THRESHOLD TRANSFORMATIONS AND DYNAMICAL SYSTEMS OF NEURAL NETWORKS

A COMBINATORIAL APPROACH

4th Edition (V 4.0)

Takao UEDA

Preface

Neural networks have been a hot field in recent years. One of the reasons is that they are now used for practical decision making through computer software and hardware. However, unlike widely-used methods such as statistical decision and linear programming, validity of applying neural networks is uncertain, since limitations in application, methods of evaluating errors, and the like are not established.

Information processing on a neural network is mathematically summarized as *threshold transformations*. Because of its non-linearity, threshold transformations are hard to analyze. However, clarifying mathematical properties of threshold transformations is essential for valid applications.

Various models of neural networks have been proposed. However, according to Amit, they fall basically into two types, *feed forward networks* and *attractor neural networks*. In essence, a simple network of the first type is a threshold transformation from the state space $X = \{-1, 1\}^n$ into the state space $Y = \{-1, 1\}^m$, and a simple network of the second type is a threshold transformation into $X = \{-1, 1\}^n$ to itself, that is, a *threshold transformation* of X. The first type is mainly concerned with the function itself regarding the state spaces X and Y as different. A network of the second type is concerned with the *dynamical system* generated by the transformation. Kamp and Hasler's book, which contains most prior main results on the second type, calls it a *recursive neural network*. I call it a *dynamical neural network* (DNN). The DNNs more closely approximate central nervous systems, and mathematically more interesting. But, what is an attractor in a neural network? Strangely enough, you can not find its definition in a book of Amit or Kamp and Hasler. Amari and others defined some kinds of attractors called *stability*, but they are too limited to be established as standards. Therefore, I had to start with the groundwork of defining basic concepts in the finite-state dynamical system (Chapter 6.1).

I first came into contact with the classical neural network model of McCulloch and Pitts in the early 1970s, but did not find any mathematically significant results, so that I was not much interested in that. Then in the mid 70s, I was informed of Arimoto's theorem (published in 1963) by M. Sato at the applied mathematics section of a semiannual meeting of the Mathematical Society of Japan. When I visited Arimoto at his office in Osaka University to get a copy of his paper containing the result, he expressed a negative opinion about the neural network model because of its excessive simplification of reality and rather discouraged me to go into that field. But I had different criteria, and soon later, got the concept of the variation of a Boolean transformation and *minimal Boolean transformations*. The *variation* of a transformation of $\{-1, 1\}^n$ is the total number of coordinates of $\{-1, 1\}^n$ changed by the transformation. A transformation F is *minimal* if the variation of F is equal or less than the variation of any transformation G such that $G = TF$ for some orthogonal transformation T. In particular, the only minimal orthogonal transformation is the identity I. However, I could not utilize the concepts for the study of threshold transformations at that time.

Without much progress, I interrupted the study of mathematical science, partly because I was disappointed with my ability in mathematics and partly because I did not expect I could ever touch reality through mathematical science. I left Japan in 1981 to come to the United States, and studied psychology, philosophy, and literature in most of the 80s.

By the time I restarted the study of threshold transformations in the late 80s, my notion of reality had changed. The environment concerning neural networks had also changed owing to the spreading of their practical application to signal processing and various decision making and also owing to the much publicized work of Hopfield. Further, I could easily simulate a model and implement an algorithm, since I had owned a personal computer for word processing. Soon, I succeeded in obtaining several simplest one-to-one threshold transformations for the purpose of extracting their non-linear properties, by introducing [ ]-representations of self-dual transformations and focusing on the class of minimal circular transformations (Chapter 4.4). The [ ]-representations are effectively used throughout this seminar note.

The concepts of minimal and circular Boolean transformations also led me to attack and solve an outstanding hard problem of constructing a kind of Gray code for necklaces. The results are both theoretically and practically significant in the area of combinatorial Gray code, but outside the scope of this seminar note.

For years I had been unable to utilize my results on one-to- one threshold transformations for the study of neural networks. Then just after the New Year's Day of 1995, the idea of incorporating *spontaneous firing* suddenly came to my mind with its relation to maximal or minimal threshold transformations, when I was browsing in Amit's book, Modeling Brain Function, at a book store. For me, it was l'oeuf de Colomb (Columbus's egg). Amit's book was defective, because it did not distinguish attractors from limit cycles, but it discussed maximum, minimum, and average firing rates. And I recalled spontaneous firing, which I had almost forgotten but had learned in an undergraduate course at the University of California at Santa Cruz that used the first edition of Kalat's text book, *Biological Psychology.*

In my primitive DNNs, the prototype DNN, in which each neuron is disconnected from each other and performs spontaneous firing at rate $1/2$, is represented by a transformation $-I$, which is the *maximal* orthogonal transformation. Thus the concept of minimal threshold transformations is related to a primitive DNN model that incorporates spontaneous firing. In fact, if F is a minimal threshold transformation, then $-F$ generates a primitive DNN, and an attractor of F is easily converted to an attractor of $-F$.

However, my earlier and primary objective was to clarify non- linear properties of threshold transformations. That is why I concentrated on one-to-one threshold transformations in the first place. The readers will find that a main difference between a linear one-to-one transformation, that is, an $orthogonal transformation$, of $\{-1,1\}^n$ and a non-linear (i.e. non-orthogonal) one-to-one threshold transformation of $\{-1,1\}^n$ is the selectiveness in the latter. For example, in a conventional digital computer, its CPU performs, in one machine cycle, parallel processing of a basic unary operation such as one-bit right rotation or complementation of any data word, which is a member of the set of n-bit binary strings $\mathbf{Q}^n = \{0,1\}^n$. These operations are *isometries*, which are equivalent to orthogonal transformations of $\{-1,1\}^n$. On the other hand, a CPU capable of non-linear threshold transformations can perform, in one machine cycle, a selective rotation or complementation in that, if n is even, a data word is rotated only when one arbitrarily predetermined

n-string or one of those obtained by rotating it is loaded into the register. Similarly, a data word is complemented only when it is one arbitrarily predetermined n-string.

Studying one-to-one threshold transformations is also good preparation for neural networks, since not only common methods can be applied to the two subjects, but also a DNN having an attractor is often constructed by modifying a one-to-one threshold transformation. In fact, an enhanced Arimoto theorem in Chapter 5.5, and most attractors in Chapter 6 were obtained in this way.

Meanwhile, in the process of seeking for one-to-one threshold transformations, I sometimes found it harder to prove a given transformation to be one-to-one than to be a threshold transformation. Then I discovered that most one-to-one minimal threshold transformations are *reflective*. Further, proof that a complex threshold transformation is one-to-one is comparatively simplified by proving it to be reflective. By a reflective transformation I mean a one-to-one transformation such that its inverse is *orthogonally similar* to itself. This kind of transformation had already appeared as a binary-reflected Gray code. It also includes isometries of $\mathbf{Q}^n$, as I proved in Chapter 3.2. At present, I have not found any one-to- one threshold transformation that is incompressible, minimal, and non-reflective. By compression, some of the minimal threshold transformations can be further reduced to simpler ones (Chapter 5.2).

The readers may be somewhat confused by the use of both $\{-1,1\}^n$ and $\mathbf{Q}^n$ throughout the seminar note for the domain of threshold transformations. However, this use is by no means accidental. Threshold transformations are connected with two different mathematical structures: the real n-space $\mathbf{R}^n$ and the Boolean algebra. Such concepts as *linear separability* and orthogonal transformations belong to the first. The use of $\{-1,1\}^n$ is comfortable in this context. On the other hand, any Boolean transformations $\mathbf{Q}^n$ can be expressed in compact form using the *Boolean operations* $\vee$ (OR) and $\cdot$ (AND) in the *Boolean algebra* $\mathbf{Q}$, so that the use of $\mathbf{Q}^n$ allows us to treat threshold transformations as a special class of general Boolean transformations. Moreover, the Boolean operations $\vee$ and $\cdot$ are basic non-linear threshold functions from $\mathbf{Q}^2$ to $\mathbf{Q}$, so that threshold transformations can be effectively dealt with in terms of Boolean operations. Therefore, we use both $\{-1,1\}^n$ and $\mathbf{Q}^n$ depending on demands.

The shortcoming that the firing rate of any neuron can not exceed 2 times the spontaneous firing rate in the primitive DNN model on the state space $X = \{-1,1\}^n$ described in Chapter 6 is due to a greater problem that any state $x(t+1)$ depends only on $x(t)$ for a given *efficacy matrix* E and time t. Therefore the *postsynaptic potential* $(Ex(t))_i$ is *spatial summation.* However, the postsynaptic potential should also include *temporal summation* to simulate real nervous systems. Therefore, a primitive DNN model with temporal summation and spontaneous firing rate 1/3 is described in Chapter 7. That DNN is no longer a dynamical system on $X = \{-1,1\}^n$, and generated by a threshold function from $X \times X$ to X. The results are a variety of more interesting attractors, although exact analysis becomes more tedious.

The fundamental limitation of the DNNs described in Chapters 6 and 7 is that they are autonomous, that is, the stable periodic firing patterns that are represented by attractors are completely determined by the efficacy matrices of synaptic connections and the initial states. However, the dynamics of any biological system depends on information that changes at every unit time and that is input from the outside of the system, from neurons of other nervous systems and/or from external stimulus. In autonomous models, if a minimal attractor consists of more than one cycle, then there are some fluidity of shifting from one pattern to another caused by noise, even with a change in firing rate in some cases. Therefore, my next step was to formulate non-autonomous DNNs, expand the concepts of attractors, and prove their existence.

New concepts are necessary with non-autonomous DNNs. One is bi-dependence, and another is invariance with the timing of input. The bi-dependence means that asymptotic properties of a DNN are dictated neither by the initial state nor the input, but dependent on both of them. The invariance with the timing of input means that assuming a periodic input sequence, asymptotic properties are independent of any shift in arrival of the input sequence. In my definition, attractors in non-autonomous DNNs must satisfy this condition. In the final Chapter 8, we will see how autonomous DNNs described in Chapter 6 are modified by input, so that a non-attractive limit cycle becomes attractive, a non-unique attractor becomes unique, and an attractor consisting of more than one cycle becomes an attractive cycle, but still the convergence to the attractor depends on the initial state.

It seems that when McCulloch and Pitts originally created a neural network model, they did not intend to use it for practical decision making but for the analysis of real neural activities. My motives for studying neural networks were first mathematical and then biological. The incorporation of spontaneous firing in my primitive DNN model, a feature distinct from prior models, served my these two motives. However, in the immediate future, the present results may be used in engineering rather than in biology, such as for a new computer architecture. The main reason, I think, is not that the DNN is "unrealistic", but that currently available experimental data are either too microcosmic or too macrocosmic for biological applications of our results.

A summary of each chapter is described in the *abstract* at the beginning of the chapter, so that readers who want to know main results may first refer to the *abstracts*. In addition to basic concepts and notations in the field of discrete mathematics, most of which are defined in Chapter 1, I had to introduce various new definitions and notations. Symbols and Notations attached in the final pages will serve the readers for easy reference.

Takao Ueda

New York
—Written for the first edition, October 1999

I have updated Chapter 6 with newly found attractors, since I compiled the first edition for the first time. The volume has grown too much to be contained in one chapter. Also, writing each time the conversion from F to $-F$ with the attractor wastes pages and makes less readable, so that, in this second edition, I have housed attractors of minimal transformations without the conversion in a separate Chapter 7.

—Added for the second edition, February 2001

I rewrote Chapter 7, 8, 9 on the common ground of extensive representations of Boolean transformations. Consequently, as Boolean [ ]-representations are basic tools for the first half part, the integer-valued extensive representations are now basic tools for analysis of attraction in the second half part. I also added Chapter 10 to illustrate tangible applications.

—Added for the third edition, August 2002

I revised Chapters 3.4, 4.4, 4.5, 5.2, and 7.1-7.5 to incorporate a class of one-to-one threshold transformations having single cycles and derived transformations having attractors. Particularly I reorganized Chapter 7 with the new results.

—Added for the fourth edition, May 2005.

Contents