ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 309-314

ORDER BETWEEN LOGIC NETWORKS AND STABLE NEURAL NETWORKS

Youichi Kobuchi

Department of Electronics and Informatics
Faculty of Science and Technology
Ryukoku University, Seta, 520-21 Japan
e-mail : kobuchi@rins. ryukoku. ac. jp

A logic network is a network of binary logic functions, which may be expressed as a higher-order neural network. We consider asynchronous operation mode and characterize stable networks through corresponding Lyapunov functions. We distinguish a special type of Lyapunov functions called *pure* and reveal their properties with relation to ordinary Lyapunov functions. We also introduce some order relations among logic functions and among logic networks, which play a crucial role in characterizing the existence condition of Lyapunov function in general for logic networks.

1. Introduction

Lyapunov function, also known as Energy function, is a function associated with a dynamical system whose function value decreases when the state change proceeds. Hopfield (1982) pointed out that there is such a function for any binary asynchronous neural network if the connecting weights are symmetric and if there are no self-loop connections. Several works which try to generalize his case followed. Braham and Hamblen (1988) showed that zero-diagonal condition of Hopfield can be replaced by non-negative diagonal condition. We extended symmetry weight condition to quasi-symmetry one (Kobuchi, 1991 Kobuchi and Kawai, 1991). Goles (1987) showed Lyapunov functions for synchronous neural networks. Baldi (1988) treated higher-order networks and introduced nonquadratic Energy functions. Lyapunov functions for continuous variable neural networks were shown by Schürmann (1989), to name a few.

We here treat logic function networks regarding them as higher-order neural networks which have Lyapunov functions. Since the existence of Lyapunov functions for such networks means that there are no cyclic states whose periods are greater than one, any initial state configuration ultimately approaches certain stable state.

We investigate how and when a given logic network or a higher-order neural network has a Lyapunov function. In so doing , we introduce the concept of pure Lyapunov function, and relate it with an injective state function. We also consider orderly dependent logic functions which are defined here to cope with arbitrary logic networks with Lyapunov functions.

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 309-314

2. Lyapunov Functions for Logic Networks

Consider n logic elements designated by c_i's where i = 1, 2, ..., n. Each element c_i has a state s_i which takes one of the two logic values 0 or 1. The next state function of an element c_i is defined by a logic function $f_i(s_1,s_2,...,s_n) : \{0,1\}^n \to \{0,1\}$. The set $\{f_i(s_1,s_2,...,s_n) \mid i = 1, 2, ..., n\}$ then defines a network architecture of logic functions or a *logic network* and we denote it as $(n; f_1,f_2,...,f_n)$. If an element is chosen arbitrarily out of the n elements, and if it undergoes possible state change through the function, we are defining an *asynchronous* operation mode. The global state transition function of $(n; f_1,f_2,...,f_n)$ under this mode is denoted as $\mathcal{F} : \{0,1\}^n \to 2^{\{0,1\}^n}$. In this note, we consider only asynchronous logic networks. A network under consideration is denoted as $(n; f_1,f_2,...,f_n; \mathcal{F})$.

2. 1. Stable Logic Networks

For a given logic network $M = (n; f_1,f_2,...,f_n; \mathcal{F})$, we define a state transition relation $\Rightarrow_M$ or simply $\Rightarrow$ (when M is understood) over the set of states $S = \{0,1\}^n$ as follows.

For any **a** and **b** in S, $\mathbf{a} \Rightarrow \mathbf{b}$ if and only if $\mathbf{b} \in \mathcal{F}(\mathbf{a})$.

Since we are considering only asynchronous operations, $\mathbf{a} \Rightarrow \mathbf{b}$ implies that the Hamming distance of **a** and **b** is less than or equal to 1. Let $\overset{*}{\Rightarrow}$ denote the reflexive transitive closure of $\Rightarrow$.

For a state **a** in S, if $\mathbf{a} \Rightarrow \mathbf{b}$ implies $\mathbf{a} = \mathbf{b}$, then **a** is called a *stable state*. We here define a stable logic network as follows.

Definition 2. 1.

Consider a logic network $M = (n; f_1,f_2,...,f_n; \mathcal{F})$. For any **a** and **b** in $S = \{0,1\}^n$ such that $\mathbf{a} \neq \mathbf{b}$, assume that $\mathbf{a} \Rightarrow_M \mathbf{b}$. The state **a** is called *acyclic* if $\mathbf{b} \overset{*}{\Rightarrow}_M \mathbf{c}$ implies $\mathbf{c} \neq \mathbf{a}$. A logic network M is called *stable* if every **a** in S is acyclic.

Note that a stable state is acyclic because the assumption part is vacuous in the above definition.

Let $\mathbf{a} = (a_1, a_2, ..., a_n)$ be a state in S. If $a_i = 0$, we write such a state as $\mathbf{a}_{i(0)}$, and if $a_i = 1$, as $\mathbf{a}_{i(1)}$. That is, $\mathbf{a}_{i(0)} = (a_1, a_2, ..., a_{i-1}, 0, a_{i+1}, ..., a_n) \in S$ and $\mathbf{a}_{i(1)} = (a_1, a_2, ..., a_{i-1}, 1, a_{i+1}, ..., a_n) \in S$.

Property 2. 2.

In a stable logic network M, one of the following three cases occurs where i is an arbitrarily chosen site of state transition.

(1) $\mathbf{a}_{i(0)} \Rightarrow \mathbf{a}_{i(1)} \Rightarrow \mathbf{a}_{i(1)}$;

(2) $\mathbf{a}_{i(1)} \Rightarrow \mathbf{a}_{i(0)} \Rightarrow \mathbf{a}_{i(0)}$;

(3) $\mathbf{a}_{i(0)} \not\Rightarrow \mathbf{a}_{i(1)}$ and $\mathbf{a}_{i(1)} \not\Rightarrow \mathbf{a}_{i(0)}$.

2. 2. Lyapunov Functions

Now we define a Lyapunov function for a logic network, which is another basic concept.

Definition 2. 3.

A state function E(s) : $\{0,1\}^n \to \mathbf{R}$ is said to be a *Lyapunov function* for a logic network M = (n; $f_1,f_2,...,f_n$; $\mathcal{F}$) if for any **a** and **b** in $\{0,1\}^n$ such that **a** ≠ **b**, **a** $\Rightarrow_M$ **b** implies E(**a**) > E(**b**). M is said to have a Lyapunov function E(s) if it is a Lyapunov function for M.

Then, we have the following simple but basic result.

Theorem 2. 4.

For a logic network M, the following propositions are equivalent.

(1) M is stable.

(2) M has an injective Lyapunov function.

(3) M has a Lyapunov function.

Proof.

(1) → (2) : Let M = (n; $f_1,f_2,...,f_n$; $\mathcal{F}$) be a stable logic network. Consider a directed graph G_M = (V, E) where the set of vertices V = $\{0,1\}^n$, and the set of edges E ⊆ V × V is given as E = {(**a**, **b**) | **a** $\Rightarrow_M$ **b**}. Since M is stable, there are no directed cycles of length more than 1 in the graph G_M . Then, there is at least one vertex **a** such that for no other vertices **b**, (**b**, **a**) ∈ E. Define a function φ_M: $\{0,1\}^n \to \{1, 2, ..., 2^n\}$ as follows. First, let φ_M (**a**) = 2^n, and delete the vertex **a** together with all the edges leaving it from G_M to obtain a graph G_M' with (2^n - 1) vertices. This new graph does not have directed cycles of length more than 1, either. Then, we can continue the process until we have a completely defined injective function φ_M : $\{0,1\}^n \to \{1, 2, ..., 2^n\}$. Thus, in general, a vertex **d** is deleted at i-th step and given a number 2^n - i +1. By definition, for any distinct **a** and **b** in $\{0,1\}^n$, **a** $\Rightarrow_M$ **b** implies φ_M (**a**) > φ_M (**b**). That is, φ_M is a Lyapunov function for M.

(2) → (3) : Self-evident.

(3) → (1) : Let M = (n; $f_1,f_2,...,f_n$; $\mathcal{F}$) has a Lyapunov function E(s). For any **a** and **b** in $\{0,1\}^n$ such that **a** ≠ **b**, **a** ⇒ **b** implies E(**a**) > E(**b**). If **b** $\overset{*}{\Rightarrow}$ **c** holds for some **c** in $\{0,1\}^n$, this means **c** ≠ **a** because the relation E(**b**) ≥ E(**c**) yields E(**c**) ≠ E(**a**). So, **a** is acyclic for every **a** in $\{0,1\}^n$, and M is stable.

3. Pure Lyapunov Functions

In this section, we consider more restrictive definition of Lyapunov functions than the one in the previous section. That is, we regard a state function as directly guiding the transition behavior. A similar concept called *strict* has been defined in (Kobuchi, 1994).

Definition 3. 1.

Let E(s) be an injective state function such that E(s) : $\{0,1\}^n \to \mathbf{R}$ where **s** = ($s_1,s_2,...,s_n$) is in $\{0,1\}^n$. It is a *pure Lyapunov function* for a logic net M = (n; $f_1,f_2,...,f_n$; $\mathcal{F}$) if the following holds :

For any **a** and **b** in $\{0,1\}^n$ such that whose Hamming distance equals one, we have

$$(\mathbf{a} \Rightarrow_M \mathbf{b}) \Leftrightarrow E(\mathbf{a}) > E(\mathbf{b}).$$

Any injective state function can be a pure Lyapunov function for some logic network, which was first pointed out by Baldi(1988).

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 309-314

Theorem 3. 2.

Let $E(s) : \{0,1\}^n \rightarrow \mathbf{R}$ be any injective state function. Then it is a pure Lyapunov function for the logic network $M = (n; f_1, f_2, \ldots, f_n; \mathcal{F})$ where

$$f_i(s) = H(-\partial E(s)/\partial s_i) \text{ for } i = 1, 2, \ldots, n$$

such that $\partial E(s)/\partial s_i \equiv E(s \mid s_i = 1) - E(s \mid s_i = 0)$ and $H(\cdot)$ is the Heaviside function.

Note that the above $f_i(s)$ is independent of s_i. We call the logic network M in Theorem 3. 2. as *independent*. If a logic network has a Lyapunov function, then it has an injective Lyapunov function by Theorem 2. 4. On the other hand, any injective state function is a pure Lyapunov function for an independent logic network by Theorem 3. 2. The relations between these logic networks and corresponding Lyapunov functions will be made clear in the following section.

4. Order Relations Between Logic Networks and Lyapunov Functions

We define two kinds of partial order relations over the set of logic networks and show their equivalence. We also show that these relations relate pure and ordinary Lyapunov functions.

4. 1. Global Relation of Logic Networks

Definition 4. 1.

Let $M = (n; f_1, f_2, \ldots, f_n; \mathcal{F})$ and $L = (n; g_1, g_2, \ldots, g_n; \mathcal{G})$ be two logic networks and let $S = \{0, 1\}^n$. We define a relation $\prec$ on the set of logic networks as follows.

$L \prec M \Leftrightarrow$ For any **a** and **b** in S such that $\mathbf{a} \neq \mathbf{b}$, $(\mathbf{a} \Rightarrow_L \mathbf{b})$ implies $(\mathbf{a} \Rightarrow_M \mathbf{b})$.

Lemma 4. 2.

Let L and M be logic networks. If $L \prec M$ and M is stable, then so is L. Furthermore, L and M can have a same Lyapunov function in this case.

Proof.

For any **a** in S, consider another **b** in S such that $\mathbf{a} \Rightarrow_L \mathbf{b}$. Also consider arbitrary **c** in S such that $\mathbf{b} \overset{*}{\Rightarrow}_L \mathbf{c}$. Since $L \prec M$, we have $\mathbf{a} \Rightarrow_M \mathbf{b}$ and $\mathbf{b} \overset{*}{\Rightarrow}_M \mathbf{c}$. As M is stable, **a** is an acyclic state of M, which means $\mathbf{c} \neq \mathbf{a}$. Thus **a** is also acyclic for L and L is stable. As to Lyapunov functions, there is one such function E(s) for M because M is stable (by Theorem 2.4.). It is easy to see that this function E(s) is also a Lyapunov function for L.

Lemma 4. 3.

Let L be a stable logic network. Then, there is an independent logic network M such that $L \prec M$ and M has a pure Lyapunov function.

Proof.

By Theorem 2. 4., $L = (n; g_1, g_2, \ldots, g_n; \mathcal{G})$ has an injective Lyapunov function E(s). By Theorem 3. 2., this E(s) is a pure Lyapunov function for the logic network $M = (n; f_1, f_2, \ldots, f_n; \mathcal{F})$ where $f_i(s) = H(-\partial E(s)/\partial s_i)$ for $i = 1, 2, \ldots, n$. We have to show that $L \prec M$. For any distinct **a** and **b** in S, $\mathbf{a} \Rightarrow_L \mathbf{b}$ implies $E(\mathbf{a}) > E(\mathbf{b})$ since E(s) is a Lyapunov function for L. Because E(s) is a pure Lyapunov function for M, $E(\mathbf{a}) > E(\mathbf{b})$ implies $\mathbf{a} \Rightarrow_M \mathbf{b}$. Thus we have $L \prec M$.

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 309-314

4. 2. Orderly Dependency

Now, another relation on the set of logic networks is introduced using an order relation on the set of logic functions.

Definition 4. 4.

Let f(s) and g(s) be any n-variable logic functions where n is an arbitrary positive integer. A relation $\leq$ is defined as

$$f \leq g \Leftrightarrow f(s) = 1 \text{ implies } g(s) = 1.$$

We denote the corresponding strict order relation as $f < g$ if $f \leq g$ and $f \neq g$.

An n-variable logic function f(s) such that $s = (s_1, s_2, ..., s_n)$ can be expanded by a variable s_i as $f(s) = p\, s_i \vee q\, \overline{s_i}$ where $p = f(s \mid s_i = 1)$ and $q = f(s \mid s_i = 0)$ are (n - 1)-variable logic functions.

Definition 4. 5.

Let $f(s) = p\, s_i \vee q\, \overline{s_i}$ be an expansion of a logic function f(s) by a variable s_i. It is *independent* of s_i (or s_i -independent) if $p = q$. It is called *orderly dependent* on s_i if $q < p$.

Definition 4. 6.

Let $f(s) = p\, s_i \vee q\, \overline{s_i}$ and $g(s) = u\, s_i \vee v\, \overline{s_i}$ be logic functions of n-variable $s = (s_1, s_2, ..., s_n)$ expanded by a variable s_i. Then, a partial order relation $\leq_i$ is defined as

$$g(s) \leq_i f(s) \Leftrightarrow p \leq u \text{ and } v \leq q.$$

We write $g(s) <_i f(s)$ when $g(s) \leq_i f(s)$ and $g(s) \neq f(s)$.

Property 4. 7.

If $g(s) <_i f(s)$ and f(s) is s_i -independent, then g(s) is orderly dependent on s_i.

Proof.

By assumption, if f(s) is expanded by s_i as $p\, s_i \vee q\, \overline{s_i}$, then $p = q$. If $g(s) = u\, s_i \vee v\, \overline{s_i}$, then $v \leq q = p \leq u$, which implies $v < u$ as desired since $p \neq u$ or $v \neq q$.

Definition 4. 8.

Let $M = (n; f_1, f_2, ..., f_n; \mathcal{F})$ and $L = (n; g_1, g_2, ..., g_n; \mathcal{G})$ be two logic networks. We define a relation $\leq$ on the set of logic networks as follows.

$$L \leq M \Leftrightarrow g_i(s) \leq_i f_i(s) \text{ for } i = 1,2,..., n.$$

Theorem 4. 9.

Let $M = (n; f_1, f_2, ..., f_n; \mathcal{F})$ and $L = (n; g_1, g_2, ..., g_n; \mathcal{G})$ be two logic networks. Then, $L \prec M \Leftrightarrow L \leq M$.

Proof. (Omitted.)

Summing up the hitherto obtained results, we have

Theorem 4. 10.

1) A logic network $L = (n; g_1, g_2, ..., g_n; \mathcal{G})$ is stable if and only if there exists an independent higher order neural network $M = (n; f_1, f_2, ..., f_n; \mathcal{F})$ such that $L \leq M$.

2) In such case, M has a pure Lyapunov function E(s) and f_i can be represented as $H(-\partial E(s)/\partial s_i)$ for $i = 1, 2, ..., n$.

3) L also has the same Lyapunov function E(s).

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 309-314

5. Concluding Remarks

We considered stable asynchronous logic networks and clarified when and how they have Lyapunov functions. The first point we noted is that logic networks can be expressed as higher order neural networks. We nevertheless retained logic function expression and tried to connect the two ways of representation.

In the logic function expression, we defined orderly dependency concept. For a logic network $M = (n; f_1, f_2, \ldots, f_n; \mathcal{F})$ to be stable, each function f_i should be s_i-independent or orderly dependent.

We also defined two partial order relations on the set of logic networks and showed their equivalence. The one is related with global state transitions of the networks, and the other is defined based on the orderly dependency of constituent logic functions. A higher order neural network with pure Lyapunov function is a maximal element under these same order relation.

These generalized analyses will widen the applicability of logic net in learning, associative memory, and combinatorial optimization problems as has been done for ordinary neural networks.

Acknowledgments

The author wishes to thank Prof. T. Mori of Kyoto Institute of Technology for his advice. He is also indebted to Mr. Sakamoto and Mr. Tujii for their comments and discussions during the various phases of this work.

References

Baldi (1988) Neural Networks, Orientations of the Hyper cube, and Algebraic Threshold Functions. *IEEE Trans. on IT* Vol. 34, No.3. 523-530.

Braham, R. and Hamblem, J. O. (1988) On the behavior of some associative neural networks. *Biological Cybernetics*, **60**, 145-151.

Goles, E. (1987) Lyapunov Functions Associated to Automata Networks in *Automata Networks in Computer Science* (Eds.Fogelman, F., Robert, Y., and Tchuente, M.) Manchester University Press.58-81.

Hopfield, J.J. (1982) Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proc. NAS. USA* **79**, 2554-2558.

Kobuchi, Y. (1991) State Evaluation Functions and Lyapunov Functions for Neural Networks. *Neural Networks*, Vol. 4, No. 4, 505-510.

Kobuchi, Y. (1994) Strict Lyapunov Functions for Higher-order Neural Networks. in "Words, Languages and Combinatorics II" (Editors M. Ito and H. Jürgensen) 252-260 World Scientific Publishing Co.

Kobuchi, Y. and H. Kawai (1991) Quasi-Symmetric Logic Networks Have Lyapunov Functions. Proceedings of IJCNN Vol. 2, 1379-1384. IEEE Service Centre

Schürmann, B. (1989) Stability and adaptation in artificial neural systems. Physical Review A Vol. 40, No.5, 2681-2688.