1 Introduction

Scenario

Recent history of the electronic and computer industry can be viewed as three waves of revolutionary processes.[1] The first revolution, making cheap computing power available via microprocessors in the 1970s, led to the PC industry of the 1980s. The cheap laser and fiber optics, which resulted in cheap bandwidth at the end of the 1980s, led to the Internet industry of the 1990s. The third wave, the sensor revolution at the end of the 1990s, will also provide for a new industry. Sensor revolution means that cheap sensor and MEMS (micro-electro-mechanical system) arrays are proliferating in almost all the conceivable forms. Artificial eyes, nose, ears, taste, and somatosensory devices as well as sensing all physical, chemical, and biological parameters, together with microactuators, etc. are becoming commodities. Thousands and millions of generically analog signals are produced waiting for processing. A new computing paradigm is needed. The cited technology assessment[1] reads:

> The long-term consequence of the coming sensor revolution may be the emergence of a newer analog computing industry in which digital technology plays a mere supporting role, or in some instances plays no role at all.

For processing analog array signals, the revolutionary Analogic Cellular Computer paradigm is a major candidate. The core of this computer is a Cellular Nonlinear/neural network[2] (CNN), an array of analog dynamic processors or cells. The computer architecture is the CNN Universal Machine,[3] with its various physical implementations. At the same time, Analogic CNN computers mimic the anatomy and physiology of many sensory and processing organs with an additional capability of stored programmability. Recent studies on optical and nano-scale implementations open up new horizons on the atomic and molecular levels.

The CNN was invented by Leon O. Chua and Lin Yang in Berkeley in 1988. Unlike cellular automata, CNN host processors accepting and generating analog signals, the time is continuous, and the interaction values are also real values. Unlike lattice dynamics, the input of the CNN array plays an important role. Moreover, CNN becomes a rigorous framework for complex systems exhibiting emergent behavior and the various forms of emergent computations. The notion of the cloning template, the representation of the local interconnection pattern, is crucial. This allows not only modeling but also engineering of complex systems.

Stored programmability, invented by John von Neumann, was the key for endowing digital computers with an almost limitless capability within the digital universe of signals, *opening the door to human invention* via digital algorithms and software. Indeed, according to the Turing–Church thesis, any algorithms on integers conceived by humans can be represented by Recursive functions/Turing Machines/Grammars. The *CNN Universal Machine is universal* not only in a Turing sense but also on analog array signals. Due to stored programmability, it is *also open to human intelligence* with a practically limitless capability within the universe of analog array signals, via analogic spatio-temporal algorithms and software.

The new world opened by the Analogic CNN computing paradigm is nowadays a reality. There are operational focal plane visual microprocessors with 4096 or 16 000 processors, which are fully stored, programmable, and there are Walkman-size self-contained units with image supercomputer speed.

The CNN Universal Chip[4] highlighted on the cover of this book represents a milestone in information technology because it is the first operational, fully programmable industrial-size brain-like stored-program dynamic array computer in the world. This complete computer on a chip consists of an array of 64×64 0.5 micron CMOS cell processors, where each cell is endowed not only with a photo sensor for direct optical input of images and videos, but also with communication and control circuitries, as well as local analog and logic memories. Each CNN cell is interfaced with its nearest neighbors, as well as with the outside world. This massively parallel focal-plane array computer is capable of processing 3 trillion equivalent digital operations per second (in analog mode), a performance which can be matched only by supercomputers. In terms of the *SPA* (*speed*, *power*, *area*) measures, this CNN universal chip is far superior to any equivalent DSP implementation by at least three orders of magnitude in either *speed*, *power*, or *area*. In fact, by exploiting the state-of-the-art vertical packaging technologies, close to *peta* (10^{15}) OPS CNN universal cube can be fabricated with such universal chips, using 200×200 arrays.

There are many applications which call for TeraOPS or even PetaOPS in a Walkman-size device. Some of these applications include high-speed target recognition and tracking, real-time visual inspection of manufacturing processes, intelligent vision capable of recognizing context sensitive and moving scenes, as well as applications requiring real-time fusing of multiple modalities, such as multispectral images involving visible, infrared, long wave infrared, and polarized lights.

In addition to the immense image and video processing power of the CNN universal chip, we can exploit its unique brain-like architecture to implement brain-like information processing tasks which conventional digital computers have found wanting. Such brain-like processing operations will necessarily be *non-numeric* and *spatio-temporal* in nature, and will require no more than the accuracy of common neurons, which is less than eight bits. Since the computation is a non-iterative wave-like process, the input–output accuracy is not constrained by the iterative digital process. The CNN universal chip is therefore an ideal tool for developing and implementing brain-like information processing schemes. It is this vision of brain-like computing via the CNN universal chip that makes the publication of this textbook both a timely and historic event, the first undergraduate textbook on this new computing paradigm.

The textbook

Cellular Nonlinear/neural Networks (CNN) is an invention with rapid proliferation. After the publication of the cited original paper by Chua and Yang in 1988, several papers explored the rich dynamics inherent in this simple architecture. Indeed, many artificial, physical, chemical, as well as living (biological) systems and organs can be very conveniently modeled via CNN. Hence, the book is written in such a way that no electronic circuit knowledge is needed to understand the first 14 chapters of this book. Indeed, it is our teaching experience, at Berkeley and in Budapest, that undergraduate students from different backgrounds and with a modest knowledge of mathematics and physics taught in engineering, physics, and chemistry departments, as well as biology students from similar backgrounds can understand the book.

In Chapter 2, the basic notations, definitions, and mathematical foundation are presented. The standard CNN architecture is introduced. The cell, the interconnection structure, the local connectivity pattern, the canonical equations and some useful notations, and the biological motivation are described. The importance of the local interconnection "synaptic weight" pattern, the cloning template, or gene, is emphasized. Indeed, these templates, mostly defined by 19 parameters, define the complete array dynamics, which generate an output "image" from an input "image."

In Chapter 3, after two examples, a simple technique for determining array dynamics, based on cell dynamics, is introduced and explained. Next, 11 useful templates are shown with examples and rigorous mathematical analysis.

Chapter 4 is devoted to the digital computer simulation of CNN dynamics. Numerical integration algorithms, digital hardware accelerators, as well as the analog implementation are discussed. An accompanying simulator CANDY is provided in the Appendix.

In Chapter 5 the characterization of the simplest form of a CNN is explored and the binary input binary output case is described. It is quite surprising that even this very basic form with a 3×3 neighborhood template could implement $2^{512} \sim 10^{134}$ different local Boolean functions.

Uncoupled CNN templates constitute a simple class of CNN. Their unified theory and applications described in Chapter 6 provide a thorough understanding of this class of CNN.

In Chapter 7, we begin the introduction of the CNN computer represented by the CNN Universal Machine architecture. We emphasize the need for local analog and logic memory, a global clock and global wire, as well as a local logic unit. It is shown, for example, that every local Boolean function can be realized by using these simple elements in each cell processor.

In Chapter 8, "Back to Basics," the mathematical analysis of the stability of CNN in terms of cloning templates is presented. It turns out that, in most cases, simple conditions are available to test the templates defining completely stable CNN.

The complete architecture of the CNN Universal Machine is shown in Chapter 9. Moreover, the computational infrastructure consisting of a high-level language, a compiler, operating system, and a development system are introduced. An example describing all the elementary details uncovers the basic implementation techniques.

Chapter 10 presents template design and optimization algorithms. The use of a simple program TEMPO for template optimization and decomposition is prepared and provided in the Appendix.

Many two-dimensional linear filters can be represented by CNN. These techniques are shown in Chapter 11 which also introduces the discrete space Fourier transform.

Once we allow spatial coupling, the dynamics of the CNN becomes not only much richer, but also exotic. The coupled CNN is described in Chapter 12 with a design method for binary propagation problems. In particular, it turns out that the global connectivity problem, long considered impossible by locally connected arrays, can be solved by a quite simple coupled CNN.

Nonlinear and delay type synaptic weights and their use are introduced in Chapters 13 and 14, respectively. These types of CNN are typical in modeling living neural networks as well as in solving more complex image processing problems.

In Chapter 15, we show the basics of the CMOS analog and digital implementation of the CNN Universal Machine. Indeed, the first visual microprocessor and its computational infrastructure are described. A computing power comparison is really breathtaking: about three orders of magnitude speed advantage for complex spatio-temporal problems on the same area of silicon.

Finally, in Chapter 16, the surprising similarity between CNN architecture and models of the visual pathway is highlighted. Models and some measurements in living retina are compared.

In addition to the many examples in the text, exercises at the end of the book help both students as well as lecturers to make practical use of the textbook.

The Appendices, provided via the Internet, contain a CNN template library (TEMLIB), a simple yet efficient simulator (CANDY), and a template design and optimization tool (TEMPO/TEMMASTER). These design tools provide for a working environment for the interested reader as well as for the students to explore this new field of modeling and computing. The text can be taught, typically, as a one-semester course.

New developments

More than 1000 reviewed papers and books have been published since the seminal paper by Chua and Yang on CNN technology. Recently, the scope has started to broaden in many directions. Various new forms of physical implementations have started to emerge. Optical implementation is already emerging using molecular level analog optical memory (Bacteriorhodopsine or polymer materials) and atomic[5] and molecular[6] level implementation of the CNN array as well as of the CNN Universal Machine may become feasible; the Analogic Cellular Computer represents a new platform for computing. However, this notion of computing contains brand-new elements and techniques, partially reflecting some forms of nature-made information processing.

Nature-made information processing has several different manifestations. On the *molecular level* this means the protein structures or interacting molecules on a two- or three-dimensional grid; on the *neuronal level* it may mean the many sensory organs and subsequent neural processing. On the *functional neuronal level* it may mean the information representation in spatio-temporal memory, the functional laterality of the brain, as well as the parallel processing places and functional units learned via PET, NMR, fNMR, etc. On the *mathematical-physical level* it may mean several dynamic spatio-temporal processes and phenomena represented by different nonlinear partial differential equations (PDEs). Autowaves, spiral waves, trigger waves are just a few of these exotic waves.

In modern image processing, PDE-based techniques are becoming the most challenging and important new directions. For the analogic CNN computer these are the native, elementary instructions like the multiplication, addition, XOR, NAND, etc. in digital computers. A new understanding about computing itself is emerging. The striking intellectual and scientific challenge is how to combine these diverse phenomena in useful algorithms running on a standard spatio-temporal computer, based on the CNN Universal Machine.

The analogic cellular *visual microprocessors*, embedded in a complete programming environment,[7] offer surprising system performance. Two types of tasks are becoming tractable:

Class K: Kilo real-time [K r/t] frame rate class.
The frame rate of the process in this class is in the order of about a thousand times faster than the real-time video frame rate (30 frames per second). A typical experiment is where a pattern classification with more than 10,000 frames per second was tested (more than 0.33 K r/t). Using current CMOS technology, 1.5 K r/t, that is about 50,000 frame per second, is feasible.

In this Class K, the high frame rate is the key in the computation. Clearly, the sensing and computing tasks are to be physically integrated. In standard digital technology, there is no time for A to D conversion and to complete the calculation, all within a few microseconds.

Class T: TeraOPS equivalent computing power class.
Even if the frame rate is small, like real-time video (30 frames per second), the required computing power (per chip) is enormous. Indeed, a trillion operations per second are to be – and can be – achieved. These TeraOPS chips are capable of solving a nonlinear PDE on a grid in a few microseconds. The detection of a moving inner boundary of the left ventricle in an echocardiogram, via an analogic CNN algorithm combining several waves, local logic, and morphology operators, took only 250 microseconds on the ACE4K analogic Visual Microprocessor Chip made in Seville. These chips hosted 4096 cell processors on a chip. This means about 3.0 TeraOPS equivalent computing power, which is about a thousand times faster than the computing power of an advanced Pentium processor.

A major challenge, not yet solved by any existing technologies, is to build analogic adaptive sensor-computers,[8] where sensing and computing understanding are fully and functionally integrated on a chip. Adaptive tuning of the sensors, pixel by pixel, is performed based on the content and context of the dynamically changing scene under sensing.

2 Notation, definitions, and mathematical foundation

2.1 Basic notation and definitions

Definition 1: Standard CNN architecture
A *standard CNN architecture* consists of an $M \times N$ rectangular array of cells $(C(i, j))$ with Cartesian coordinates (i, j), $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, N$ (Fig. 2.1).

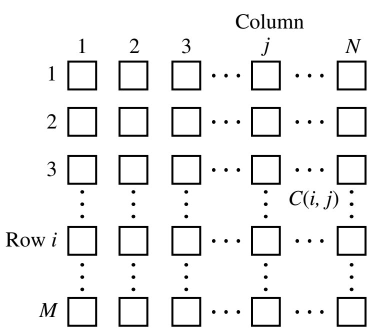

Fig. 2.1.

Remark:
There are applications where $M \neq N$. For example, a 5×512 CNN would be more appropriate for a scanner, fax machine, or copy machine.

Definition 2: Sphere of influence of cell $C(i, j)$
The *sphere of influence*, $S_r(i, j)$, of the radius r of cell $C(i, j)$ is defined to be the set of all the neighborhood cells satisfying the following property

$$S_r(i, j) = \{C(k, l) | \max_{1 \le k \le M, 1 \le l \le N} \{|k - i|, |l - j|\} \le r\} \tag{2.1}$$

where r is a positive integer.

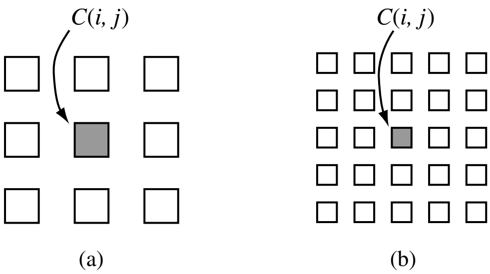

Fig. 2.2. (a) $r = 1$ (3×3 neighborhood), (b) $r = 2$ (5×5 neighborhood).

We will sometimes refer to $S_r(i, j)$ as a $(2r + 1) \times (2r + 1)$ neighborhood. For example, Fig. 2.2(a) shows an $r = 1$ (3×3) neighborhood. Fig. 2.2(b) shows an $r = 2$ (5×5) neighborhood.

Remarks:

1 In IC implementations, every cell is connected to all its neighbors in $S_r(i, j)$ via "*synaptic*" circuits.

2 When $r > N/2$, and $M = N$, we have a fully connected CNN where every cell is connected to every other cell and $S_r(i, j)$ is the entire array. This extreme case corresponds to the classic Hopfield Net. It is impractical to build any reasonable size (several thousand cells) Hopfield Net in a VLSI chip. There exists a "commercial" Hopfield-like chip by INTEL called "ETANN," type 80170 ($500 in 1995). This chip has 64 cells which makes it more of an expensive "toy."

Definition 3: Regular and boundary cells

A cell $C(i, j)$ is called a *regular cell* with respect to $S_r(i, j)$ if and only if all neighborhood cells $C(k, l) \in S_r(i, j)$ exist. Otherwise, $C(i, j)$ is called a *boundary cell* (Fig. 2.3).

Remark:

The outermost boundary cells are called **edge cells**. Not all boundary cells are edge cells if $r > 1$.

Definition 4: Standard CNN

A *class 1* $M \times N$ *standard CNN* is defined by an $M \times N$ rectangular array of cells $C(i, j)$ located at site (i, j), $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, N$. Each cell $C(i, j)$ is defined mathematically by:

1 State equation

$$\dot{x}_{ij} = -x_{ij} + \sum_{C(k,l)\in S_r(i,j)} A(i, j; k, l)y_{kl} + \sum_{C(k,l)\in S_r(i,j)} B(i, j; k, l)u_{kl} + z_{ij} \tag{2.2}$$

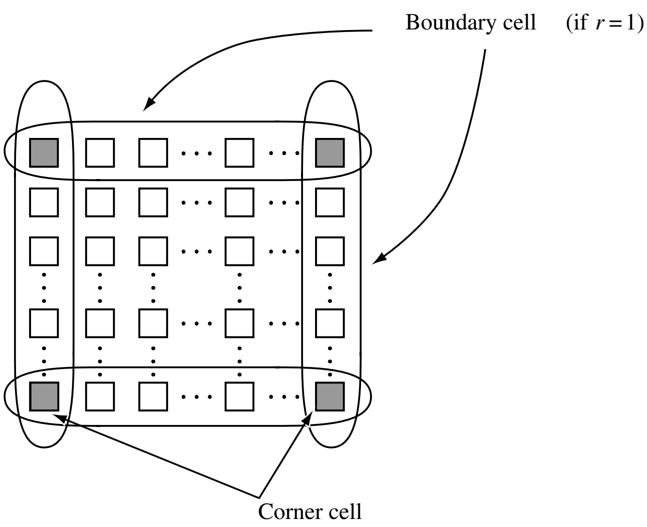

Fig. 2.3.

where $x_{ij} \in R$, $y_{kl} \in R$, $u_{kl} \in R$, and $z_{ij} \in R$ are called **state**, **output**, **input**, and **threshold** of cell $C(i, j)$, respectively. $A(i, j; k, l)$ and $B(i, j; k, l)$ are called the **feedback** and the **input synaptic** operators to be defined below.

2 Output equation

$$y_{ij} = f(x_{ij}) = \frac{1}{2}|x_{ij} + 1| - \frac{1}{2}|x_{ij} - 1| \tag{2.3}$$

This is called standard nonlinearity (Fig. 2.4).

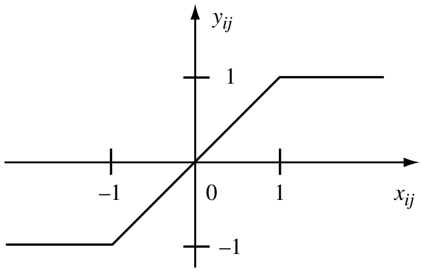

Fig. 2.4.

3 Boundary conditions

The boundary conditions are those specifying y_{kl} and u_{kl} for cells belonging to $S_r(i, j)$ of edge cells but lying outside of the $M \times N$ array.

4 Initial state

$$x_{ij}(0), \quad i = 1, \ldots, M, \quad j = 1, \ldots, N \tag{2.4}$$

Remarks:

1 The input u_{kl} is usually the *pixel* intensity of an $M \times N$ gray-scale image or picture **P**, normalized without loss of generality, to have the range $-1 \leq u_{kl} \leq +1$ where "white" is coded by -1 and "black" is coded by $+1$. For a *still* image, u_{kl} is a constant for all time, for a moving image (video) u_{kl} will be a function of time. Other variables $(x(0), y, z)$ can also be specified as images.

2 In the most general case, $A(i, j; k, l)$, $B(i, j; k, l)$, and z_{ij} may vary with position (i, j) and time t. Unless otherwise stated, however, we will assume they are space and time invariant.

3 In the most general case both $A(i, j; k, l)$ and $B(i, j; k, l)$ are nonlinear operators[1] which operate on $x_{kl}(t)$, $y_{kl}(t)$, $u_{kl}(t)$, $x_{ij}(t)$, $y_{ij}(t)$, and $u_{ij}(t)$, $0 \leq t \leq t_0$, to produce a scalar $(A(i, j; k, l) \circ y_{kl})(t_0)$ and $(B(i, j; k, l) \circ u_{kl})(t_0)$, $0 \leq t \leq t_0$.

4 We may also introduce synaptic laws depending on the states (C template) and on mixed variables (D template), respectively.
That is $(C(i, j; k, l) \circ x_{kl})(t_0)$ and $(D(i, j; k, l) \circ (u_{kl}, x_{kl}, y_{kl})(t_0)$.

Unless otherwise stated, however, $A(i, j; k, l)y_{kl}$ and $B(i, j; k, l)u_{kl}$ will denote ordinary multiplication with real coefficients where they may be nonlinear functions of states, inputs, and outputs of cells $C(i, j)$, $C(k, l)$ and **may** involve some **time delays** (i.e., they may contain a finite time history, as in the case of having a time delay).

The following are some space and time invariant nonlinear examples chosen from the CNN catalog of applications (CNN Software Library). See some of them in TEMLIB (Appendix A).

EXAMPLE 2.1:

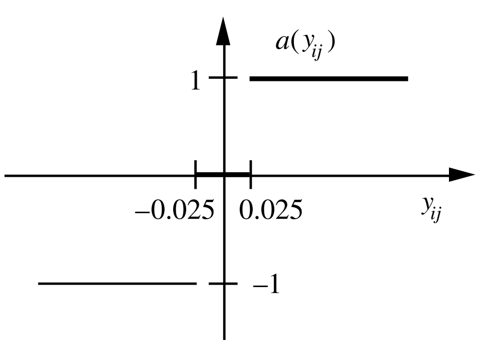

Fig. 2.5.