Tang Tang

Image Compression by Means of Cellular Neural Networks

Beiträge aus der Informationstechnik

Tang Tang

Image Compression by Means of Cellular Neural Networks

VOGT

Dresden 2015

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.dnb.de abrufbar.

Bibliographic Information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available on the Internet at http://dnb.dnb.de.

Zugl.: Dresden, Techn. Univ., Diss., 2015

Die vorliegende Arbeit stimmt mit dem Original der Dissertation „Image Compression by Means of Cellular Neural Networks“ von Tang Tang überein.

Gesetzt vom Autor

ISBN 978-3-938860-84-7

Jörg Vogt Verlag
Niederwaldstr. 36
01277 Dresden
Germany

Phone: +49-(0)351-31403921
Telefax: +49-(0)351-31403918
e-mail: info@vogtverlag.de
Internet : www.vogtverlag.de

Technische Universität Dresden

Image Compression by Means of Cellular Neural Networks

Tang Tang

von der Fakultät Elektrotechnik und Informationstechnik der Technischen Universität Dresden

zur Erlangung des akademischen Grades eines

Doktoringenieurs

(Dr.-Ing.)

genehmigte Dissertation

Vorsitzender:	Herr Prof. Dr.-Ing. habil. R. Schüffny		
Gutachter:	Herr Prof. Dr. phil. nat. habil. R. Tetzlaff	Tag der Einreichung:	03.11.2014
	Frau Prof. Dr. A. Slavova	Tag der Verteidigung:	16.02.2015

Thanks to the supervision of Mr. Prof. Dr.-phil.nar.habil. Ronald Tetzlaff. I will specially thank Mr. Dr. Alon Ascoli, Mr. Dr. Weiran Cai and all the colleagues in the Lehrstuhl für Grundlagen der Elektrotechnik.

Abstract

The requirement of high-quality image compression methods is becoming more and more critical with the flourishing of multimedia applications. The development of the cellular computing, e.g. cellular automata (CA), Cellular Neural Networks (CNN), in the past few decades has revealed various feasibility to implement these kinds of computational frameworks for signal processing. Due to their massive parallel nature, CNN have been proven well suitable for image processing. In this thesis, inspired by Dogaru's work, a wavelet-based image compression method is proposed. The CNN paradigm is implemented in an image compression scheme as far as possible. In this thesis, the nonlinear dynamics and the parallel computing capability of CNN have been investigated. Different CNN-based algorithms of operations involved in the image compression have been developed. The proposed method has proven a comparable performance in both objective quality and the perceptual quality to that of the JPEG 2000 standard for image compression applications where a high compression ratio is required. The results obtained in this thesis show that the application of a CNN-based image compression method can lead to a high-quality while retaining a low system complexity by taking advantage of the CNN paradigm.

Contents

Nomenclature

AC:	alternating current
ADPCM:	Adaptive Differential Pulse Code Modulation
AR:	autoregressive
CA:	cellular automata
CALIC:	Context-based, Adaptive, Lossless Image codec
CDF:	Cohen-Daubechies-Feauveau filter
CNN:	Cellular Neural Networks
CNN-UM:	CNN Universal Machine
DM:	Delta modulation
DCT:	discrete cosine transform
DC:	direct current
DFT:	discrete Fourier transform
DPCM:	differential Pulse Code Modulation
DT-CNN:	Discrete-Time Cellular Neural Networks
EBCOT:	Embedded Block Coding with Optimal Truncation
GAP:	gradient-adjusted prediction
HPC:	higher order predictive coding
ID:	identity
JPEG:	Joint Photographic Experts Group
JPEG-LS:	JPEG-lossless standard
KLT:	Karhunen Loeve Transform
LBG:	Linde, Buzo, and Gray
LOCO-I:	a Low Complexity, Context-Based, Lossless Image Compression Algorithm
LSB:	Least Significant Bit
LZW:	Lempel-Ziv-Welch coding scheme

MPC:	multistage predictive coding
MSB:	Most Significant Bit
MSE:	mean square error
ODE:	ordinary differential equation
Pixel:	picture element
RMSE:	relative mean square error
PDE:	partial differential equation
PPC:	polynomial predictive coding
PSNR:	peak signal-to-noise ratio
ROI:	region of interest
RD-CNN:	reaction-diffusion CNN
SNR:	signal-to-noise ratio
SOFM:	self-organizing feature map
STFT:	Short Time Fourier Transform
VQ:	vector quantization
WFT:	Windowed Fourier transform
WHF:	Walsh-Hadamard transform
2D:	two dimensional
A:	feedback synapses in CNN
B:	control/ feed-forward synapses in CNN
Bior:	biorthogonal wavelet
iid:	independent and identically distributed random variable with zero-mean
Round(·):	rounding function
x:	state of CNN
y:	output of CNN
u:	input of CNN
z:	bias value in CNN
$\psi(t)$:	mother wavelet
$\varphi(.)$:	activation function in neural networks
∇^2:	Laplacian operator
$\oplus$:	xor (exclusive or) operation

Chapter 1

Introduction

1.1. Motivation

Cellular Neural Networks/ Cellular Nonlinear Networks (CNN), developed in the end of the 1980s by L. Chua and L. Yang, were the results of advances in the development of Cellular Automata (CA) and neural networks. The CNN paradigm serves not only as a parallel computing framework, but also as a kind of neural networks. As a circuit-oriented architecture, CNN have been considered for circuit implementation since the very beginning. Due to their massive parallel nature, CNN have been proven well suitable for image processing. In the last few years, plenty of works of image processing have been made by means of CNN, on the other hand, investigations on using CNN for image compression are still relatively rare.

Dogaru et al. introduced a vector quantization-based image compression method [Dog06] where the codebook is generated as emergent patterns of CA/ CNN. Even though this method is unsuitable for practical application due to the low compression ratio and performance, their attempt enlightened a feasibility of using the complex phenomena, e.g. emergent patterns, for image compression.

In this thesis, Dogaru’s work should be firstly analysed in detail. A more efficient CNN-based image compression method should be developed. In the coding process, the coding image, represented initially in spatial domain, should be represented by different independent frequency components by considering a wavelet transform. Therefore, a CNN-based algorithm for 2D wavelet transform should be developed. The obtained wavelet coefficients should be quantized. Regarding the statistic characteristics of the coefficients in different frequency channels, distinctive quantization strategies should be proposed. Correspondingly, quantized wavelet coefficients in different frequency channels should be encoded by applying different strategies. The coefficients in a low frequency channel should be coded with high accuracy as far as possible. A CNN-based lossless coding method should be developed. The quantized coefficients in the high frequency channels could be coded by performing a vector quantization. A CNN-based codebook generation method should be developed.

The aim of this thesis is to prove the feasibility of implementing CNN methods in image coding. A CNN-based image compression scheme exploiting the dynamics of the parallel CNN computing structure should be developed. Computational intensive operations involved in an image compression process should be realized by CNN calculations as far as possible. The performance of the proposed method should be comparable to the modern image compression standard, i.e. the JPEG 2000 (much better than the JPEG standard). By taking advantage of CNN dynamics, the computing complexity of coding systems to be developed should be reduced significantly as compared to conventional systems.

1.2. Organization of the thesis

In this thesis, firstly the concepts of CA and CNN in chapter 2 will be presented. Subsequently, in chapter 3 the theory of image compression and wavelet transform will be introduced, and different kinds of image compression techniques will be described. In chapter 4, Dogaru's method will be reviewed and an overview of the coding methods to be developed will be given. The method of computing a wavelet transform on a CNN platform will be discussed in detail in chapter 5. The quantization strategy is explained in chapter 6. In chapter 7 and 8, coding methods for quantized coefficients in the low and high frequency channels are discussed in extenso. An evaluation of the performance of the proposed method in this thesis and a comparison to results obtained by applying modern compression standards will be given in chapter 9. In the last chapter, conclusions and perspectives will be outlined.

Chapter 2

Cellular Automata and Cellular Neural Networks

2.1. Overview of Complex Systems

As a new scientific discipline complex systems have been intensively investigated in recent years. The interaction between elements, parts and the entire system is the foundation of complex systems. The primary topic in the research on complex systems is how the complexity of the entire system is associated with the complexity of the parts [Bar03].

From empirical observation, three cases of the relationship about the complexity of the whole system to the individual parts can be summarized [Hal08].

One case is that a system having complicated collective behaviour is composed of parts, which have complicated behaviour as well. It is intuitive that a collection of parts with complicated behaviour would result into a complex system.

The second case is that a system having complex collective behaviour is composed of parts, which have simple behaviour yet. In such a system, the collective behaviour of entire system is more than the sum of individual behaviours of their parts, and thus the behaviour of the whole system is difficult to expect. This system is said to have *emergent complexity* [Cor02]. There exist a huge number of instances of this kind of systems in the real world. The understanding, modelling and simulation of these systems have a special meaning for the research on complex systems [Bar03]. Therefore, the investigation of emergent complexity draws a great deal of attention.

The third case is that a system is composed of parts with complex behaviour while shows simple collective behaviours. This system is called to have *emergent simplicity*.

2.2. Cellular Automata

As mentioned in the above section, the concept of emergent simplicity/ complexity is fundamental in the field of complex systems. Cellular automata (CA) provide an approach to model and demonstrate the phenomena of emergent complexity. The essential element of CA is called *cell*. Each cell is located on a regular spatial grid and has a finite number of states. The cell-to-cell influence relies upon local interactions. An initial state (at time t=0) is assigned to each cell. The states of the cells evolve according to particular fixed rule. Typically, the evolving rule is identical for each cell, remains invariant over time, and is applied to the whole cells simultaneously (space/ time-invariant). Both space and time are discretized.

The concept of CA traces back to the 1940s, when it was introduced by John von Neumann and Stanislaw Ulam. The first system of CA was developed by John von Neumann to build a self-replicating machine [Neu51], [Neu66]. In the work of Stanislaw Ulam, graphical constructions on the evolution of states governed by simple rules were presented. It was noticed that the state updating rule of CA allows to generate complex and graceful patterns and that some of these patterns could self-reproduce. Extremely simple rules permit to create very complex patterns [Ulm50].

In the 1970s, the research in CA received an enormous boost in popularity thanks to the introduction of the *Game of Life*, a two-dimensional CA developed by John Horton Conway [Gar70]. It has been proven that the Game of Life can emulate a universal Turing machine and thus, it has the capability of universal computation. In later years, an increasing number of CA applications to applied sciences were investigated [Sch08].

Another monumental work of CA is made by Stephen Wolfram. In his work *A New Kind of Science*, he argued that “the discoveries in the field of CA are not isolated mathematical artefacts but have significance for all disciplines of science” [Wof02].

Stephen Wolfram named the one-dimensional CA as elementary CA, where there are two possible states (usually labelled as 0 and 1) and the rule to determine the state of a cell in the next generation depends only on the current state of the cell and of its two immediate neighbours. The cell and its two neighbours build a three cell-neighbourhood. There are totally $2^3=8$ possible patterns for a neighbourhood. These eight possible patterns have then $2^8=256$ possible rules.

neighbourhood	111	110	101	100	011	010	001	000
G	g_7	g_6	g_5	g_4	g_3	g_2	g_1	g_0

Table 2.1: General rule of one-dimensional CA

The general rule of one-dimensional CA is given in Table 2.1. For each possible pattern for a neighbourhood, the state of the middle cell in the next generation is denoted as g_i. With the definition $G = [g_7, g_6 \ldots g_0]$, G is a binary string, often represented as a decimal number, denoted ID. Each rule of the CA can be denoted as Rule *ID*.

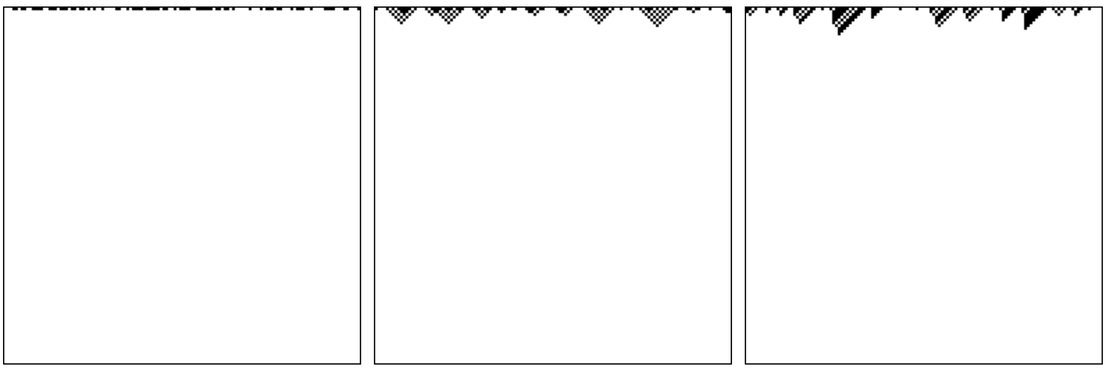
Stephen Wolfram sorted the CA into four classes according to their complexity of behaviour [Wof02]. Some examples are shown in Figure 2.1, where each example gives a two-dimensional representation of the evolution of a CA. The top row corresponds to the initial state, which is a uniformly random pattern. The evolution over the following 255 steps is displayed in row 2 to 256.

Class 1: Independent of initial state of cells, nearly all cells evolve quickly into a stable, homogeneous state. No randomness can be found. From the examples in Figure 2.1, with Rule 0, 160 and 168, even beginning from a random pattern, after a small number of steps, all random patterns disappear.

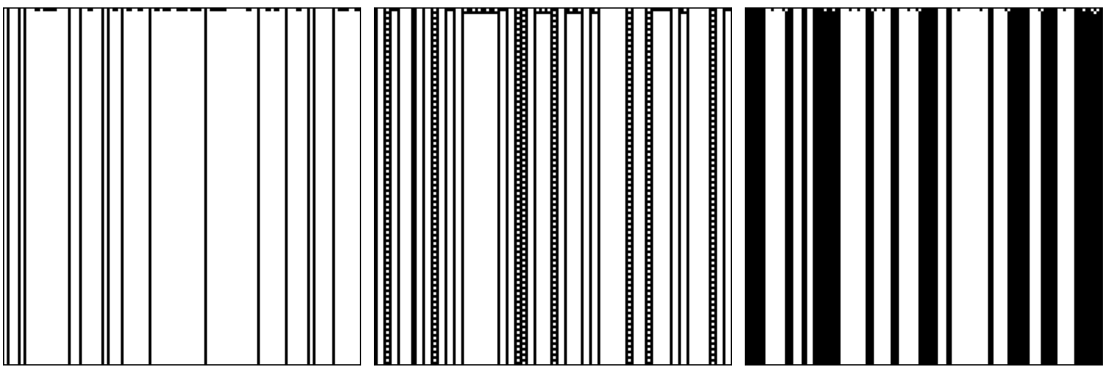
Class 2: Nearly all cells evolve quickly into stable (e.g. Rule 4, 232 in Figure 2.1) or oscillating (e.g. Rule 108 in Figure 2.1) state.

Class 3: Nearly all cells evolve in a pseudo-random or chaotic manner. Any stable structure is quickly destroyed by the surrounding noise.

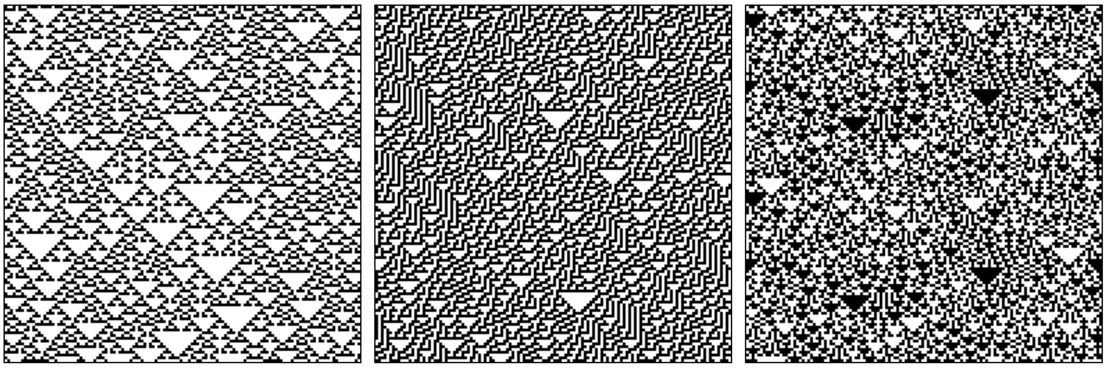
Class 4: Nearly all cells advance into structures that interact in a manner, with the local structures that can continue for a long time.



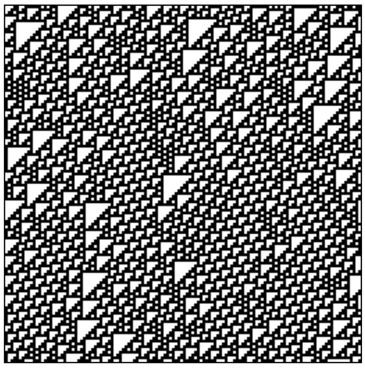
Examples of class 1: Rule 0 160 168



Examples of class 2: Rule 4 108 232



Examples of class 3: Rule 22 30 150



Examples of class 4: Rule 110

Figure 2.1: Four classes of CA (images are generated by Matlab)

Wolfram's work on CA [Wof02] is mainly based on some empirical observations from computer simulations, while Leon Chua presented a comprehensive mathematical analysis. In his works [Chu06] [Chu07] [Chu09] [Chu11] [Chu12] [Chu13], by introducing concepts of nonlinear dynamics and attractors, some fuzzy concepts in Wolfram's work can be defined and justified through mathematical analysis. The local rule of CA expressed as a true table in Wolfram's work are mapped into scalar ordinary differential equations. Hence, the collective behaviour of CA can be analysed by applying methods in the theory of nonlinear differential equations instead of empirical observations.

When the spatial structure of CA extends to two-dimensions, with the increasing number of involved cells, the behaviour becomes more complicated. There are many possible neighbourhood structures. Two of the most common are a von Neumann neighbourhood and a Moore neighbourhood (see Figure 2.2, 3×3 neighbouring structure, the state of the circled cell depends on state values of the grey cells).

von Neumann neighbourhood Moore neighbourhood

Figure 2.2: Von-Neumann and Moore neighbourhood

In two-dimensional case, the number of possible CA becomes incredibly huge (2^{32}=4294967296 with von Neumann neighbourhood and $2^{512}=1.34 \times 10^{154}$ with Moore neighbourhood), hence a thorough exploration of the two-dimensional CA properties is realistically unfeasible.

A particular type of CA is the *totalistic* CA. The next state of a cell relies on the sum of the states of its neighbouring cells. If its current state is considered additionally, then the CA are properly called *outer*-totalistic, or *semi*-totalistic. Game of Life is an example of an outer-totalistic with binary state values. Similar to the definition of the general rule in one-dimensional CA in Table 2.1, the general rule of semi-totalistic CA can be given as shown in Table 2.2.

G	g_9	g_8	g_7	g_6	g_5	g_4	g_3	g_2	g_1	g_0
α	4	3	2	1	0	4	3	2	1	0
β	1	1	1	1	1	0	0	0	0	0

Table 2.2: Rule of semi-totalistic CA (considering only binary valued neighbourhood)

α denotes the sum of the states of the neighbouring cells, β is the current state of the central cell. $G = [g_9,g_8,\ldots g_0]$ is a binary string, often represented as a decimal number, denoted ID. For semi-totalistic CA with a von Neumann neighbourhood, there are totally five possible values of α and two states of β, hence ten possible combinations of α and β altogether. These ten possible combinations have then 2^{10}=1024 possible rules, therefore, the value of the ID ranges between 0 and 1023.

All the above CA are called deterministic since the state of a cell can be uniquely determined by the states of the neighbouring cells through some predefined rules. As an important extension, in stochastic or probabilistic/random CA, the updating rule of these CA is stochastic, i.e. the next state of a cell is chosen according to some probability distribution of a random experiment. The state updating rule of a cell depends on the product of the state probability of neighbouring cells. Despite the simplicity of the updating rules in stochastic CA complex behaviour e.g. self-organization may emerge [Gac86], [Gut91].

Due to the extreme simplicity of construction and abundant emergent complexity, CA are often considered in the analysis and modelling of complex systems. A large number of CA implementations are addressing simulations of phenomena in biology, chemistry, physics and economics. Some classical models based on CA were proposed, e.g., Lovelock's Daisyworld model in ecology [Ack03], disease spreading model in Epidemiology [Fu04], [Mag04], forest fire model [Kar97], [Cho98], plant infection model [Sch95], [Har94] etc.

2.3. Cellular Neural/Nonlinear Networks

Before the introduction of Cellular Neural/Nonlinear Networks (CNN), it is helpful to review the concept of neural networks shortly.

2.3.1 Neural Networks

The term *neural networks* is often referred to artificial neural networks that are inspired by biological neural systems.

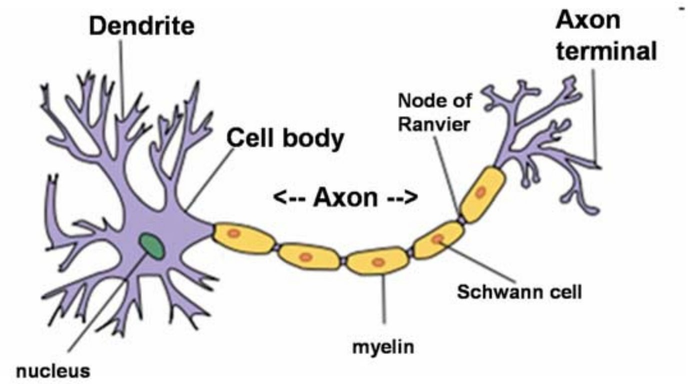


Figure 2.3: Neuron taken from （image taken from [htt00])

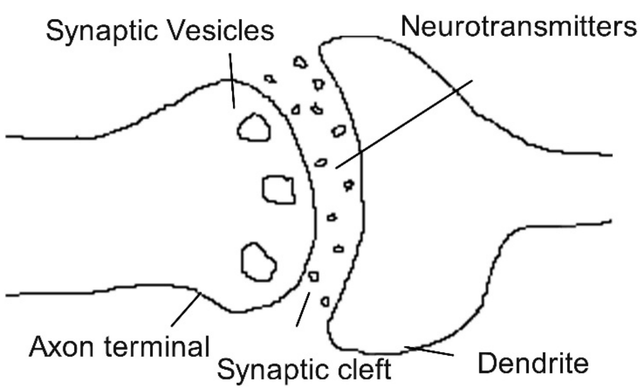


Figure 2.4: Mechanism of synaptic dynamics (reproduced, image taken from [htt01])

In biological neural systems, the fundamental information processing unit is called *neuron* or *nerve cell* shown in Figure 2.3. A neuron consists of *dendrites* (inputs), a *cell body* and an *axon* (output). There are usually more than one dendrites in the neuron, while there is only one axon. Synapses are the junctions where neurons pass signals to other neurons, or muscle cells [Lod07]. A neuron receives inputs from other neurons. Once the total sum of inputs exceeds a critical level (threshold), the neuron discharges an electrical pulse called spike that is transferred from the axon to the next neuron via a synaptic path. An illustration of the mechanism of synaptic dynamics is shown in Figure 2.4

Inspired by the above described mechanism of the signal processing in a biological neuron, a simplified model, which takes only a few features of each neuron and some neuron-to-neuron interactions into account, was designed.

Neural networks are a form of multiprocessor computing systems, characterized by [htt02]

- simple processing units,

- a high degree of interconnection, and
- adaptive interaction between units.

The first mathematical model of neural networks was developed by W. McCulloch and W. Pitts in 1943 [McC43].

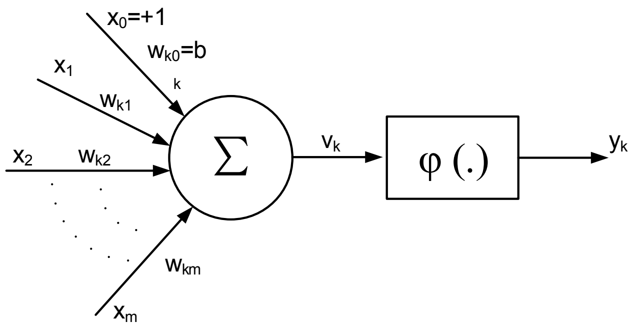


Figure 2.5: Artificial neuron

In an artificial neuron (see Figure 2.5), there are m+1 inputs x_0, x_1 … x_m and weight w_0, w_1,.. w_m. The value +1 is assigned to input x_0 and w_{k0} is a bias value *b*. The output of the k^th^ neuron is calculated as

$$y_k = \varphi(v_k) = \varphi\left(\sum_{j=0}^{m} w_{kj} x_j\right), \tag{2.1}$$

where φ is an activation function, which could have a number of forms, e.g. a step function, a linear function, a sigmoid function and so on.

There are various kinds of neural networks, which can be roughly classified into

- Feed-forward networks: signals flow only one way (from input to output). No feedbacks (loops) exist.
- Feed-back networks (recurrent networks): signals can flow in both directions by introducing loops in the network. The dynamic of feed-back networks could be extremely complicated. Their state evolves continuously from an initial state until reaching an equilibrium point. If the inputs are changed later, the networks shall resume evolution until a new equilibrium point is reached.

The simplest neural network is called *perceptron* originally proposed by F. Rosenblatt in 1958 [Ros62]; it is formed by a single layer of neurons. A perceptron is a feed-forward model: all the inputs are fed into the neurons, and then processed before being transferred to the output.

A perceptron consists of one layer of neurons only. The output obtained by Eq. 2.1 is a function of a linear combination of inputs and thus a perceptron can solve linearly separable problems barely. Hence, the computation capability is limited. A more powerful form of a feed-forward neural network is a system of neurons, which are distributed in different layers. This type of neural network is called *multilayer perceptron* shown in Figure 2.6. In this kind of networks, the neurons that receive inputs build up an *input layer*, while the neurons generating final outputs of the overall network construct an *output layer*. Between the input and output layer, the middle layers are named as *hidden layers*.

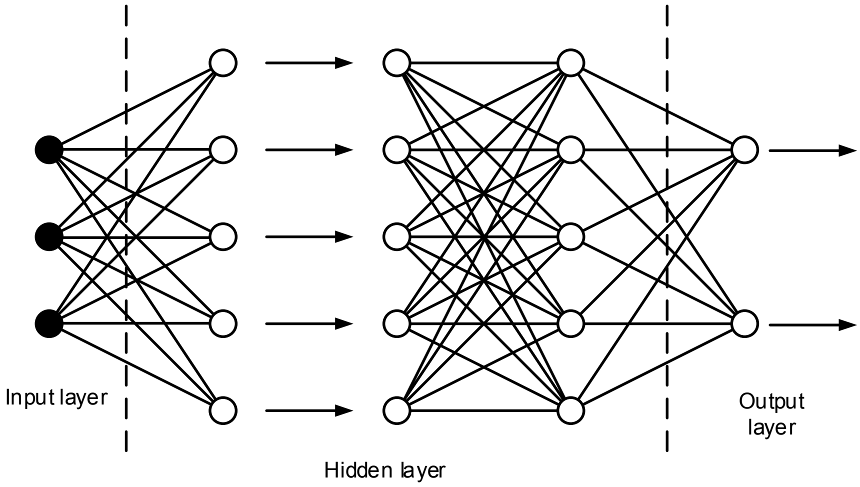


Figure 2.6: Model of multilayer perceptron

From Figure 2.6, it is apparent that each neuron in one layer has connections to all the neurons in the next layer (*full-connection*).

Being an adaptive system, a neural network possesses a configurable internal structure, which may be varied by adjusting the weights of the connections between the neurons. This property is the basis of the learning ability of neural networks. There are several strategies for learning. Theoretical results indicate that, a network shown in Figure 2.6 can approximate any function to any required degree of accuracy, provided that it contains an adequate number of hidden units [Hay08]. This makes neural networks very useful in the field of artificial intelligence. Many applications based on neural networks have been developed, e.g. pattern recognition, time series prediction, signal processing, control, anomaly detection and so on. [Zha00] [Rot90] [Sap09] [Don95].

2.3.2 CNN Paradigm

The concept of Cellular Neural/ Nonlinear Networks were firstly introduced by L. Chua and L. Yang in 1988 [Chu88]. According to a later given definition of Chua: "A CNN is any spatial arrangement of locally-coupled cells, where each cell is a dynamical system that has an input, an output and a state evolving in accordance with some prescribed dynamical laws" [Chu98]. For an isolated cell in a two-dimensional CNN, the general form of a dynamical law can be written as

$$\dot{x}_{ij} = f(x_{ij}, u_{ij}, z_{ij}). \quad (2.2)$$

A basic model of a CNN by introducing coupling to the neighbourhood cells, which is called a standard CNN, can be defined by

$$\dot{x}_{ij} = -x_{ij} + \sum_{kl \in S_{ij}(r)} A(i,j;k,l) y_{kl} + \sum_{kl \in S_{ij}(r)} B(i,j;k,l) u_{kl} + z_{ij}, \quad (2.3a)$$

$$y_{ij} = f(x_{ij}) = \frac{1}{2}(|x_{ij}+1| - |x_{ij}-1|),$$
$$i = 1,2,\ldots,M,\ j = 1,2,\ldots,N. \tag{2.3b}$$

where x_{ij}, u_{ij}, y_{ij} and z_{ij} are *state*, *input*, *output* and *threshold (offset)* of a cell C_{ij}, respectively A(i,j;k,l) and B(i,j;k,l) are called *feedback* and *input synaptic* operator (weight functions). Each CNN cell C_{ij} is coupled locally only to the neighbour cells that are inside a prescribed sphere of influence $S_{ij}(r)$ of radius r, where

$$S_{ij}(r) = \{C_{kl} : \max(|k-i|, |l-j) \le r, 1 \le k \le M, 1 \le l \le N\} \tag{2.4}$$

In the general case, A(i,j;k,l), B(i,j;k,l) and z_{ij} could change with their position (i,j) and time. In most widely used standard CNN with linear coupling, A(i,j;k,l), B(i,j;k,l) and z_{ij} are space and time invariant (translation invariant). Hence, A(i,j;k,l) and B(i,j;k,l) are simplified into a_{kl} and b_{kl} respectively, only a 3×3 neighbourhood is considered and then CNN is defined as

$$\dot{x}_{ij} = -x_{ij} + \sum_{kl \in S_{ij}(r)} a_{kl} y_{kl} + \sum_{kl \in S_{ij}(r)} b_{kl} u_{kl} + z, \tag{2.5}$$

where a_{kl} and b_{kl} are scalars called *feedback* and *input synaptic* weights. Above standard CNN is uniquely defined by 19 real numbers (r=1, a uniform bias z_{ij}=z, nine feedback synaptic weights a_{kl}, and nine input synaptic weights b_{kl}). These 19 real numbers are named as *CNN template* or *CNN gene*.

In Eq. 2.3b, the output function $y_{ij}=f(x_{ij})$ is a piecewise linear function (see Figure 2.7), but any function from a large family of sigmoid functions can be assumed instead (see Figure 2.8).

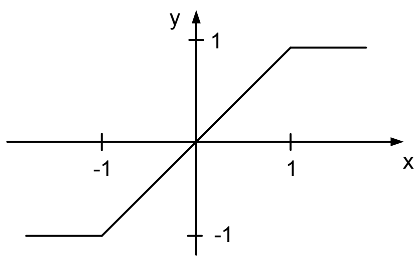


Figure 2.7: Output function (piecewise linear form in Eq. 2.3b)

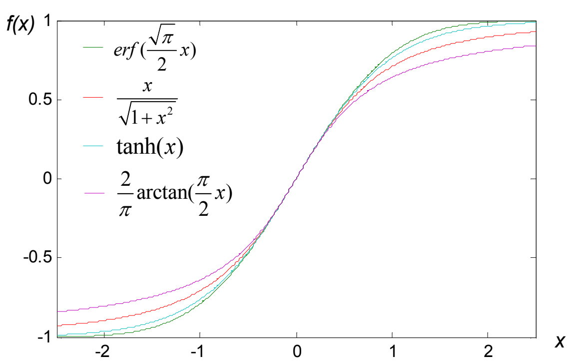


Figure 2.8: Sigmoid functions

CNN were designed as a circuit-oriented architecture since their first introduction in the late 1980s. In [Chu88], a cells circuit of a standard CNN, corresponding to the state equation in Eq. 2.3 and the data flow chart in Figure 2.9, as shown in Figure 2.10 is introduced. In this circuit, C denotes a linear capacitor; R_x denotes linear resistor; I denotes an independent current source; $I_{xu}(i, j; k, 1)$ and $I_{xy}(i, j; k, l)$ are linear voltage-controlled current sources with the characteristics $I_{xy}(i, j; k, l) = A(i, j; k, l)$ and $I_{xu}(i, j; k, l) = B(i, j; k, l)$ for each cell in the neighbourhood; $I_{yx} = (l/R_y)f(v_{xij})$ is a piecewise-linear voltage-controlled current source with its characteristic $f(\cdot)$ as shown in Eq. 2.3b; E_{ij} is an independent voltage source.

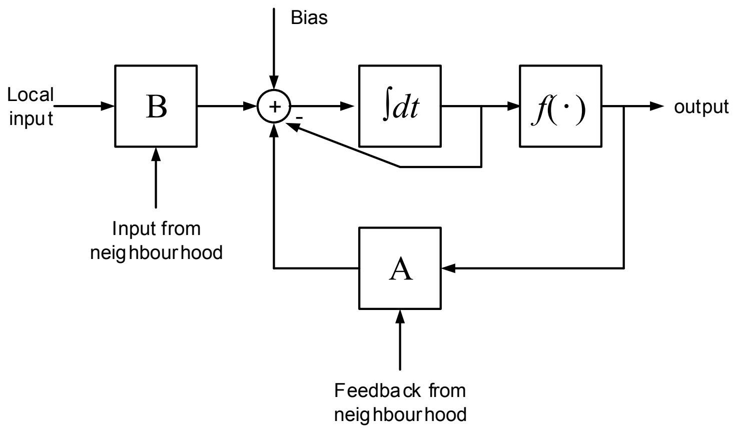


Figure 2.9: Data flow chart of CNN

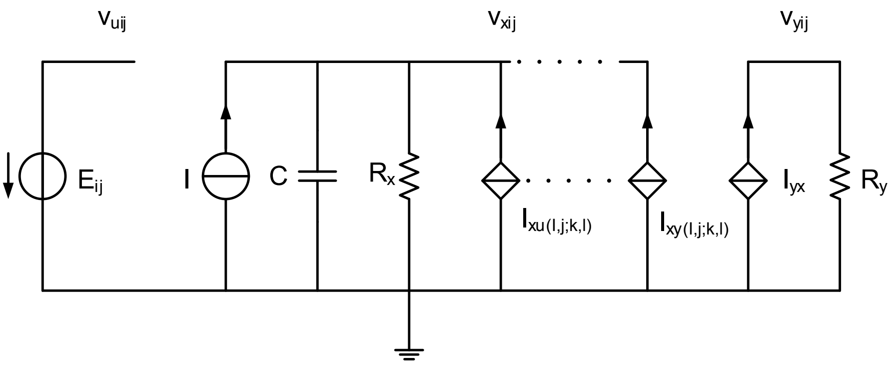


Figure 2.10: Circuit of a standard CNN cell (taken from [Chu88], reproduced)

When all b_{kl} in Eq. 2.5 are zero, i.e. the cells have no inputs, the CNN is called *autonomous* CNN. This type of CNN can exhibit various complex phenomena, e.g. pattern formation, autowaves and so on. If all a_{kl} are zero, except the central coefficient a_{00}, this kind of CNN is named *uncoupled* CNN. Eq. 2.5 reduces to

$$\begin{aligned}\dot{x}_{ij} &= -x_{ij} + a_{ij} f(x_{ij}) + w_{ij} \\ &= g(x_{ij}) + w_{ij} \\ &\triangleq h(x_{ij}, w_{ij}),\end{aligned} \tag{2.6}$$

where

$$w_{ij} = \sum_{kl \in S_{ij}(r)} b_{kl} u_{kl} + z,$$

$$\begin{aligned}g(x_{ij}) &\triangleq -x_{ij} + a_{ij} f(x_{ij}) \\ &= -x_{ij} + \frac{1}{2} a_{ij} [|x_{ij} + 1| - |x_{ij} - 1|].\end{aligned}$$

It is observed the shape of function $h(x_{ij},w_{ij})$ is only dependent on the self-feedback coefficient a_{ij} and the offset level w_{ij}. Its final state can be explicitly determined. An exhaustive analysis to the final state of Eq. 2.6 is presented in [Chu98].

Comparing now the definition of CNN to that of CA and neural networks given in the previous section, CNN are closely related to those systems. CNN inherit the local connectivity and cellular structure from CA, thus they could be regarded as a time-continuous version of CA, in which the states assume continuous values and the evolution rule for the states of the cells is described by certain ordinary differential equations. Similar to CA, CNN can be used to prove theories or model some physical, biological processes. Additional, CNN were intended from the very beginning to be a practical signal processing paradigm. From the perspective of neural networks, CNN could be considered as a locally connected version of a neural network. Due to the fully connected structure of neural networks, their hardware realization is not easy. In fact, even nowadays most of the applications of neural networks are still based on software simulations. In contrast, the locally connected structure of CNN simplifies their hardware realization.

In the standard CNN defined in Eq. 2.3, the input u_{kl} and the output y_{kl} of each neighbour C_{kl} are coupled by feedforward synapse b_{kl} and feedback synapse a_{kl} respectively. Here the weighted sum $B(u_{ij})$, $A(y_{ij})$ are linear combinations, where

$$B(u_{ij}) = \sum_{kl \in S_{ij}(r)} b_{kl} u_{kl}, \text{and } A(x_{ij}) = \sum_{kl \in S_{ij}(r)} a_{kl} y_{kl}. \tag{2.7}$$

If generalizing the $B(u_{ij})$ and $A(y_{ij})$ to polynomial functions, the so-called *polynomial type CNN* can be defined with:

$$\begin{aligned}\dot{x}_{ij} &= -x_{ij} + \sum_{kl \in S_{ij}(r)} a_{kl}(y_{kl}) + \sum_{kl \in S_{ij}(r)} b_{kl}(u_{kl}) + z_{ij}, \\ a_{kl}(y_{kl}) &= \sum_{d=1}^{D} a_{kl,d} \cdot {y_{kl}}^{d}, \\ b_{kl}(u_{kl}) &= \sum_{d=1}^{D} b_{kl,d} \cdot {u_{kl}}^{d} \\ & i = 1,2,\ldots,M, j = 1,2,\ldots,N.\end{aligned} \tag{2.8}$$

This kind of CNN was firstly introduced in [Puf96] and the stability was analysed in [Cor02]. By introducing polynomial terms, this kind of CNN can represent nonlinear system in a wider range. It can be applied for elementary non-linearly separable problems, e.g. XOR operation

[Gom06], to the analysis of human brain activity, e.g. the epilepsy seizures prediction problem [Nie05].

For some applications, a discrete time version CNN can be useful. A. Nossek introduced the *Discrete-Time Cellular Neural Networks* (DT-CNN). A translation-invariant DT-CNN is defined as [Har92]:

$$\begin{aligned} x_{ij}(k) &= \sum_{kl\in S_{ij}(r)} a_{kl} y_{kl}(k) + \sum_{kl\in S_{ij}(r)} b_{kl} u_{kl}(k) + z, \\ y_{ij}(k) &= f(x_{ij}(k-1)) = \begin{cases} 1 & for\ x_{ij}(k-1) > 0 \\ -1 & for\ x_{ij}(k-1) < 0, \end{cases} \end{aligned} \tag{2.9}$$

CNN hardware

The motivation of the CNN invention was to develop a more practical network architecture to replace fully connected classical neural networks, which can be hardly realized in hardware. The number of wires and circuitry used to connect each cell to every other cells in a fully-connected neural network increases dramatically with the number of cells. In contrast, thanks to the local connected structure, in CNN, the calculation and interconnection exist only within a prescribed sphere of influence [Chu88] [Chu98]. Thus a CNN is conceptually suitable for hardware implementation.

Analog-input analog-output CNN chips called CNN universal machine chips (CNN Universal Machine, abbr. CNN-UM) have been developed over the years [Ros93] [Ros00]. A series of CNN templates organized as a CNN subroutine can be executed on the chip with an incredibly high speed. One of the first CNN-UM chips was the ACE400 composed of 20×22 cells [Dom97]. After that, the second generation the ACE4k chip with a resolution of 64×64 cells has been developed [Lil00]. In 2003, a new generation the ACE16k with 128×128 cells has been released [Lin02]. In recent years, a number of CNN-based visual analog processors have been realized, such as Q-Eye [Rod08], MIPA 4K [Poi09] and VISCUBE [Zar10].

All above introduced CNN-UM chips are ASICs (Application Specific Integrated Circuit), another possibility of CNN hardware realization is using an FPGA platform. Due to the considerable flexibility of FPGA solutions, an FPGA-based CNN implementations draw more and more attention. In [Nag03] an emulated digital multilayer CNN-UM chip architecture called *Falcon* was introduced by Z. Nagy and P. Szolgay. In recent years, some work about more general type of CNN on FPGA platform have been developed by J. Mueller et al. in [Mul12] [Bra13]. N.Yildiz et al. proposed a fully pipelined Real-Time CNN architecture implemented on FPGA platform capable of processing full-HD video streams [Yil14]. G. Borgese et al. presented a DCMARK system based on CNN paradigm used to investigate complex physical dynamics by solving partial differential equations. The system is implemented on FPGA [Bor13].

After the launch of general purpose computing by graphics processing units (GPUs), some works of design a CNN simulator using GPU have been suggested [Dol09] [Her08].

Thanks to the parallel nature and locally connected structure, CNN are considered to be appropriate for image processing especially. A large class of CNN templates have been developed for the image processing tasks (see [Cel07]). In this thesis, for the purpose of image compression, different CNN architectures are proposed and investigated.