

Context-Dependent Processing of Spatiotemporal Patterns Based on Interaction Between Neurodynamical Systems

Takashi Hasuo¹, Ken Yamane¹, and Masahiko Morita¹

Graduate School of Systems and Information Engineering, University of Tsukuba,
Tsukuba-city, 305-8573, Japan
hasuo@bcl.esys.tsukuba.ac.jp

Abstract. Dynamics of traditional neural network models are generally time-invariant. For that reason, they have limitations in context-dependent processing. We present a new method, dynamic desensitization, of varying neurodynamics continuously and construct a basic model of interaction between neurodynamical systems. This model comprises two nonmonotone neural networks storing sequential patterns as trajectory attractors. The dynamics of respective networks are modified according to the states of other networks. Using numerical experiments, we also show that the model can recognize and recall complex sequences with identical patterns in different positions.

1 Introduction

The brain can be regarded as an assemblage of dynamical systems comprising many neurons that perform spatio-temporal pattern processing. Consequently, neurodynamical systems comprising artificial neural networks with recurrent connections and continuous-time dynamics are thought to offer great potential for brain-like information processing. However, existing neurodynamical systems have limited capability, particularly in terms of context-dependent processing.

Two explanations for this limited ability are the following. First, the dynamics are usually time-invariant (at least in the short run). For that reason, the system always makes the same transition at the same state unless it receives external input. Second, no appropriate method is known for making different neurodynamical systems interact. If two networks are interconnected in a usual manner, they constitute a single neurodynamical system with time-invariant dynamics, rather than two different systems. In such a case, the influence of system A on system B depends on the state of A, but not on that of B.

In this paper, we propose a method of varying the dynamics of a recurrent neural network according to another network. We construct a model for context-dependent processing such as recognition and recall of complex sequential patterns.

2 Dynamic Desensitization

As a method of varying the dynamics of a neural network discretely, Morita et al.[1] proposed selective desensitization. This method desensitizes about half

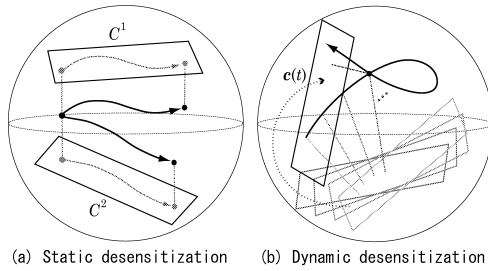


Fig. 1. Methods of varying neurodynamics: Thick arrows represent trajectories of the network state. Planes represent subspaces. The dashed arrows represent trajectory attractors in the subspaces.

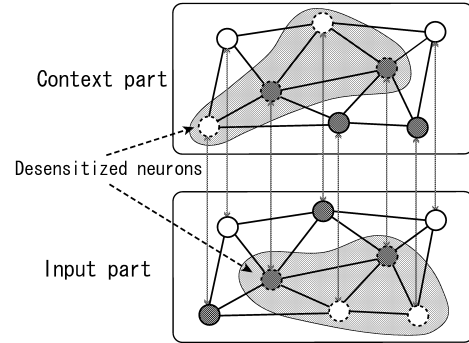


Fig. 2. Model structure.

of the neurons or renders their output as neutral, depending on a given modification pattern. Through this operation, the modified state of the network is projected onto a subspace of the state space, and produces transitions according to the dynamics in the subspace. If trajectory attractors[2] are formed in different subspaces, the network state reaches different target patterns according to modification patterns, as shown schematically in Fig. 1(a).

In this model, however, the modification pattern is restricted to a static pattern so that trajectory attractors can be formed within individual subspaces. For that reason, we call this “static desensitization.”

Here we propose “dynamic desensitization” by extending the modification pattern of static desensitization to a dynamic or spatiotemporal pattern. Then the set of desensitized neurons, or the subspace onto which the network state is projected, varies continuously with time, as depicted in Fig. 1(b). In this case, a single trajectory attractor to the target state cannot be formed. If the change of the modification pattern is sufficiently slow, however, the network state makes short transitions in each of a series of subspaces and can thereby reach the target.

3 Model

Using the method described above, we constructed a basic model of interaction between neurodynamical systems (Fig. 2). This model consists of two nonmonotone neural networks that have the same number of neurons and obey the same dynamic equation.

Specifically, the i -th neuron ($i=1, \dots, n$) of either network acts according to

$$\tau \frac{du_i}{dt} = -u_i + \sum_{j=1}^n w_{ij} y_j + z_i, \quad y_i = g(v_i) \cdot f(u_i), \quad (1)$$

where u_i denotes the instantaneous potential, y_i is the output, z_i is the external input. In addition, w_{ij} represents the synaptic weight from the j -th neuron of

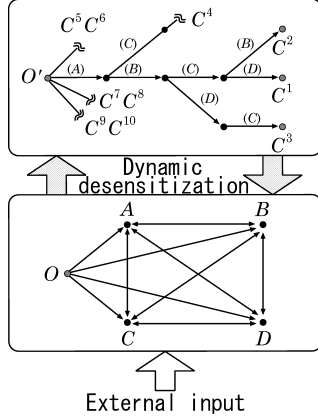


Fig. 3. State transition diagram.

Table 1. Input sequences: Letters represent n -dimensional binary patterns selected randomly; $\{OABCD\}_T$ denotes a spatiotemporal pattern changing from a pattern O via A , B and C into D over a period of T . Each trajectory overlaps with others in every unit section, e.g., $\{AB\}$ is included in \mathbf{s}^1 , \mathbf{s}^2 and \mathbf{s}^3 , and $\{BC\}$ is included in \mathbf{s}^1 , \mathbf{s}^2 and \mathbf{s}^5 .

$\mathbf{s}^1 = \{OABCD\}_T$	$\mathbf{s}^2 = \{OABCB\}_T$
$\mathbf{s}^3 = \{OABDC\}_T$	$\mathbf{s}^4 = \{OACBD\}_T$
$\mathbf{s}^5 = \{OBCBD\}_T$	$\mathbf{s}^6 = \{OBDAC\}_T$
$\mathbf{s}^7 = \{OCADB\}_T$	$\mathbf{s}^8 = \{OCADA\}_T$
$\mathbf{s}^9 = \{ODADC\}_T$	$\mathbf{s}^{10} = \{ODCDA\}_T$

the same network, v_i is the instantaneous potential of the corresponding neuron of the other network, and τ is a time constant. Also, $g(v_i)$ is a sigmoid function which takes a value from 0 to 1, and $f(u_i)$ is a nonmonotonic output function[2]. If v_i takes a positive large value, the output is $y_i \simeq f(u_i)$, but if v_i takes a negative large value, then $g(v_i)$ and y_i are nearly zero, which means that the neuron is desensitized. Here we consider $x_i \equiv \text{sgn}(u_i)$ ($\text{sgn}(u) = 1$ for $u > 0$ and -1 for $u \leq 0$) and refer to the binary vector $\mathbf{x} = (x_1, \dots, x_n)$ as the current state of the network.

The two networks have a similar structure, but they differ in function (Fig. 3). One network receives an input of complex spatiotemporal patterns from the outside and stores them. This network is termed the input part, but it also shows the recalled pattern when no input pattern is fed. The other network, termed the context part, does not receive the external input directly, but it autonomously changes its state from a fixed initial state O' , via some branch points, to a terminal state C^μ in accordance with the state transitions of the input part.

4 Computer Simulation

We carried out simulation experiments using the 10 spatiotemporal patterns $\mathbf{s}^1, \dots, \mathbf{s}^{10}$ presented in Table 1. Their dimensions (and size of each network) were $n = 800$ and their temporal length was $T = 20\tau$. We also generated $\mathbf{c}^\mu = \{O' \bullet \bullet \bullet C^\mu\}_T$ ($\mu = 1, \dots, 10$) corresponding to \mathbf{s}^μ , where \bullet represents a branch point described above. Subsequently, we trained the model by giving \mathbf{s}^μ and \mathbf{c}^μ as learning signals to the input and context parts, respectively, to store them as trajectory attractors (see refs. [2] and [1] for the learning algorithm).

After completion of 30 training cycles, we gave various input patterns and examined the behavior of the model. Figure 4 presents two examples in which (a) \mathbf{s}^2 was fed with 10% noise and with temporal expansion and contraction, and (b) only short sections of \mathbf{s}^9 were fed. The state of the context part reaches C^2 in (a) and C^9 in (b), meaning that the model recognized the input pattern as

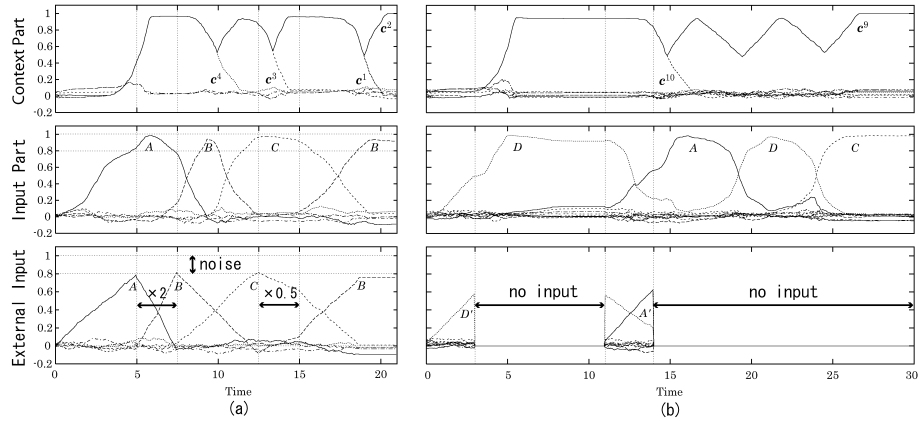


Fig. 4. Behavior of the model: Similarities of the network state (upper and middle graphs) or the current input (lower graph) to reference patterns are plotted against time scaled by the time constant τ .

s^2 or s^9 . It is also apparent that a spatiotemporal pattern similar to the original is recalled in the input part.

This result, together with those of many other experiments, indicates that the two networks act autonomously and cooperatively through mutual dynamic desensitization.

5 Concluding Remarks

We have proposed a dynamic desensitization method to modify a neurodynamical system using another system. We have also shown that two systems which are interconnected by this method can recognize and recall complex sequences with many overlapping sections, even if they have substantial noise, have some blank sections, or are temporally expanded or contracted.

Interaction through dynamic desensitization differs greatly from interaction which takes place through ordinary synaptic connections between neurons: the influence on dynamics of one system depends on the states of both systems. This method also presents an advantage in that the number of interconnections required is merely n . By developing this method, more than two neurodynamical systems can interact to provide more powerful capability, which remains as a subject for future study.

References

1. Morita, M., Murata, S., Morokami, S.: Context-dependent sequential recall by a trajectory attractor network with selective desensitization. Proceedings of the Third International Conference on Neural Networks and Artificial Intelligence. (2003) 235–238
2. Morita, M.: Memory and learning of sequential patterns by nonmonotone neural network. *Neural Networks* **9** (1996) 1477–1489