Top-down and Bottom-up Processing of Spatiotemporal Patterns in a Fully Recurrent Network of Nonmonotonic Neurons

Satoshi Murakami †and Masahiko Morita ††

†Doctoral Program in Engineering, University of Tsukuba, Ibaraki 305-8573, Japan

††Institute of Engineering Mechanics and Systems, University of Tsukuba, Ibaraki 305-8573, Japan

Email: mare@bcl.esys.tsukuba.ac.jp

Abstract

In the present paper, a fully recurrent neural network with a nonmonotonic activation function that treats temporal sequences without expanding them into spatial patterns is described. This network associates a complex spatiotemporal pattern with a simple one using trajectory attractors formed by simple learning. Computer simulations show that the model not only has high recognition and generation abilities but can also perform advanced processing using bidirectional interactions.

Keywords: temporal sequence, bidirectional processing, recognition and generation, nonmonotone neural network.

1. Introduction

We previously proposed a new type of model for spatiotemporal pattern processing [1,2] using a nonmonotone neural network [3] or a fully recurrent neural network with a nonmonotonic activation function.

This model can recognize temporal sequences without expanding them into spatial patterns by delays, since it is carried out by converting complex spatiotemporal patterns that have long, overlapping trajectories in the pattern space into simple ones with short, separate trajectories. Such conversion can be regarded as bottom-up processing.

Conversely, it is possible, in principle, for the model to generate a complex spatiotemporal pattern from a simple one, which is regarded as top-down processing.

In the present study, we improve the model so that it can convert patterns bidirectionally and perform top-down and bottom-up processing simultaneously. By doing so, we expect to realize advanced processing of temporal sequences as performed by the brain.

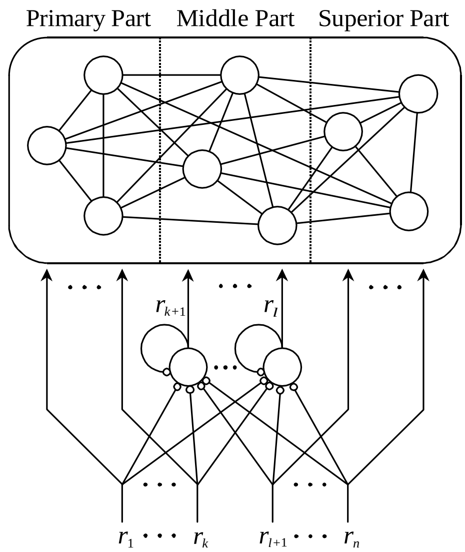

Fig. 1: Architecture of the model.

2. Architecture of the Model

The architecture of the model is shown in Fig. 1, which is the same as the previous model [2], but input and output parts are renamed as primary and superior parts, respectively.

The network in the top half is called the association network, where complex spatiotemporal patterns are associated with simple ones. This network has a simple structure composed of n nonmonotonic neurons with fully recurrent connections. These neurons are divided into three parts, primary, middle and superior, though all the neurons obey the same dynamics and learning rule. The primary and superior parts treat complex and simple spatiotemporal patterns, respectively;

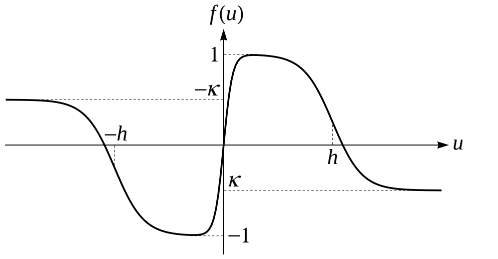

Fig. 2: Nonmonotonic activation function.

the middle part is composed of hidden neurons and mediates between the two parts.

For convenience, we give serial numbers to the neurons such that neurons 1 to k comprise the primary part, $k+1$ to l the middle part, and $l+1$ to n the superior part.

In recognition, a complex spatiotemporal pattern is given to the primary part, and a corresponding simple one appears in the superior part. In generation, conversely, a simple pattern is given to the superior part and the corresponding complex one is reproduced in the primary part.

The network in the bottom half of the figure is called the training network, which makes a learning signal $\boldsymbol{r}_{\rm mid} = (r_{k+1}, \ldots, r_l)$ for the middle part from $\boldsymbol{r}_{\rm pri} = (r_1, \ldots, r_k)$ and $\boldsymbol{r}_{\rm sup} = (r_{l+1}, \ldots, r_n)$ [2]. This network consists of $l-k$ binary neurons corresponding one-to-one to the hidden neurons in the middle part. Each neuron has a self-connection of a positive strength ρ, receives $\boldsymbol{r}_{\rm pri}$ and $\boldsymbol{r}_{\rm sup}$ through random synaptic weight a_{ij} and b_{ij}, and outputs r_i $(k < i \le l)$. In mathematical terms,

$$r_i = \mathrm{sgn}\left(\sum_{j=1}^{k} a_{ij} r_j + \sum_{j=l+1}^{n} b_{ij} r_j + \rho r_i\right), \quad (1)$$

where $r_i = -1$ for $t < 0$. In the following experiments, a_{ij} are normally distributed random numbers with mean $1/k$ and variance $1/k$, b_{ij} are those with mean $1/(n-l)$ and variance $1/(n-l)$, and $\rho = 1$.

3. Dynamics

Dynamics of the association network are expressed by

$$\tau \frac{du_i}{dt} = -u_i + \sum_{j=1}^{n} w_{ij} y_i + z_i, \quad (2)$$

where u_i is the potential of neuron i, and w_{ij} is the synaptic weight from neuron j, z_i is the external input, and τ is a time constant. The output y_i is given by

$$y_i = f(u_i), \quad (3)$$

where $f(u)$ is a nonmonotonic function as shown in Fig. 2. We use, as the nonmonotonic activation function,

$$f(u) = \frac{1 - e^{-cu}}{1 + e^{-cu}} \cdot \frac{1 + \kappa e^{c'(|u|-h)}}{1 + e^{c'(|u|-h)}}, \quad (4)$$

where c, c', h and κ are constants ($c = 50$, $c' = 10$, $h = 0.5$, $\kappa = -1$ in the following experiments).

Since the polarity of u_i is important in nonmonotone neural networks, we consider $x_i = \mathrm{sgn}(u_i)$ and treat the vector $\boldsymbol{x} = (x_1, \ldots, x_n)$ as the network state, where $\mathrm{sgn}(u) = 1$ for $u > 0$ and -1 for $u \le 0$.

The network state $\boldsymbol{x}$ at any instant is represented by a point in the state space consisting of 2^n possible states. When $\boldsymbol{x}$ changes, it almost always moves to an adjacent point in the state space because x_i changes asynchronously. Consequently, $\boldsymbol{x}$ leaves a track with time, which we call the trajectory of $\boldsymbol{x}$. Similarly, we call $\boldsymbol{x}_{\rm pri} = (x_1, \ldots, x_k)$, $\boldsymbol{x}_{\rm mid} = (x_{k+1}, \ldots, x_l)$ and $\boldsymbol{x}_{\rm sup} = (x_{l+1}, \ldots, x_n)$ the states of the primary, middle and superior parts, respectively, and consider the trajectories of $\boldsymbol{x}_{\rm pri}$, $\boldsymbol{x}_{\rm mid}$ and $\boldsymbol{x}_{\rm sup}$ in the state space of each part.

4. Learning Algorithm

Let us consider m complex spatiotemporal patterns $\boldsymbol{c}^1(t), \ldots, \boldsymbol{c}^m(t)$ $(0 \le t \le T)$ and corresponding simple patterns $\boldsymbol{s}^1(t), \ldots, \boldsymbol{s}^m(t)$. We assume that $\boldsymbol{c}^\mu$ and $\boldsymbol{s}^\mu$ are k and $(n-l)$-dimensional binary vectors, respectively,

Table 1: Training schedule

cycle	λ_{pri}	λ_{mid}	λ_{sup}	cycle	λ_{pri}	λ_{mid}	λ_{sup}
1	0.20	0.20	0.20	11	0.20	0.10	0.10
2	0.20	0.20	0.20	12	0.10	0.10	0.20
3	0.20	0.18	0.18	13	0.20	0.08	0.08
4	0.18	0.18	0.20	14	0.08	0.08	0.20
5	0.20	0.16	0.16	15	0.20	0.06	0.06
6	0.16	0.16	0.20	16	0.06	0.06	0.20
7	0.20	0.14	0.14	17	0.20	0.04	0.04
8	0.14	0.14	0.20	18	0.04	0.04	0.20
9	0.20	0.12	0.12	19	0.20	0.00	0.00
10	0.12	0.12	0.20	20	0.00	0.00	0.20

whose elements c_i^μ and $\boldsymbol{s}_i^\mu$ are ± 1 and change asynchronously. Then, we can consider the trajectories of $\boldsymbol{c}^\mu$ and $\boldsymbol{s}^\mu$ in the pattern space regarded in the same light as the state spaces of the primary and superior part, respectively.

We train the association network using a learning signal vector $\boldsymbol{r} = (r_1, \ldots, r_n)$ with binary elements ($r_i = \pm 1$) so that $\boldsymbol{c}^\mu$ and $\boldsymbol{s}^\mu$ may be associated with each other. In our previous model [1,2], the learning signals $\boldsymbol{r}_{\text{pri}} = (r_1, \ldots, r_k)$ and $\boldsymbol{r}_{\text{sup}} = (r_{l+1}, \ldots, r_k)$ corresponding to the primary and superior parts were $\boldsymbol{c}^\mu(t)$ and $\boldsymbol{s}^\mu = \{O^\mu S^\mu\}_T$, respectively, where $\{O^\mu S^\mu\}_T$ denotes a spatiotemporal pattern which changes gradually from a static pattern O^μ to another pattern $S^\mu = (s_{l+1}^\mu, \ldots, s_n^\mu)$ in time T. In this case, however, the network cannot generate $\boldsymbol{c}^\mu(t)$ from $\boldsymbol{s}^\mu(t)$ unless we give the initial state $(\boldsymbol{c}^\mu(0), \boldsymbol{r}_{\text{mid}}(0))$ to the primary and middle parts. Accordingly, we change the starting point of $\boldsymbol{r}$ to $O = (-1, \ldots, -1)$ for all μ; that is, $(\boldsymbol{r}_{\text{pri}}, \boldsymbol{r}_{\text{sup}})$ moves from O via

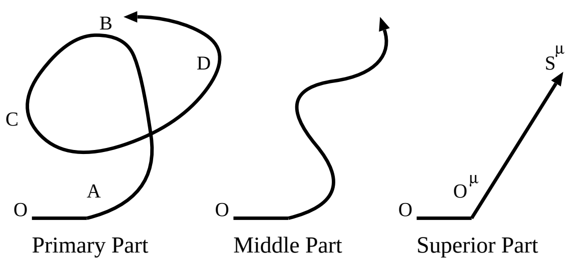

Fig. 3: Learned trajectory in each part.

Table 2: Temporal sequences for the experiment.

$\boldsymbol{c}^\mu$	$\boldsymbol{s}^\mu$	$\boldsymbol{c}^\mu$	$\boldsymbol{s}^\mu$
$\{ABECD\}_T$	$\{O^1S^1\}_T$	$\{DBAFB\}_T$	$\{O^7S^8\}_T$
$\{AEGFA\}_T$	$\{O^1S^2\}_T$	$\{EDGEC\}_T$	$\{O^9S^9\}_T$
$\{BACAF\}_T$	$\{O^3S^3\}_T$	$\{EGABD\}_T$	$\{O^9S^{10}\}_T$
$\{BEFGF\}_T$	$\{O^3S^4\}_T$	$\{FBDAE\}_T$	$\{O^{11}S^{11}\}_T$
$\{CEABD\}_T$	$\{O^5S^5\}_T$	$\{FGFCB\}_T$	$\{O^{11}S^{12}\}_T$
$\{CGBAB\}_T$	$\{O^5S^6\}_T$	$\{GDADE\}_T$	$\{O^{13}S^{13}\}_T$
$\{DAFGE\}_T$	$\{O^7S^7\}_T$	$\{GFAEF\}_T$	$\{O^{13}S^{14}\}_T$

$(\boldsymbol{c}^\mu(0), O^\mu)$ to $(\boldsymbol{c}^\mu(T), S^\mu)$ as schematically shown in Fig. 3.

The learning algorithm is as follows. First, we give an initial state $\boldsymbol{x} = O$ and input $\boldsymbol{r}$ in the form $z_i = \lambda_i r_i$ to the network while it acts according to Eq. (2). Here, λ_i denotes the input intensity of r_i and takes respective values λ_{pri}, λ_{mid} and λ_{sup} for the primary ($i \leq k$), middle ($k < i \leq l$) and superior ($i > l$) parts.

We simultaneously modify all synaptic weights w_{ij} according to

$$\tau' \frac{dw_{ij}}{dt} = -w_{ij} + \alpha r_i y_j, \qquad (5)$$

where τ' denotes a time constant of learning ($\tau' \gg \tau$) and α is a learning coefficient. Since learning performance is better when α is a decreasing function of $|u_i|$ [3], we set $\alpha = \alpha' x_i y_i$, α' being a positive constant. In the following experiments, $\alpha' = 2.0$ and $\tau' = 40000\tau$.

We apply this procedure for all μ, and repeat it over several cycles, gradually decreasing λ_{mid} and either λ_{pri} or λ_{sup}. An example of the training schedule is shown in Table 1. If $\boldsymbol{x}$ can move along $\boldsymbol{r}$ in both cases $\lambda_{\text{mid}} = \lambda_{\text{pri}} = 0$ and $\lambda_{\text{pri}} = \lambda_{\text{mid}} = 0$, then the learned trajectories are thought to have become attractors [3], and training is completed.

5. Computer Simulation

We carried out computer simulations with 400 input, 600 hidden and 400 output neu-

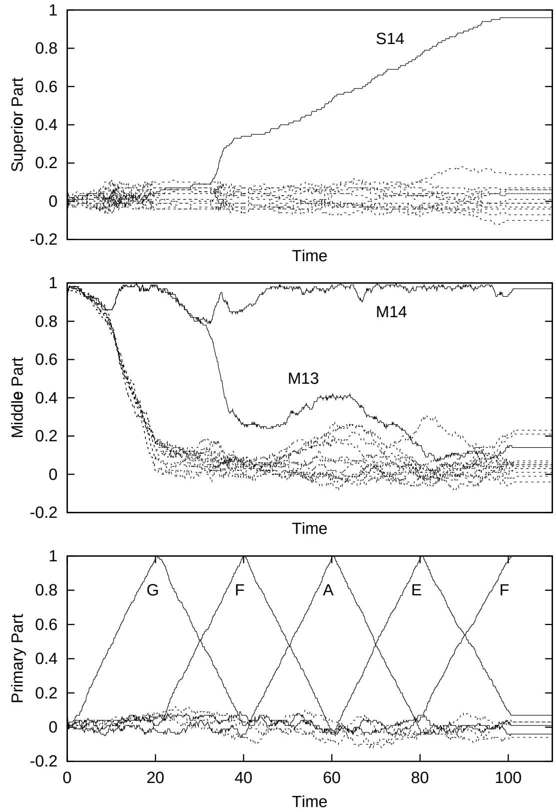

Fig. 4: A process of recognition.

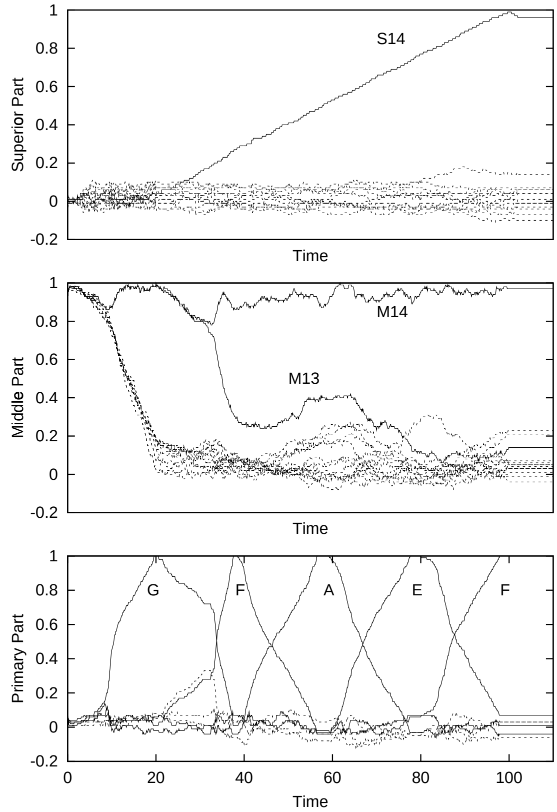

Fig. 5: A process of generation.

rons ($k = 400$, $l = 1000$, $n = 1400$).

We prepared 14 complex patterns as shown in Table 2. These patterns are formed by connecting seven static patterns A–G which are 400-dimensional binary vectors selected at random. Static patterns S^μ and O^μ ($\mu = 1, \ldots, 14$) are also selected at random, but if the head patterns $\boldsymbol{s}^{\mu 1}(0)$ and $\boldsymbol{s}^{\mu 2}(0)$ are identical, we set $O^{\mu 1} = O^{\mu 2}$. The temporal length $T = 80\tau$ for all $\boldsymbol{c}^\mu$ and $\boldsymbol{s}^\mu$.

After completing 20 cycles of learning, we input various patterns to the model and examined its behavior.

5.1 Recognition

Figure 4 shows a process of recognition when a learned sequence $\{OGFAEF\}_{100\tau}$ was given to the primary part with intensity $\lambda_{\rm pri} = 0.2$. To the middle and superior parts, we gave the initial states O at $t = 0$, but nothing ($z_i = 0$) thereafter.

Similarities (direction cosines) between $\boldsymbol{x}_{\rm sup}$ and S^μ denoted by $d_{\rm sup}(S^\mu)$ are plotted in the top graph, and those between $\boldsymbol{x}_{\rm pri}$ and A–G denoted by $d_{\rm pri}(A)$–$d_{\rm pri}(G)$ are plotted in the bottom one. The middle graph shows a change of $\boldsymbol{x}_{\rm mid}$ in a different manner, where similarities between $\boldsymbol{x}_{\rm mid}(t)$ and $\boldsymbol{r}^\mu_{\rm mid}(t)$ (denoting the μ-th learning signal) are plotted. The abscissa is time scaled by the time constant τ.

In the top graph, we see that $d_{\rm sup}(S^{14})$ increases consistently with time at $t > 30\tau$, and finally $\boldsymbol{x}_{\rm sup}$ almost reaches S^{14}. This indicates that the model has correctly recognized the input pattern as $\boldsymbol{c}^{14}$.

In the same manner, the other 13 sequences were correctly recognized. For the detailed process of recognition, see [1,2].

5.2 Generation

Figure 5 shows a process of generation when a sequence $\{OO^{13}S^{14}\}_{100\tau}$ was given to the superior part. To the primary and middle parts, we gave nothing but the initial states O.

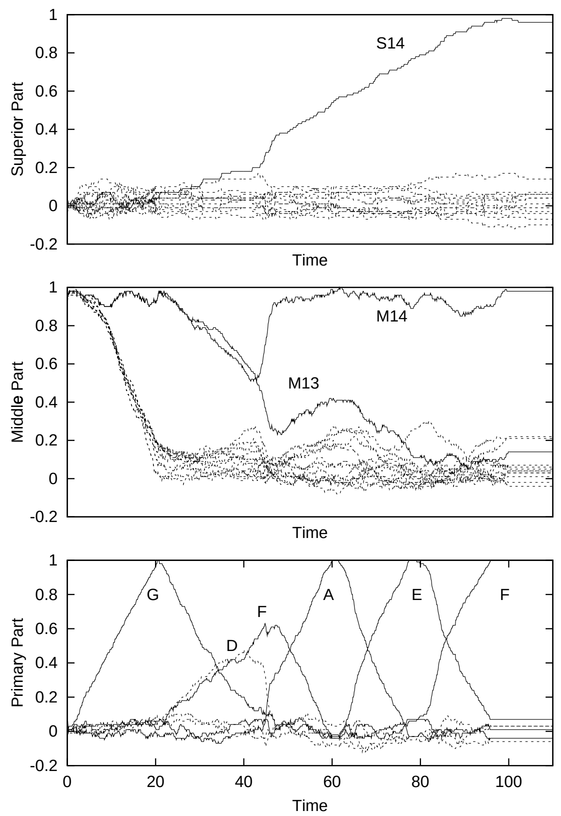

Fig. 6: Behavior when a vague pattern is inputted.

In the bottom graph, we see that $\boldsymbol{c}^{14} = \{GFAEF\}_T$ is reproduced. In this way, it was confirmed that the model can generate $\boldsymbol{c}^\mu$ from $\boldsymbol{s}^\mu$ for the other μ.

5.3 Bidirectional Interactions

Figure 6 shows another example, where equivocal sequences $\{O(O^9O^{13})(S^9S^{14})\}_{100\tau}$ and $\{OG(DF)A(DE)(EF)\}_{100\tau}$ were inputted simultaneously into the primary and superior parts, respectively. Here (DF) denotes a vector lying midway between D and F; (DE), (EF), (O^9O^{13}) and (S^9S^{14}) are also middle vectors so that each input spatiotemporal pattern lies in the middle of $\boldsymbol{c}^{13}$ and $\boldsymbol{c}^{14}$ or $\boldsymbol{s}^9$ and $\boldsymbol{s}^{14}$. Accordingly, either input alone cannot elicit a learned sequence from the network.

By giving the two inputs at the same time, however, $\boldsymbol{c}^{14}$ and $\boldsymbol{s}^{14}$ are reproduced in the primary and superior parts at $t > 45\tau$. This indicates that vague inputs to the primary and superior parts complement each other through bidirectional interactions.

6. Conclusion

We have described a nonmonotone neural network model that performs recognition and generation of spatiotemporal patterns based on a mutual conversion between simple and complex patterns.

This model is characterized by bidirectional processing in a single network, which enables active recognition, for example, determining a consistent sequence even if the input is incomplete.

Also, this model preserves the merits of the previous model [2] such as the simple architecture and learning algorithm and tolerance to noise in spatial and temporal dimensions. For these reasons, we think that this model shares some fundamental principles with the brain and is highly promising.

Theoretical analysis and application of the model are subjects for future study.

References

[1] M. Morita and S. Murakami, "Recognition of spatiotemporal patterns by nonmonotone neural networks", Proceedings of ICONIP'97, vol. 1, 6/9 (1997).

[2] S. Murakami, M. Morita and N. Sakamoto, "Recognition of spatiotemporal patterns using a nonmonotone neural network with hidden neurons", Proceedings of ICONIP'98, vol. 1, 287/290 (1998).

[3] M. Morita: "Memory and learning of sequential patterns by nonmonotone neural networks", Neural Networks, vol. 9, no. 8, 1477/1489 (1996).