CHAPTER 1

INTRODUCTION

1.1 BACKGROUND OF ARTIFICIAL NEURAL NETWORKS (ANNs)

Artificial neural networks (ANNs) are a parallel system that are constructed a biological systems resemble the human brain. They are a simply large class of mathematical algorithms that consists of massive and simple processing elements. The massive and simple elements connect with each other, operate in parallel, and communicate by sending signals to each other over a large number of weighted connections. The function of ANN can be used to learn the training patterns or data, recover the test noisy patterns or data to the correct ones or any tasks, and associate the other patterns depend on the learned patterns or data. It need massively distributed and parallel processing element to process the signal computation in biological system to speedup the tasks. Hence, the massive and simple processing element is also called as “neuron” or “cell”. Based on the parallel structure, the ANNs are suitable for many tasks that need massive parallel processing such as pattern recognition and classification, image processing, optimization, and data clustering while the traditional computers are inefficient for the above tasks. Despite the operational speed of each simple element is much slower than the traditional computers with the Von Neumann structure, but the speed of the traditional computers is limited to the

bottleneck of the I/O for input signals and a large amount computation of the analog signals between the CPU and storing memory, or image sensing circuit for the image processing.

The analysis and modeling of human neural phenomena has a very interesting research field with the potential applications in machines performing human-like speech and image processing. So far, two approaches [1] in neural models have been proposed. One is the biological model that studies the structure and function of real brains to explain biological phenomena. The other is the technological model that studies brains to extract concepts and applies them in new computational methodologies. The schematic diagram of a biological neuron is shown in Fig. 1.1(a), where the corresponding schematic diagram of a McCulloch-Pitts (M-P) neuron [2] is shown in Fig. 1.1(b). Generally, the second approach is based on present understanding of the first one, the biological nervous systems. Based upon the second approach, artificial neural networks (ANNs) have been developed to mimic both structures and functions of brains and nervous systems. The comparisons of biological and technological approaches are summarized in Table 1.1.

The biological model of the human brain consists of approximately 10^{11} neurons of many types. Some of them are responsible for the visual signals, some of them handle with the aural signals, some of them are for the olfactory signals, and some of them are for the tactile signals. The fact is that their behavior model is very different, but their architecture of the processing is very similar. The schematic diagram of a biological neuron is shown in Fig. 1.1(a). The Nucleus, 5-100 microns in diameter, is inside the cell body, and works as a control unit. All weighted messages incoming impulse from Dendrites are summed in the Nucleus, and then it decides the neuron to be excited or inhibited and sends the exciting or inhibiting message to the other neurons through neuron's Axon. The Axon is a strand of single and long output path

of the neuron whereas the Dendrites are several and short tree-like input paths of the neuron. The Axon is electrically active, and it produces and propagates the electrical pulse emitted from the Nucleus. The synapses are located at the end of the Axon, and excrete neurochemical to weight and transfer the message from one neuron's Axon to another neuron's Dendrite. Each synaptic connection is caused by the excitatory or inhibitory reactions of the receiving neuron. The weight is practically assigned positive or negative, a positive weight corresponding to excitatory synapse, whereas the negative weight to inhibitory synapse. There are approximately 10^4 synapses for each neuron in a human brain.

A simple block diagram of the biological neuron presented by McCulloch and Pitts [2] is shown in Fig. 1.1(b). In this model, the expression of the output y_i of cell C(i) can be written as

$$y_i = f(x_i) = f\left(\sum_{j=1}^{N} w_{ij} y_j + z_i\right) \tag{1.1}$$

where the x_i is the cell state of cell C(i), y_i is the output from the cell C(i), y_j is the output from another cell C(j), z_i is the threshold of the cell C(j), and $f(x_i)$ is the nonlinear activation function. The activation function $f(x_i)$ in the McCulloch-Pitts (M-P) neuron [2] is a unit-step function written as

$$f(x_i) = \begin{cases} 1 & \text{if } x_i \equiv 0 \\ 0 & \text{otherwise} \end{cases} \tag{1.2}$$

The i-th processing element, also called as cell C(i), computes a weighted sum of its inputs, the outputs y_j of other processing elements, and then outputs $y_i = 1$ (excitatory) or $y_i = 0$ (inhibitory) if the weighted sum is above or below a certain threshold z_i. The weight w_{ij} represents the strength of the synapse connecting cell C(j) (source) to cell C(i) (destination).

The technological models of ANNs have been studied for a long time. In the

early time, McCulloch and Pitts [2], Hebb [3], Rosenblatt [4], Widrow [41], and Grossberg [5] developed several mathematical models. The work of Hopfield [6]-[7], Rumelhart and McClelland [8]-[9], Sejnowski [10], Feldman [11], Grossberg [12]-[13], and others has led to a new resurgence of the field. Due to the development of new network topologies, algorithms [6]-[9], [11]-[18], applications, and new VLSI implementation techniques with some intriguing demonstrations, a growing research effort in this field has been reported [19]-[28]. Especially, more and more efforts and interests have focused on the development of technological models and VLSI hardware for various neural networks.

An ANNs structure has been proposed that are based on the present understanding of biological nervous system [138]. Basically, the models of neural networks explore many complete hypotheses to simultaneously use massively parallel networks containing many cells or processing elements connected by linking with variable weights called the synapses. These neural network behaviors models are specified by the network topology, neuron characteristics, and training or learning algorithms. The training or learning rules used to update the initial weights and indicate how the weights adapted to the variable environment [5], [12], [41, [91]. As the ANNs adapt to the changes in their environment, they can develop their own internal rules. Also, the ANNs employ an enormous number of communication synapses among the processing elements to perform distributed parallel processing.

The key features of ANNs are asynchronous parallel processing, continuous or discrete time dynamics, and global interaction of the neural systems [138]. Their structure can be defined an interconnection of neurons, through the weights to the other neurons may including themselves in the neural network. The organization of interconnection has two models; one is elementary feed-forward architecture of m neurons receiving n inputs. The expression of the i-th neuron output y_i is

$$y_i = f(x_i) = f(\sum_{j=1}^{n} w_{ij} x_j + z_i) \text{ for i} = 1,2,.....\text{m} \quad (1.3)$$

where the weight w_{ij} connect the i-th neuron with the j-th input x_j, the activation function f can transform the state to output of the i-th neuron. The other organization is the feedback system that the neurons output are feedbacks connecting to their inputs.

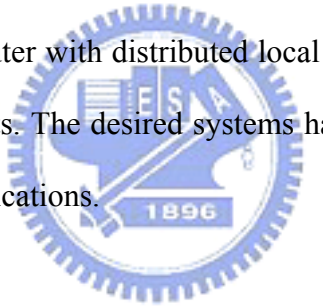
Typically, ANNs system can provide a greater degree of robustness or fault tolerance than Von Neumann sequential computers because the ANNs contain more processing elements. A few nonfunctional processing elements or synapses computation not greatly affect the overall operation in the neural network. Generally, inherently massive parallel structures with learning and fault tolerance are the most attractive and distinguishing features of the ANNs. Another significant advantage of the ANNs is their ability easy to handle fuzzy or Chaos operation for the gray data, noisy data, or data stream.

Recently year, the ANNs system has been greatly used that usefully to solve the engineering problems for several applications, such as pattern recognition [20], image processing [21]-[22], speech processing [23]-[24], retinal vision [25]-[34], optimization [35], communication [36], [38], signal classification [39], robotics [40], control systems [42]-[43], monitoring applications [44], power systems [43], learning grammars [45], neural computing [46]-[49], pattern association [173] and so on. The above applications will be addressed in more details later.

The useful ANNs in addressing problem requiring recognition of complex patterns and performing nontrivial mapping functions is called the back-propagation network (BPN). The network is designed to operate in the multilayer and feed-forward neural network for the supervised mode of learning, as content-addressable memory. The exemplar patterns are applied as a stimulus to the input layer of the network, it is propagated through each hidden layers until the output

is generated. The output pattern is compared with the desired output and the produced error signal is computed to update from the output unit. The BPN need massive weights to connect with each neuron for the inter-layers.

The cellular neural network (CNN) is a special form of ANNs that a large-scale analog nonlinear circuit to process the signals in parallel. A CNN have a feature of local connectivity property, each neuron connected only the set of neighboring cells. It has some processing ability in many applications, such as real-time image process [44], pattern recognition [80], [84], motion detection [22], etc. Thus the computations of CNN are very suitable for VLSI implementation. Based upon the CNN structure, the CNN universal machine (CNN-UM) architecture was designed for various pattern applications [157]-[162]. The CNN-UM is a cellular analogic stored-program multidimensional array computer with distributed local analog and logic memories as well as analog computing units. The desired systems have a capability to process the input patterns for various applications.



1.2 APPLICATIONS OF NEURAL NETWORKS

There are very powerful applications of ANNs to process the engineering problems [46]-[50]. Since the ANNs are applied to emulate and implement the brain information for image processing, it is not surprising that many applications of the ANNs are similar to the functions of a human brain such as vision processing, speech processing, communications, and adaptive control. Those functions can easily outperform a human brain activates from the fastest supercomputer with the smartest algorithms.

The neural networks have been applied in many major engineering problems to solve a good solution. Some applications are listed as the following:

(1) **Pattern recognition** [20], [51]-[54] using the Hopfiled, Cellular, or Hamming neural networks structure to implement as an associative memory. The association learning memory can be separated to two parts, one is auto-association that is to retrieve a complete pattern, given partial information of the desired pattern, the other is hetero-association that is to retrieve a corresponding pattern in one subset, given a pattern in another subset;

(2) **Image processing** [55]-[58] using different kinds of cellular nonlinear (neural) networks, including the Single-neighborhood CNNs (SN-CNNs), Large-neighborhood CNNs (LN-CNNs), Continue-time CNNs (CTCNNs), Discrete-time CNNs (DTCNNs), Mutilayer CNNs (MCNNs), Delay-time CNN (DCNN), and CNN universal machine (CNNUM);

(3) **Speech processing** [23]-[24] using the transmission-line Zwislocki model neural network;

(4) **Retinal vision** [29]-[34] using complex-log mapping model or the neuron BJT(νBJT);

(5) **Optimization** [35] using Hopfield neural network;

(6) **Communication routing** [36]-[38] using the modification of the neural network traveling salesman algorithm;

(7) **Signal classification** [54], [62], [132] using two-layer back-propagation associative memory (BAM) for missile homing and handwritten digit recognition;

(8) **Control system and robot** [40] using the structured hierarchical network of the back-propagation model;

(9) **Self-learning adaptive control and comparison systems** [41]-[43] using the popular cerebella model arithmetic computer (CMAC) developed by Albus;

(10) **Vision applications** [44], [66] using Hopfield model neural network to

recognize anomalies in ultrasound images;

(11) **Power systems** [42] using the layered perceptron neural networks for electrical load forecasting in power systems;

(12) **Learning grammars** [45] using the second-order recurrent neural networks for grammatical inference;

(13) **Chaotic** [88], [93], [95], [154]-[155] using chaotic to implement the associative memory or to find the weights base on energy function;

(14) **DNA analysis** using the neural system to processing the DNA analysis.

In the pattern recognition, one of the important functions is pattern classification. The goal of a pattern classifier is to decide which of M exemplar patterns is the most similar to an unknown (noisy or distorted) input pattern that is composed of N elements. Pattern classifiers can be used for three different tasks. 1) They can identify which class best represents the input pattern that even could have been corrupted by noise or some other process. 2) They can be used as a content-addressable or associative memory. 3) They can be used to vector quantities or cluster the N inputs into M clusters.

Although the conventional pattern classifiers have been developed for a long time, the approach is quite different from the neural network classifier. There are two major differences between neural network classifiers and conventional classifiers as described below. 1) Neural network classifiers are nonparametric. They make weak or no assumptions about the probabilistic distributions of the input and the exemplar patterns. However, the conventional classifier needs to make these assumptions. 2) Neural network classifier is of parallel processing whereas the conventional classifier is of sequential processing.

In many structures of ANNs, the Hopfield neural network has been used in several pattern recognition applications. Character recognition, recognition of random

patterns, and bibliographic retrieval has been demonstrated in the applications of Hopfield neural network. However, the fully interconnection structure of Hopfield neural network limits the numbers of cells. On the other hand, CNNs have broad applications in image processing [55]-[58], video signal processing [58], [70], artificial vision [66], solving PDEs [62], modeling biological systems [67]-[68], robotic and biological visions [65]-[66], and higher brain functions [69]-[70]. More recently, it has also been used as a paradigm for generating state and dynamic patterns, autowaves, spiral waves, scroll waves, and spatial-temporal chaos [71] with diverse applications in image and video signal processing [72]. Generally, all applications where the signal array can be represented by a geometric grid, call for a CNN approach. The most results of learning are not directly input from the patterns or data in CNNs.

1.3 HARDWARE IMPLEMENTATION

The ANN hardware can be implemented by digital, analog, or mixed signal circuits [89], [159] that is used many application for image processing. Design methodology of VLSI neural networks that purely analog signal design has a significant performance for a massive number of computing tasks using the transistors in the sub-threshold region can greatly reduce the power dissipation. The desired ANN hardware implementation can to solve the variety of problems and the theory. The integrated circuits are able to improve the execution speed. Generally, the major problems in VLSI implementation are described as [64]

(1) The analog memories and storages are difficult realized

(2) The massive interconnections among neurons are difficult to design in VLSI.

(3) The fabricated chip areas limit the realizable neurons.

(4) The visual input circuits are aligned with the pixels for learn and test patterns.

In the ANNs, the hardware implementation of neurons, weights and optic has been proposed [20], [28], [161], [182]. The implementation approaches are based on the types of digital [63]-[65], [110]、analog [28]-[29], [72]or both combination mixed mode [89], [159], [182]. The digital approaches were most used in the coherent or digitalize neural system for the hardware implementation. Generally, the analog and mixed mode approaches can be implied into the continuous-time [139] or the discrete-time [64], [142] analog CMOS systems that perform the variety of operation for the neural systems.

Among the proposed many VLSI hardware structure [19]-[22], [36]-[37], [89]-[90] one of the storage element is stored the analog value in the off-chip digital memories, eg., SRAM or DRAM [179]. Another analog storage element is used the floating-gate device as memory devices [112]. The analog value can be stored for a long-time. The floating-gate device can also be combined with the multiplier circuit that to process the synapse computation and memory problems. The circuit is as well as the real nervous system with the memory and weighting characteristics. However, the integrated circuit has been limited the numbers of neurons and synaptic to implement on a single chip. The several hardware implementation techniques have been developed to solve the chip limitation problems. There techniques include the wafer scale integration (WSI) [21], the modular chip design [39], [123], [158]-[159], and the expandable chip design [179]-[182]. The WSI technique can offer the higher density and speed in the neural circuit implementation. The multi-chip design technique is used in modular design methodology. The system is divided to different function block are designed in the separated chip modular, respectively. The expandable chip is combined design from the sub-systems to the fully system. The

required scale of the system can expanded from the derived system to process the input patterns.

In the last few years, the Cellular Neural Network Universal Machine (CNN-UM) [158]-[159], [179]-[182] chip implementation overcome the CNN used in the real-time for image processing applications. The implementation of the CNN-UM is constituted from the CNN local memory, CNN neurons prototyping, the system processor, and the chip controlled software. The time-multiplexed methods need included within the definition and development of an appropriate hardware platform for the CNN chipset.

1.4 RESEARCH MOTIVATION AND ORGANIZATION OF THIS THESIS

1.4.1 Research Motivation

The ANNs with the specific algorithm has been widely used for the various processing and applications. The dynamic image or biological neurons signals processing is the key feature work. The real-time images or signals are processed by the hardware of neural system to satisfy the time required. The hardware implementation of ANNs can practically applied the real system to makes the real-time image processing systems. The activity of the hardware circuit is able to understanding the operations of the biological neurons or system models. However, there are still some problems for the hardware implementation of the complicated ANNs models and algorithms to be solved. Generally, the hardware implements have five major problems, as

(1) It is hard to implement the massive even fully interconnections between the cells because the limit on the 2D VLSI technology.

(2) The complicated learning to enhance the ability of ANNs but the hardware is increased the design overhead.

(3) It is difficult to implement the massive and matching analog storages or memories for the storing weights or inputs.

(4) The total learning time is dependent on the numbers of learned patterns. It means that the total learning time cannot easily used in flexibility.

(5) The used chip area exceeds the desired wafer scale as comparing the biological neuron or the system with the same numbers of neuron.

Based upon the described reasons, it is important to reduce the area of basic neural cells, enhance the integration density for efficient neural function implementation, and develop easy-implemented learning. To achieve these goals and solve the problems of hardware implementation problems mentioned before, an attractive implementation method of the ANN is to use locally connected implementation instead of full-connected implementation. In the full-connected implementation method, it always needs massive connections to realize a function. In the locally connected implementation, some effort [55]-[56] has been contributed to implement ANN functions using the local connectivity. Thus, the neural circuit became a better simple network and the large-size ANNs can be integrated on a single chip. Based upon this approach, a new structure called the cellular nonlinear network [57]-[63] has been proposed recently. But the effectively applications on the CNN implementation has some limited.

It is one aim of this thesis research to explore new learning rules suitable for ANN implementation and demonstrate the applications of the new learning rule in ANN implementation, which could solve the interconnection, chip area, and power dissipation problems as above mention. Base upon the physical characteristics of semiconductor devices, a new structure called the ratio-memory cellular nonlinear network (RMCNN) [77], [166]-[175] for the compact implementation of VLSI neural network has been

proposed and analyzed in this thesis. In the new structure, the parasitic PNP bipolar junction transistor (BJT) in the CMOS processing is used to implement the function of multiplication and division. The BJT-based multiplication/division structure has the advantages of compact structure and small chip size. The proposed BJT-based multiplication/division structure in [77] has been used to realize the analog cellular nonlinear network with ratio memory [165]-[175].

Many VLSI hardware structure have been proposed in the past years to implement the associative memory [81]-[89], the Hopfield neural network with Hebbian learning rule is one of the solutions [6]-[7]. But the original Hopfield neural network has massive interconnections between cells, this puts some limitations on the applications to pattern classifications in the real world. One of the improvement methods is to reduce the interconnections between neuron with some modified learning rules [77], [169]-[172]. A spare connections structure such as cellular nonlinear (neural) network with ratio memory has better ability in the pattern recognition applications. In this research, the proposed parasitic BJT is applied to the multiplication/division implementation in the analog cellular nonlinear (neural) network with ratio memory that can store more exemplar patterns.

Due to the advantageous feature of local connectivity, the cellular nonlinear (neural) network (CNN) introduced by Chua and Yang [55] is very suitable for VLSI implementation and many applications [56]-[58]. So far, many research works on the applications of CNNs as neural associative memories for pattern learning, recognition, and association have been explored [59]-[61], [91], [80]-[103], [107]-[109]. Among them, many innovative algorithms and software simulations of CNN associated memories were reported [59]-[61], [77], [81]-[84]. Moreover, CMOS chip implementation of CNN associative memory was reported in [171].

In realizing CNN associative memories, the learning circuitry can be integrated

with CNNs on-chip. The major advantages of on-chip learning are: 1) The host computer isn't needed to perform the learning task at off-line. Therefore, the simple interface for the neural system chips in many practical applications 2) The spatial-variant template weights can be on-chip learned on the CNN chips without being loaded from the computer system. Thus long loading time, complex cell global interconnection, and analog weight storage elements to perform the loading operation for large numbers of spatial-variant template weights can be avoided; 3) The adaptability to the process variations of CNN chips can be enhanced.

Another aim of this thesis research is to explore a new indirectly connective image-processing platform suitable for the implementation of CNNUM with large-neighborhood templates. In the some CNN applications [72] like Muller-Lyer illusion, image halftoning, color vision, and de-blurring, the templates with large neighborhood, i.e. $r > 1$, are required. To realize large-neighborhood templates in CNN structures, the template decomposition methods have been proposed that can be implemented on CNN universal machine (CNNUM) [158]-[159]. Generally, it is difficult to directly implement with the large-neighborhood templates through single CNN operation. It has the advantages of small chip area and high integration capability. This approach is able to explore in the future. Thus, the indirectly connective CNN structure is very suitable to realize CNNs with large-neighborhood templates [160].

1.4.2 Thesis Organization

This thesis contains six chapters, which include introductions, the design and analysis of self-feedback ratio-memory cellular nonlinear network (SRMCNN), the design of self-feedback ratio-memory cellular nonlinear network with template **B** for hetero-associative memory applications, the analysis for large-neighborhood cellular

nonlinear network (LN-CNN) and ratio memory, and the hardware implementation of the SRMCNN with on-chip learning and storage.

Chapter 1 introduces the background, some kinds of ANNs and how to implement them, the applications of ANNs, the hardware implementation of ANNs, and describes the main research motivation.

In Chapter 2, comparisons among the recently reported ANN basic activation functions and learning rules are given. The fundamental concepts and models of the cellular nonlinear (neural) networks and their application are also reviewed in this chapter.

In Chapter 3, the RMCNN with the modified Hebbian learning algorithm with self-feedback is proposed and analyzed. The new RMCNN is called self-feedback RMCNN (SRMCNN). In the learning process of the proposed SRMCNN, the features from input exemplar patterns are considered to update the weights. The operation of SRMCNN retains the feature enhancement effect of the RM. With RM and the modified Hebbian learning algorithm with self-feedback, the SRMCNN can be used as the associative memory for learning, recognizing, and recovering patterns. Detailed analysis and simulation results has shown that the SRMCNN can recognize up to 93 noisy patterns with a 100% success rate and 98 noisy patterns with a 97% success rate after learning the input exemplar patterns in uniform (normal) noise level is 0.8 (0.3). Thus, the capacity for learning and recognizing patterns is greatly improved.

In Chapter 4, the architecture with embedded ratio memory and realized the modified Hebbian learning algorithm in the SRMCNN with **B** template is presented. It can learning the input exemplar patterns and correctly output the recognized patterns. The weights of the **B** template are generated from the desired output pixel value produced by the nearest five neighboring element as associative memory for all input exemplar patterns. The learned weights are processed in the ratio with the

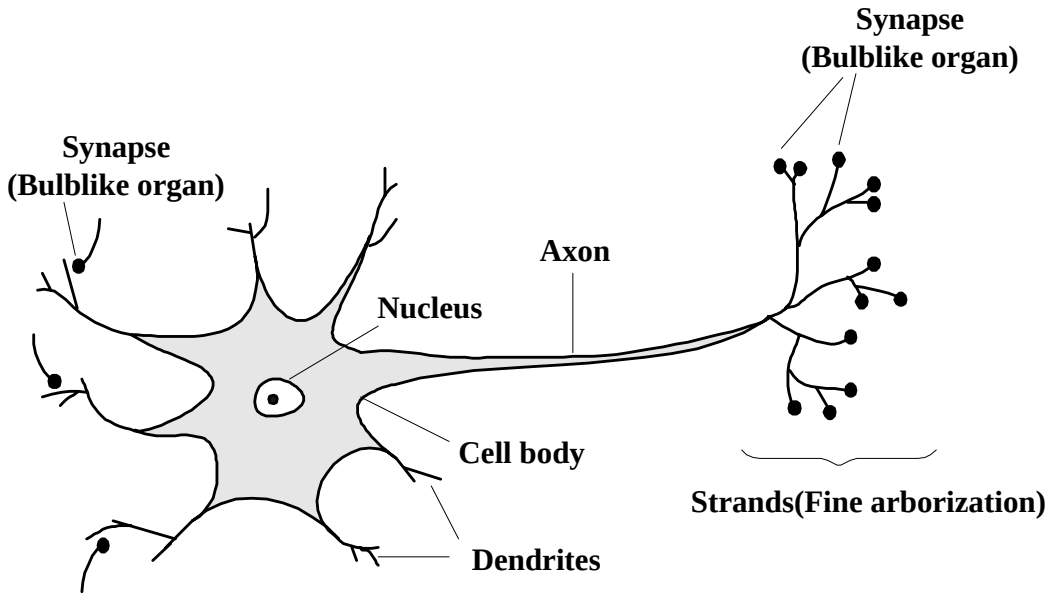
summation of absolute coefficients on the **B** template. The efficiency of ratio memory is enhanced the feature of pattern. The simulation results of the behavior and function of SRMCNN with **A** and **B** templates for hetero-associative memory applications are demonstrated and analyzed. The capability of SRMCNN for pattern learning and recognition is improved.

In Chapter 5, the structure of the SRMCNN with **B** template and the modified Hebbian learning algorithm for auto-associative memory are proposed. The function blocks have implemented in the VLSI circuits for the 0.25 μm 1P5M n-well CMOS technology. The characteristics of the proposed circuits are correctly verified by the HSPICE software. The function of ratio memory for one bit SRMCNN with B template was realized in the VLSI chip and their operation was shown. The simulation results of the 18x18 SRMCNN behavior and function are demonstrated and analyzed. The layout graphic of 9x9 SRMCNN is presented and its function is verified. The 18x18 SRMCNN is combined for four chips of 9x9 SRMCNN. The capability of pattern learning and recognition is improved. The conceptual design for the general architecture of the Large-Neighborhood Cellular Nonlinear (Neural) Network Universal Machine (LN-CNNUM) is described.

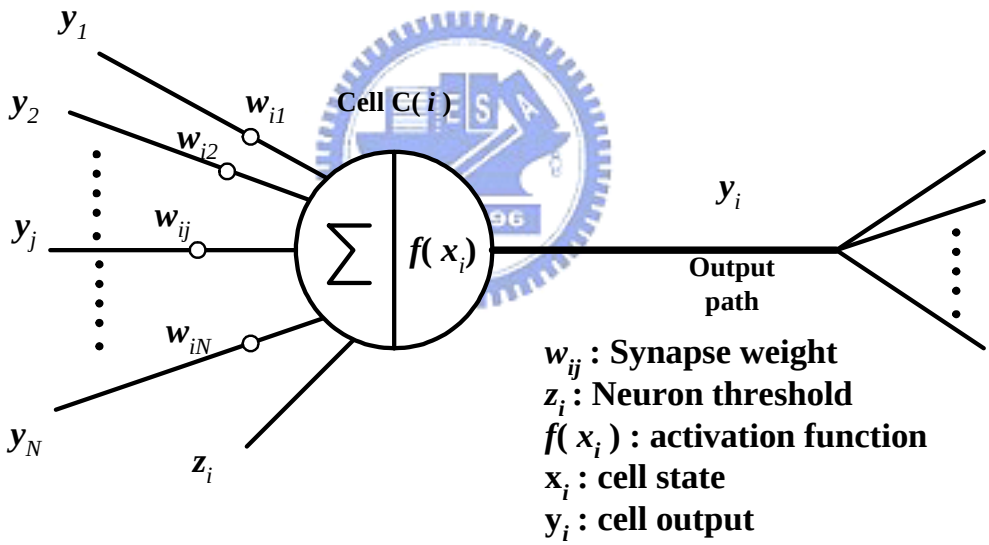
Finally, the conclusion of this thesis is summarized in Chapter 6. Some suggestions for the future works about the implementation of CNNs with ratio memory and their associative memory applications to various image processing are also addressed in the same chapter.

Table 1.1 The comparison of biological and technological approaches.

	Biological Approach	Technological Approach
Implementation	Biological neurons	Integrated circuits
Interconnections	Massive	Spare
Message transferring form between neurons	Chemical material	Current or voltage
Architecture	Complex	Simple
Structure	Weak	Solid
Computational speed	Slow (1 ms)	Fast (1ns~1ms)
Transmission	Pulse transmission	Activity value and connection strengths



(a)



(b)

Fig. 1.1 (a) The schematic diagram of a biological neuron; (b) The corresponding schematic diagram of a McCulloch-Pitts (M-P) neuron.