{\partial E}{\partial u_{t+1,i}} \cdot \left(1 - \frac{1}{\tau_i}\right)$$

$$= \sum_{k \in I_{IO}} \frac{\partial E}{\partial y_{t,k}} \cdot \frac{\delta_{ik} \exp(u_{t,k} + b_k) \sum_{m \in I_{IO}} \exp(u_{t,m} + b_m) - \exp(u_{t,k} + b_k) \exp(u_{t,i} + b_i)}{(\sum_{m \in I_{IO}} \exp(u_{t,m} + b_m))^2}$$

$$+ \frac{\partial E}{\partial u_{t+1,i}} \cdot \left(1 - \frac{1}{\tau_i}\right)$$

$$= \sum_{k \in I_{IO}} \frac{\partial E}{\partial y_{t,k}} \cdot \left(\frac{\delta_{ik} \exp(u_{t,k} + b_k)}{\sum_{m \in I_{IO}} \exp(u_{t,m} + b_m)} - y_{t,k} \cdot y_{t,i}\right) + \frac{\partial E}{\partial u_{t+1,i}} \cdot \left(1 - \frac{1}{\tau_i}\right)$$

$$= \frac{\partial E}{\partial y_{t,i}} \cdot y_{t,i} - \sum_{k \in I_{IO}} \frac{\partial E}{\partial y_{t,k}} \cdot y_{t,k} \cdot y_{t,i} + \frac{\partial E}{\partial u_{t+1,i}} \cdot \left(1 - \frac{1}{\tau_i}\right) \qquad \cdots (i \in I_{IO})$$

$$(C.3)$$

We define $E(t)$ as follows.

$$E(t) = \sum_{i \in I_{IO}} y_{t,i}^* \cdot \log\left(\frac{y_{t,i}^*}{y_{t,i}}\right) \qquad \left(E = \sum_{t} E(t)\right)$$

Now, we introduce the following approximation.

$$\frac{\partial E}{\partial y_{t,k}} \simeq \frac{\partial E(t)}{\partial y_{t,k}} = y_{t,k}^* \cdot \frac{y_{t,k}}{y_{t,k}^*} \cdot \left(\frac{-y_{t,k}^*}{y_{t,k}^2}\right) = -\frac{y_{t,k}^*}{y_{t,k}} \qquad (C.4)$$

Substituting Eq. (C.4) into Eq. (C.3), we obtain the following.

$$\frac{\partial E}{\partial u_{t,i}} = -\frac{y_{t,i}^*}{y_{t,i}} \cdot y_{t,i} - \sum_{k \in I_{IO}} \left(-\frac{y_{t,k}^*}{y_{t,k}}\right) \cdot y_{t,k} \cdot y_{t,i} + \frac{\partial E}{\partial u_{t+1,i}} \cdot \left(1 - \frac{1}{\tau_i}\right)$$

$$= -y_{t,i}^* + y_{t,i} \cdot \sum_{k \in I_{IO}} y_{t,k}^* + \frac{\partial E}{\partial u_{t+1,i}} \cdot \left(1 - \frac{1}{\tau_i}\right)$$

$$= -y_{t,i}^* + y_{t,i} \cdot \sum_{k \in I_{IO}} \frac{\exp(u_{t,k}^*)}{\sum_{m \in I_{IO}} \exp(u_{t,m}^*)} + \frac{\partial E}{\partial u_{t+1,i}} \cdot \left(1 - \frac{1}{\tau_i}\right)$$

$$= y_{t,i} - y_{t,i}^* + \left(1 - \frac{1}{\tau_i}\right) \frac{\partial E}{\partial u_{t+1,i}} \qquad \cdots (i \in I_{IO}) \qquad (C.5)$$

Appendix C.4. Derivation of Equation (8b)

$$
\begin{aligned}
\frac{\partial E}{\partial u_{t,i}} &= \frac{\partial E}{\partial y_{t,i}} \cdot \frac{\partial y_{t,i}}{\partial u_{t,i}} + \frac{\partial E}{\partial u_{t+1,i}} \cdot \frac{\partial u_{t+1,i}}{\partial u_{t,i}} \\
&= \frac{\partial E}{\partial y_{t,i}} \cdot \frac{\partial y_{t,i}}{\partial u_{t,i}} + \frac{\partial E}{\partial u_{t+1,i}} \cdot \left(1 - \frac{1}{\tau_i}\right) \\
&= \frac{\partial E}{\partial y_{t,i}} \cdot y_{t,i}(1 - y_{t,i}) + \frac{\partial E}{\partial u_{t+1,i}} \cdot \left(1 - \frac{1}{\tau_i}\right) \\
&= y_{t,i}(1 - y_{t,i}) \cdot \frac{\partial E}{\partial x_{t+1,i}} + \frac{\partial E}{\partial u_{t+1,i}} \cdot \left(1 - \frac{1}{\tau_i}\right) \\
&= y_{t,i}(1 - y_{t,i}) \sum_{k \in I_{all}} \frac{\partial E}{\partial u_{t+1,k}} \cdot \frac{\partial u_{t+1,k}}{\partial x_{t+1,i}} + \frac{\partial E}{\partial u_{t+1,i}} \cdot \left(1 - \frac{1}{\tau_i}\right) \\
&= y_{t,i}(1 - y_{t,i}) \sum_{k \in I_{all}} \frac{\partial E}{\partial u_{t+1,k}} \cdot \frac{1}{t_k} \cdot w_{ki} + \frac{\partial E}{\partial u_{t+1,i}} \cdot \left(1 - \frac{1}{\tau_i}\right) \\
&= y_{t,i}(1 - y_{t,i}) \sum_{k \in I_{all}} \frac{w_{ki}}{t_k} \frac{\partial E}{\partial u_{t+1,k}} + \left(1 - \frac{1}{\tau_i}\right) \frac{\partial E}{\partial u_{t+1,i}} \qquad \cdots (i \notin I_{IO})
\end{aligned}
$$

$$(C.6)$$

References

[1] Bot, K.D., Lowie, W., & Verspoor, M. (2007). A dynamic systems theory approach to second language acquisition. *Bilingualism: Language and Cognition*, 10, 7–21.

[2] Chomsky, N. (1986). *Barrier*. MIT Press.

[3] Chomsky, N. (2005). *Rules and Representations*. Columbia University Press.

[4] Christiansen, M.H., & Chater, N. (1999). Toward a connectionist model

of recursion in human linguistic performance. *Cognitive Science*, 23, 157–205.

[5] Cleeremans, A., Servan-Schreiber, D., & McClelland, J.L. (1989). Finite state automata and simple recurrent networks. *Neural Computation*, 1(3), 372–381.

[6] Devaney, R.L. (2003). *An introduction to chaotic dynamical systems.* Westview Press.

[7] Elman, J.L. (1995). Language as a dynamical system. In Port, R., & Gelder, T.v , editors, *Mind as Motion: Explorations in the Dynamics of Cognition*, (pp. 195–223). MIT Press, Cambridge, MA.

[8] Elman, J.L. (1990). Finding structure in time. *Cognitive Science*, 14, 179–211.

[9] Elman, J.L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7(2-3), 195–225.

[10] Elman, J.L. (1993). Learning and development in neural networks: the importance of starting small. *Cognition*, 48(1), 71–99.

[11] Frasconi, P., & Gori, M. (1996). Computational capabilities of local-feedback recurrent networks acting as finite-state machines. *IEEE Transactions on Neural Networks*, 7(6).

[12] Giles, C.L., Miller, C.B., Chen, D., Chen, H.H., Sun, G.Z., & Lee, Y.C. (1992). Learning and extracting finite state automata with second-order recurrent neural networks. *Neural Computation*, 4(3), 393–405.

[13] Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. In *Proc. NatL Acad. Sci.*, volume 79, (pp. 2554–2558).

[14] Lawrence, S., Giles, C.L., & Fong, S. (2000). Natural language grammatical inference with recurrent neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 12(1), 126–140.

[15] Namikawa, J., & Tani, J. (2010). Learning to imitate stochastic time series in a compositional way by chaos. *Neural Networks*, 23, 625–638.

[16] Ogata, T., Murase, M., Tani, J., Komatani, K., & Okuno, H.G. (2007). Two-way translation of compound sentences and arm motions by recurrent neural networks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-2007)*, (pp. 1858–1863).

[17] Pollack, J.B. (1991). The induction of dynamical recognizers. *Machine Learning*, 7(2–3), 227–252.

[18] Rodriguez, P. (2001). Simple recurrent networks learn context-free and context-sensitive languages by counting. *Neural Computation*, 13(9).

[19] Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning internal representations by error propagation. In Rumelhart, D., McClelland, J., & Williams, R., editors, *Parallel Distributed Processing*, (pp. 318–362). MIT Press.

[20] Smolensky, P. (1986). Information processing in dynamical systems: Foundation of harmony theory. In Rumelhart, D., McClelland, J., &

Williams, R., editors, *Parallel Distributed Processing*, chapter 6, (pp. 194–281). MIT Press.

[21] Steels, L. (1997). The synthetic modeling of language origins. *Evolution of Communication*, 1(1), 1–34.

[22] Sugita, Y., & Tani, J. (2005). Learning semantic combinatoriality from the interaction between linguistic and behavioral processes. *Adaptive Behavior*, 13(1), 33–52.

[23] Tani, J., & Ito, M. (2003). Self-organization of behavioral primitives as multiple attractor dynamics: A robot experiment. *IEEE Trans. on Systems, Man, and Cybernetics Part A: Systems and Humans*, 33(4), 481–488.

[24] Weckerly, J., & Elman, J.L. (1992). A pdp approach to processing center-embedded sentences. In *Fourteenth Annual Conference of the Cognitive Science Society*, volume 14, (pp. 414–419). Routledge.

[25] Yamashita, Y., & Tani, J. (2008). Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment. *PLoS Comput. Biol.*, 4.