8-2001

Stability in N-Layer recurrent neural networks

Rodica Ion Waivio

To the Graduate Council:

I am submitting herewith a thesis written by Rodica Ion Waivio entitled "Stability in N-Layer recurrent neural networks." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

Bruce Whitehead, Major Professor

We have read this thesis and recommend its acceptance:

Kenneth Kimble, Roy Joseph

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Rodica Ion Waivio
entitled "Stability in N - Layer Recurrent Neural Networks." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

Bruce Whitehead

Dr. Bruce Whitehead, Major Professor

We have read this thesis
and recommend its acceptance:

K R Kunkle

Roy D Joseph

Accepted for the Council:

Vice Provost and Dean of Graduate Studies

Stability in N-Layer Recurrent Neural Networks

A Thesis
Presented for the
Master of Science Degree
The University of Tennessee, Knoxville

Rodica Ion Waivio
August, 2001

Acknowledgments

I would like to thank everybody who contributed to the completion of this work in one way or another. Special thanks goes to my advisor, Dr. Bruce Whitehead who offered me support and coordination. Special thanks to Dr. Kenneth Kimble and Dr. Roy Joseph for their support and encouragement. I deeply appreciate the knowledge gained during study for the master's degree. I strongly appreciate the specialization in Computer Science offered with coordination and help by Dr. Bruce Whitehead. A strong specialization in Computer Visualization I owe to Dr. Kimble.

Abstract

Starting with the theory developed by Hopfield, Cohen-Grossberg and Kosko, the study of associative memories is extended to N - layer recurrent neural networks. The stability of different multilayer networks is demonstrated under specified bounding hypotheses. The analysis involves theorems for the additive as well as the multiplicative models for continuous and discrete N - layer networks. These demonstrations are based on continuous and discrete Liapunov theory. The thesis develops autoassociative and heteroassociative memories. It points out the link between all recurrent networks of this type. The discrete case is analyzed using the threshold signal function as the activation function. A general approach for studying the stability and convergence of the multilayer recurrent networks is developed.

Table of Contents

List of Figures

NOMENCLATURE

A_k stimuli

b or θ threshold

B_K response

c, α constant real value

$D = z^{-1}$ unit delay operator, originated from the z-transform

e error

F_X, F_Y layers

$f()$ functions of differential or difference equations

$F()$ general type of function

$g()$ the function of heteroassociative recall mechanism

i, j index

h hidden layer

k discrete time or the k-th layer in a N layer network

k_0 initial condition

L the number of PEs, neurons in the hidden layer

M function of the differential or difference equation

m number of pairs in a pattern

N number of layers in multilayer recurrent neural network

n number of neurons of F_X layer

n^k number of neurons of the k-th layer from N layers

n_1 number of neurons of the first layer

n_2 number of neurons of the second layer

n_3 number of neurons of the third layer

o derivative order

p number of neurons of F_Y layer

r constant value

q index in summing expresion

$S, \varphi()$ activation function

S_i^X denotes the activation function of the i-th neuron in the layer F_X

$s(k, k_0, .)$ the solution evaluated at the k-th time ($k \geq k_0$)

$s(t, t_0, x_0)$ denotes the solution of the differential equation

$t, t1, t_0$ times

t_s sampling time

$T(M, \epsilon)$ time estimation

V Liapunov candidate function or system's energy

v inner product of the weight and input vectors

x input vector of the neural network

y output signal

$y(n+1)$ is the future value of the vector y

$y(n)$ is the current value of the vector y

Z_+ the set of nonnegative integers

W weight matrices of the neural networks

$W^h or Wh$ weight matrices of the hidden layers

W_Y layer output weight matrix

α function

$\delta(\epsilon), \delta(\epsilon, t_0)$ equilibrium sequential estimations

η the constant adaptation rate

Abbreviations

ABAM adaptive bidirectional associative memory model

BAM bidirectional associative memory model

C^1 set of functions continuously differentiable

lpdf locally positive definite function

pdf positive definite function

PE processing element, neuron

RABAM random adaptive bidirectional associative memory model

Chapter 1

Introduction

1.1 Background

A neural network is a massively parallel distributed processor made of simple processing units known as neurons. Knowledge is acquired by the network from its environment through a learning process. The inter-neuronal connection strengths, known as synaptic weights, are used to store the acquired knowledge. Recurrent neural networks compute their future state based on the present or past states and outputs. Neural networks with recurrent connections are finding increasing applications in diverse areas of electronics and engineering. Past efforts at designing such networks have focused mainly on steady-state fixed point behavior for applications such as associative memories. These networks can also solve dynamic problems such as the recognition of continuous signals and adaptive dynamic control. Their diversity in topology as well as their rapid convergence and stability make them powerful tools. This thesis proposes to generalize the global stability of associative memories, including both additive and multiplicative models for any number of layers.

Research on associative memories dates back to the early work of Cohen and Grossberg [24]. More recently, the study of bidirectional associative memories has been extended by Kosko [1], [2], [3], [4]. The problem to be addressed by this thesis is to

demonstrate the generality of associative memories to multilayer architectures, under different topologies. The analysis is based on continuous and discrete Liapunov theory.

This thesis is composed as follows. Chapter 2 presents a general overview of neural networks. Chapter 3 discusses the main theorems and results of stability theory. Chapter 4 is devoted to multilayer continuous recurrent neural networks. Section 4.2 defines multilayer recurrent neural networks (as generalizations of Cohen-Grossberg models [23]) and establishes the basic mathematical equations for the additive and multiplicative multilayer models. Section 4.3 introduces and proves the theorems contributed by this thesis including the Continuous N - Layer Associative Memory Theorems - for both additive and multiplicative models. Section 4.4 states the lemma regarding the relationship between heteroassociative and autoassociative networks. Chapter 5 presents the theorems and the proofs of discrete N layer associative memory models. Chapter 6 presents the conclusions.

1.2 Review of Literature

The bidirectional associative memory (BAM) model introduced by Kosko in [1], [2], [3], [4] as an extension of the unidimensional Hopfield autoassociator model [16], has been investigated extensively over the past ten years. In [4], Kosko develops a theoretical analysis of discrete and continuous bidirectional associative memories. He investigates the convergence and stability of different recurrent neural networks. The theorems regarding BAM and their relation to Hopfield and Grossberg models [16], [24] are analyzed as heteroassociative neural networks, as adaptive BAMs and as random adap-

tive BAMs. Kosko [4] discuses the global stability of recurrent neural networks as well as the stability-convergence dilemma of adaptive BAMs. The fundamentals presented in chapter 2 and chapter 3 are based on the Cohen-Grossberg Theorem [24], the Cohen-Grossberg-Kosko Theorem [4] and on Liapunov stability [26], [23].

Further results obtained by Kosko [27] for the BAM analyze the structural stability of unsupervised learning for recurrent neural networks using stochastic calculus to derive the random adaptive bidirectional associative memory model, RABAM. Kosko [4] also presents the ABAM Model and ABAM Theorem, and extends the results to the RABAM Theorem and the RABAM Noise Suppression Theorem, concluding that the average RABAM behavior is the ABAM behavior. Similar matters are presented for stochastic models in [22]. In [28] Kosko examines competitive learning systems as stochastic dynamic systems under continuous and discrete models. He derives general expressions of unsupervised, supervised and differential competitive learning systems. An estimation of unknown probability density functions based on random pattern samples is made using adaptive vector quantization (AVQ). In feedforward competitive neural networks, the synaptic vectors quantize the pattern space and converge to pattern class centroids or local probability maxima. Kosko proves this using a stochastic Liapunov procedure under competitive AVQ algorithms. [15] is dedicated to differential competitive learning compared with supervised competitive learning. An extension of feedback neural stabilization theory to feedback fuzzy theory analyzes similar global stability properties in [21].

Chapter 2

Foundations of Artificial Neural Systems

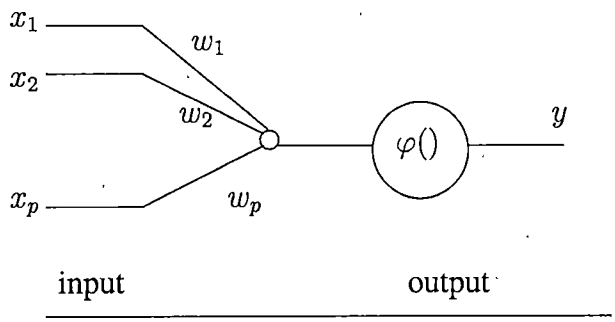
2.1 Introduction

Artificial neural networks are nonlinear information processing devices, built from interconnected elementary neurons. A neural network is a parallel, distributed structure consisting of interconnected processing elements. Each processing element has a single output which is sent to as many other processing elements as desired. The PE's output signal can be any desired mathematical function of the inputs. The processing elements are abstractions of the biological neuron.

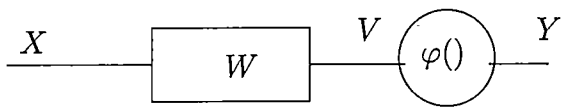
Neural networks can be programmed or trained to store, recognize, and associatively retrieve patterns in order to solve combinatorial optimization problems, control ill-defined problems, and estimate sampled unknown functions.

2.2 Processing Elements, Neurons

Processing elements (PE), referred to as nodes, short-term memory, or neurons are the components of the neural networks. Artificial neural networks are nonlinear information signal processing units interconnecting different architectures and structures of PEs. An artificial neuron, PE is a p-input single-output signal processing element. Graphically, a PE is described in Fig.2.1. The pre-synaptic activities are represented by



A) One neuron



B) Block diagram representation

Fig. 2.1. A single p-input (p-synapse) neuron, PE

the p-input column vector,

$$x = [x_1 ... x_p]^T \tag{2.1}$$

in the p-dimensional input space. The synapses are characterized by the adjustable parameters, called weights or synaptic strength parameters, arranged in a p-element row vector:

$$w = [w_1 ... w_p] \tag{2.2}$$

A synapse is classified as excitatory if the corresponding weight is positive, $w_i > 0$, and as inhibitory if the corresponding weight is negative, $w_i < 0$.

The activation potential is written as a linear combination of input signals and synaptic weight parameters, mathematically as an inner product of the weight and input

vectors:

$$v = \Sigma_{i=1}^{p} w_i x_i = w \cdot x^T \tag{2.3}$$

The total post-synaptic PE activity, the output signal, is computed based on the composition of the activation potential and a chosen activation function, $\varphi()$:

$$y = \varphi(v) \tag{2.4}$$

Mathematically, it is convenient to add an additional parameter called threshold, θ or bias, $b = -\theta$, by constantly fixing one signal input.

2.3 Activation Functions

Many mathematical forms have been used for activation functions. Often, activation functions transform an unbounded input activation, v, into a bounded output signal $S(v)$. Three common activation functions are the linear, binary and sigmoidal functions. The linear function has the equation

$$\varphi(v) = \alpha v \tag{2.5}$$

where α is a constant real value that adjusts the magnification of the PE activation, v. The parameter α controls the slope of the function.

The step function is defined in (2.6) followed by graphical representation in Fig.2.2.

$$y = \varphi(v) = \begin{cases} 1, if, v \geq 0 \\ 0, if, v < 0 \end{cases} \tag{2.6}$$

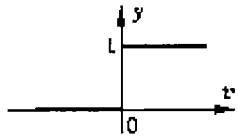


Fig. 2.2. The step function, unipolar case

This processing element is traditionally called a perceptron, and it works as a threshold element with a binary output. The step function for the bipolar case has the following mathematical definition

$$y = \varphi(v) = \begin{cases} 1, if, v \geq 0 \\ -1, if, v < 0 \end{cases} \tag{2.7}$$

The step function with bias, unipolar case, is shown in Fig.2.3. The bias can be added to both the unipolar and the bipolar step functions.

The sigmoidal function is a bounded, monotonic, nondecreasing function that provides a graded, nonlinear response. A common sigmoidal function is the logistic function

$$S(x) = (1 + e^{-cx})^{-1} \tag{2.8}$$

The saturation levels of the equation (2.8) are 0 and 1. The function is strictly increasing for positive constant $c > 0$ because the derivative of S with respect to its argument is

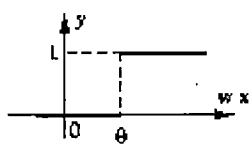


Fig. 2.3. The step function with bias

positive:

$$S' = \frac{dS}{dx} = cS(1 - S) > 0 \tag{2.9}$$

The family of the logistic sigmoidal functions, indexed by c, approaches asymptotically the threshold function as c increases to positive infinity. A bipolar sigmoidal function is the hyperbolic tangent

$$S(x) = tanh(x) \tag{2.10}$$

which has saturation levels at -1 and 1. The derivative with respect to its argument is:

$$S' = c(1 - S^2) > 0 \tag{2.11}$$

for positive constant $c > 0$ and so the function is strictly increasing. The sigmoidal functions can be unipolar or bipolar and they are graphically represented in Fig.2.4 and Fig.2.5.

When a linear function is bounded to the range $[-\gamma, +\gamma]$ it becomes a nonlinear ramp function described by the equation

$$S(x) = \begin{cases} +\gamma \text{ if } x \geq \gamma \\ x, \text{ if } |x| < \gamma \\ -\gamma \text{ if } x \leq -\gamma \end{cases} \tag{2.12}$$

where γ and $-\gamma$ are, respectively, the PE's maximum and minimum output values, commonly referred to as the saturation levels.

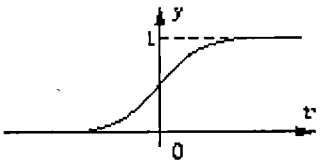


Fig. 2.4. Unipolar Sigmoid

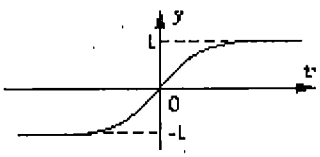


Fig. 2.5. Bipolar Sigmoid

Signal and Activation Velocities The signal velocity $\frac{dS}{dt}$ measures the instantaneous signal change. The chain rule gives $\frac{dS}{dt}$ as

$$\frac{dS}{dt} = \frac{dS}{dx}\frac{dx}{dt} \tag{2.13}$$

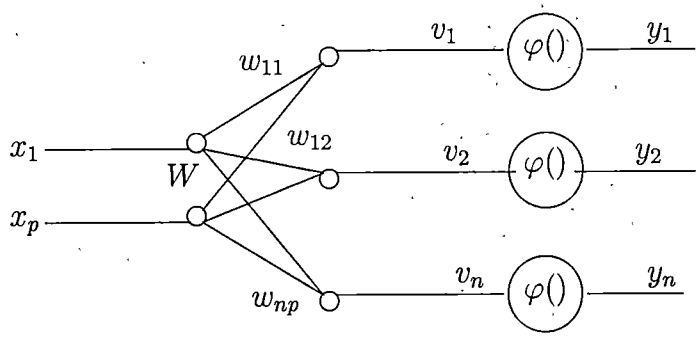
The signal velocity thus depend explicitly on the activation velocity, $\frac{dx}{dt}$.

2.4 A Layer of Neurons

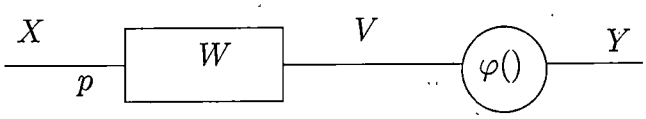
A layer of neurons is a structure composed of PEs which are not connected with each other. It can be basically represented as in Fig.2.6. A layer of neurons is described by a $n \times p$ matrix W of synaptic weights. Every row of the weight matrix is associated with one neuron. Mathematically a layer of neurons can be represented as follows:

$$v = W \cdot X \tag{2.14}$$

$$y = \varphi(W \cdot X) = \varphi(v) \tag{2.15}$$



A) Signal flow graph



B) Block diagram

Fig. 2.6. A n-PEs single layer neural network with p-input

where v is the activation potential vector defined in section 2.2. Figure 2.6 shows that each weight parameter w_{ij} (synaptic strength) is related to the connection between the input signals and the layer nodes.

The (2.3) and (2.4) single-neuron equations can be generalized to a layer of neurons as follows. The input signals come from either the environment or another layer forming an input vector

$$X = (x_1, ..., x_i, ...x_n), \tag{2.16}$$

where x_i is the activity level of the i-th input. The synaptic weights leading into the j-th PE, y_j, forms a vector

$$W_j = (w_{1j}, ..., w_{nj}) \tag{2.17}$$

where the weights w_{ij} are the connection strength from the input x_i to the j-th PE, y_j. Sometimes there is an additional parameter, an internal threshold value, Θ_j, modulated by the weight w_{0j}. The weights W_j, the input vector X and the possible extra parameter Θ_j are used to compute the output value y_j of the j-th PE. Mathematically this operation is defined as

$$y_j = \varphi(X \cdot W_j - w_{0j}\Theta_j) \tag{2.18}$$

or, in point-wise notation, as

$$y_j = \varphi(\sum_{i=1}^{n} x_i w_{ij} - w_{0j}\Theta_j) \tag{2.19}$$

2.5 Multi-Layer Feedforward Neural Networks

If two or more layers are successively connected, they form a multilayer feedforward neural network. Figure 2.7 represents the architecture of a two-layer neural network also known as a single hidden layer neural network. There are L neurons in the hidden layer (hidden neurons), and m neurons in the output layer (output neurons). The input signals which have p neurons, X, are passed through synapses of the hidden layer with connection strengths described by the hidden weight matrix, W_h, and L hidden activation signals are generated. The hidden signals, L, are converted into the output m-dimension activation signals, Y, by means of the output weight matrix, W_Y, composed with the respective activation functions.

2.6 Dynamic Systems

The systems in which the current output signals depend, in general, on current and past input signals are called dynamic systems. There are two classes of dynamic systems: continuous and discrete.

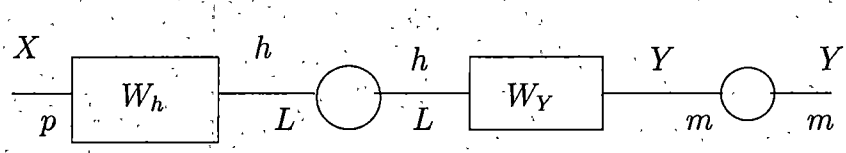


Fig. 2.7. The two layers or the single hidden layer, feedforward neural network

2.7 Continuous Dynamic Systems

In general, continuous dynamic systems work with signals which are continuous functions of continuous variables. Differential equations describe such systems. The first-order differential equation is used in the following form:

$$\frac{dy(t)}{dt} = f(x(t), y(t)) \tag{2.20}$$

where $y(t)$ is the system's output at time t and $x(t)$ is the system's state at time t.

2.8 Discrete Dynamic Systems

Discrete dynamic systems operate with signals which are functions of discrete variables. The discrete model is obtained by discretizating the continuous systems,

$$t = nt_s; t \in R, n \in N \tag{2.21}$$

where t_s is the sampling time. Discrete dynamic systems can be described by discrete difference equations. The first-order difference equations are :

$$y(n+1) = f(x(n), y(n)) \tag{2.22}$$

where $y(n+1)$ is the future value and $y(n)$ is the current value of the vector y. The unit delay operator, $D = z^{-1}$ which originated from the z-transform, is used to model a discrete dynamic system, or to obtain the output signals. Using the delay operator, we have

$$z^{-1}y(n+1) = y(n) \tag{2.23}$$

which leads to the Fig.2.8 in which p is the dimension of the system's state at time n, $x(n)$, and m is the dimension of the system's outputs $y(n), y(n+1)$ at times n and $n+1$.

2.9 Neural Dynamic Systems

A neural dynamic system can be described by a differential equation system that governs the time evolution of the neural activations. For two layer network, we define the state of the neuronal dynamic system, at time t, to be the instantaneous vectors of activations

$$X(t) = (x_1(t), ..., x_n(t)) \tag{2.24}$$

$$Y(t) = (y_1(t), ..., y_p(t)) \tag{2.25}$$

and denote F_X as the first layer whose state at time t is $X(t)$ and F_Y as the second layer whose state at time t is $Y(t)$. For this two layer architecture the activation differential equations are in vector notation

$$\frac{dX}{dt} = g(X(t), Y(t)) \tag{2.26}$$

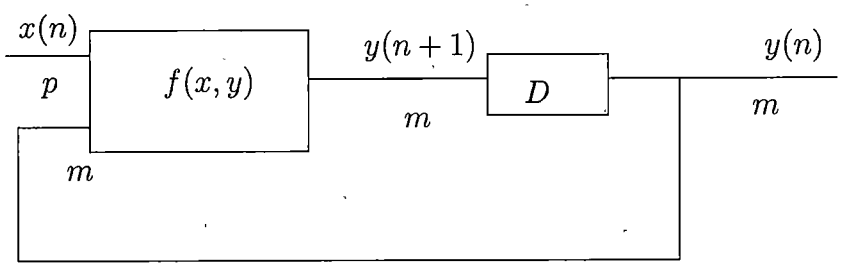


Fig. 2.8. A discrete dynamic system

$$\frac{dY}{dt} = h(X(t), Y(t)) \tag{2.27}$$

or in point-wise notation

$$\frac{dx_1}{dt} = g_1(X(t), Y(t)) \tag{2.28}$$

$$\vdots$$

$$\frac{dx_n}{dt} = g_n(X(t), Y(t)) \tag{2.29}$$

$$\frac{dy_1}{dt} = h_1(X(t), Y(t)) \tag{2.30}$$

$$\vdots$$

$$\frac{dy_p}{dt} = h_p(X(t), Y(t)) \tag{2.31}$$

where x_i and y_j denote the respective states of the i-th neuron in F_X and the j-th neuron in F_Y, g_i and h_j are the activation functions of the i-th neuron in F_X and the j-th neuron in F_Y including the synaptic and input information, without independent variables. This dynamic neural network is an autonomous system. It represents a distinction between neural network models and the classical neural modelling, where often the differential equation gives a detailed, multivariable description of how individual neurons or synapses behave.

Neural State Space We define the state of the neuronal dynamic system, at time t, to be the instantaneous vectors of activations

$$X(t) = (x_1(t), ..., x_n(t)) \tag{2.32}$$

$$Y(t) = (y_1(t), ..., y_p(t)) \tag{2.33}$$

The state spaces of layers F_X and F_Y are the real vector spaces R^n and R^p. The state space of the entire or joint neural dynamic system is the product space $R^n \times R^p$. The trajectory in the state space describing the evolution of the network over time is a smooth curve. The signal state $S(X)$ of the F_X at time t is described as

$$S(X(t)) = (S_1^X(x_1(t)), ..., S_n^X(x_n(t))) \tag{2.34}$$

where S_i^X denotes the activation function of the i-th neuron in the layer F_X. The boundedness of the activation functions implies an n-dimensional hypercube as the signal state space. When the signal functions have values in the unit interval [0,1] the signal space is the unit hypercube I^n, $[0,1]^n$ and the product cube $I_n \times I_p$ defines the signal state space of the two-layer dynamic network F_X, F_Y.

2.10 Topology

Neural network architectures are based on PE's layer organization and weighted interconnections. The topology of a neural network consists of the interconnection scheme and the layer configuration.

2.11 Interconnection Schemes

The three primary PE interconnection schemes [5] are the intra-layer, inter-layer and recurrent connections as represented in Fig.2.9. The intra-layer connections are connections among PEs in the same network layer. The inter-layer connections are connections among PEs from different layers. The recurrent connections are connections from

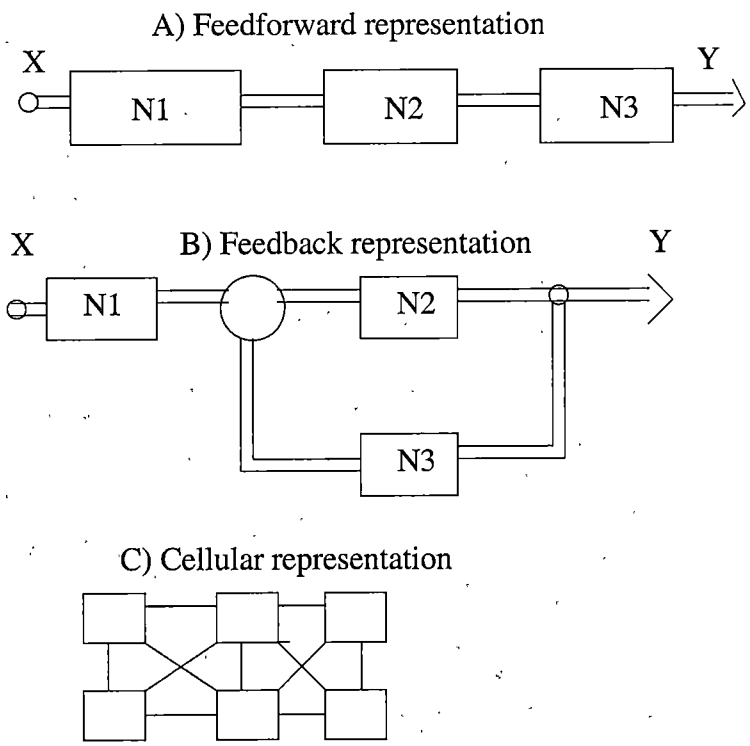


Fig. 2.9. Neural network architectures

a PE to itself. Interlayer connections are classified by their signals. A feedforward signal allows information to flow among PEs in one direction. A feedback signal allows information to flow among PEs in either direction or recursively. Signals that propagate forward are feedforward networks. Signals that propagate from one time step to the next are feedback networks.

2.12 Layer Configuration

The layer configuration combines the neural layers, information flow, and interconnection schemes into a coherent architecture. A layer that receives input signals from the environment is called an input layer and a layer that emits signals to the environment is called an output layer. Any layer lying between the input and the output layers is called a hidden layer and has no direct environment contact.

2.13 Recall

Assume that the pairs (A_k, B_k), $k = 1, 2, ..., m$ have been stored in the learning matrix, W. A heteroassociative recall mechanism is a function $g()$ that takes W (memory) and A_k(stimuli) as input and returns B_K(the response) as output. This relation is illustrated by the equation

$$B_k = g(A_k, W) \tag{2.35}$$

Using this notation, the two primary neural network recall mechanisms can be defined: the nearest-neighbor recall and the interpolative recall. The nearest neighbor recall finds

the stored input that most closely matches the stimulus and responds with the corresponding output. This operation can be conveniently illustrated by the equation

$$B_k = g(A', W) \tag{2.36}$$

where A' is the closest neighbor of A_k, with the lowest $dist(A', A_k)$ denoted by

$$dist(A', A_k) = \min_{q=1}^{m} dist(A', A_q)) \tag{2.37}$$

where $dist()$ is typically the Hamming or Euclidean distance function.

An interpolator is a mathematical function that estimates the outputs based on given data and rules. Interpolative recall accepts a stimulus and interpolates (in a possibly nonlinear fashion) from the entire set of stored inputs to produce the corresponding output. Assuming that the interpolation $g()$ is linear, this operation can be illustrated by the equation

$$B' = g(A', W) \tag{2.38}$$

$$A_p \leq A' \tag{2.39}$$

where A' is an interpolating point between A_p and A_q, with $A' \leq A_q$ and B' is the interpolated response with $B_p \leq B' \leq B_q$ for some pattern pairs (A_p, B_p) and (A_q, B_q).

2.14 Learning

Learning is defined as any change in the memory weight matrix, W, defined mathematically as

$$\frac{dw}{dt} \neq 0 \tag{2.40}$$

Configuration of neural networks as learning units

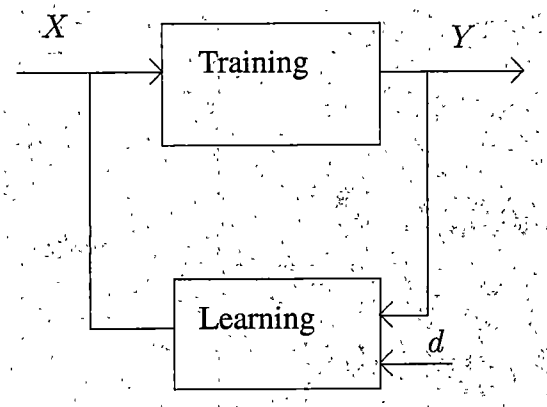


Fig. 2.10. Learning in neural networks

In simple or specialized cases the weight matrix can be pre-computed, but more commonly it is obtained through a learning process. Learning is a dynamic process which modifies the weights of the network in some desirable way. As in any dynamic process, learning can be described in either a continuous or a discrete framework. Generally, a neural network including the learning process can be illustrated by the Fig.2.10.

The learning process can be described either by differential equations in continuous-time

$$\frac{dW(t)}{dt} = M(W(t), x(t), y(t), d(t)) \tag{2.41}$$

or by the difference equations in discrete-time

$$W(n+1) = M(W(n), x(n), y(n), d(n)) \tag{2.42}$$

where x is the input or the system state, y is the output and d is an external teaching/supervising signal used in supervised learning (usually the desired output). The d signal is not present in unsupervised networks. The discrete learning law is often used in the form of the update weight equation

$$W(n+1) = W(n) + \Delta W(n) \tag{2.43}$$

$$\Delta W(n) = M(W(n), x(n), y(n), d(n)) \tag{2.44}$$

All learning methods can be classified into two categories, supervised learning and unsupervised learning, although they may coexist in a given architecture. Supervised learning is a process that incorporates an external teacher or global information. Supervised learning is based on desired output and entails deciding when to turn off the learning, deciding how long and how often to present each training association and how often to supply the performance error. Unsupervised learning, also referred to as self-organization, is a process that incorporates no external teacher and relies upon the local information and internal control. Examples of supervised and unsupervised learning are discussed in the next two sections.

2.15 Supervised Error-Correction Learning

Error-correction learning is a supervised learning procedure that adjusts the connection weights between PEs in proportion to the difference between the desired and computed output values of each PE. If the desired value of the j-th neuron output is d_j and the computed value of the same neuron is y_j then the updating equation takes the general form

$$\Delta w_{ij} = \alpha x_i [y_j - d_j] \tag{2.45}$$

where w_{ij} is the memory connection strength from x_i to y_j, and α is the learning rate, typically $0 < \alpha \ll 1$.

2.16 Unsupervised Hebbian Learning

Hebbian learning adjusts the connection weight based on the correlation over time of the two PE's states or outputs. The simplest mathematical form of Hebbian correlation learning is

$$\Delta w_{ij} = x_i x_j \tag{2.46}$$

where the weight value w_{ij} is the correlation of the i-th PE's state, x_i with the j-th PE's state, x_j, using the discrete time equation Δw_{ij}. This represents the discrete change of w_{ij}. There are several variations of the simple Hebbian learning rule. One such variation is to use the PE's covariance correlation in the equation

$$\Delta w_{ij} = \eta[x_i - \bar{x}_i][x_j - \bar{x}_j] \tag{2.47}$$

where the bracketed terms represent the covariance and $0 < \eta < 1$. Here η is the constant adaptation rate, the overbar represents the mean, and the quantity in brackets represents the j-th PE's variance. Another variation is to correlate x_i's mean value with the variance of x_j as expressed by the equation

$$\Delta w_{ij} = \eta \bar{x}_i[x_j - \bar{x}_j] \tag{2.48}$$

Drive reinforcement learning is a variation which correlates the changes in x_i and the changes in x_j, as expressed by

$$\Delta w_{ij} = (\Delta x_i)(\Delta x_j) \tag{2.49}$$

The discrete time equation, (2.50), correlates the activation value of x_i with the changes in x_j, regulated by the current connection strength value w_{ij} and a constant η

$$\Delta w_{ij} = \eta x_i w_{ij}(\Delta x_j) \tag{2.50}$$

Casting the simplest Hebbian learning rule into continuous time, and adding a decay term yields the equation

$$\frac{dw_{ij}}{dt} = -w_{ij} + x_i x_j \tag{2.51}$$

An extension of the continuous Hebbian learning rule is

$$\frac{dw_{ij}}{dt} = -w_{ij} + S(x_i)S(x_j) \tag{2.52}$$

where $S()$ is the sigmoid activation function. It correlates the activation values passed through the nonlinear activation function. Another differential Hebbian learning rule is described by the equation

$$\frac{dw_{ij}}{dt} = -w_{ij} + \frac{dS(x_i)}{dt}\frac{dS(x_j)}{dt} \tag{2.53}$$

where the continuous sigmoid derivative is

$$\frac{dS(x)}{dt} = S'(x)\frac{dx}{dt} \tag{2.54}$$

2.17 Competitive and Cooperative Learning

The competitive and cooperative processes are described as neural networks with self-exciting recurrent connections and neighbor connections which are either inhibiting (competitive) or exciting (cooperative). The system of differential equations

$$\frac{dx_i}{dt} = F_i(x_1, x_2, ..., x_n) = F_i(X) \tag{2.55}$$

for i=1,2,...,n, is competitive if

$$\frac{\partial F_i}{\partial x_j} \leq 0, \forall j \neq i \tag{2.56}$$

and cooperative if

$$\frac{\partial F_i}{\partial x_j} > 0, \forall j \neq i \tag{2.57}$$

Competitive learning is a pattern classification procedure for training intra-layer connections in a two-layer neural network. In its simplest form ("winner take all"), competitive learning works in the following manner:

1. An input pattern is presented to the layer F_A.
2. The PEs of F_A send their activations to F_B by intra-layer connections.
3. Each PE of F_B competes with the others by sending positive signals to itself (recurrent self-excitation) and negative signals to all its neighbors (lateral neighbor-inhibition). Sometimes the PE of F_B with the greatest activation is singularly active and all others are nullified. The PE of F_B with the largest activation is called the F_B winner.

The competition takes place using the equation

$$\frac{dw_{ij}}{dt} = S(x_i)[-w_{ij} + S(y_j)] \tag{2.58}$$

where w_{ij} is the connection strength from the i-th PE of F_A, to the j-th PE of F_B, and $S()$ is the sigmoidal function. The above equation automatically adjusts only those connections emanating from the winner PE of F_B, leaving all other connections unaffected.

One extension of competitive learning is Grossberg's competitive cooperative

learning. A competitive-cooperative learning neural network has excitatory (positive) or inhibitory (negative) connections to each PE. The competitive learning equation is the same, but the activation dynamics are entirely different [4].

2.18 Weight Matrix Computation

The initial values of learning matrices in recurrent networks can be determined by supervised or unsupervised learning methods. The most popular method for constructing the weight matrices for recurrent networks is bipolar Hebbian or outer-product learning method [2], [4]. The method [2], [4] sums weighted correlation matrices as follows. Given m binary vectors (X_i, Y_i) where X_i denotes a binary vector of length n (it can be identified with the initial input condition of the first layer in BAM, F_X) and Y_i denotes a binary vector of length p (it can be identified with the initial input condition of the second layer in BAM, F_Y), the bipolar outer-product law sums the individual correlations

$$W = \sum_{i=1}^{m} X_i^T Y_i \tag{2.59}$$

In [4] Kosko suggests that outer-product recurrent networks apply to only a small set of association training samples, and overlooks the signal-processing potential of many stable recurrent networks. Thus outer-product can be used to determine the initial condition of weight matrices in recurrent networks. [4] describes other methods of computing as the optimal linear associative memory based on pseudo-inverse computation.

2.19 Autoassociators

This section presents the main architectures used in the following chapters. Cohen [1988] calls two-layer networks heteroassociative, and one-layer networks autoassociative. By the generalization proposed in chapters 4 and 5, N-layer networks are called heteroassociative.

Non-Adaptive Autoassociators Non-adaptive autoassociators are one layer neural networks that operate in continuous time, employ feedback recall, and have constant connection weights. Associators compute their future state based on present or past states and outputs of one layer's neurons.

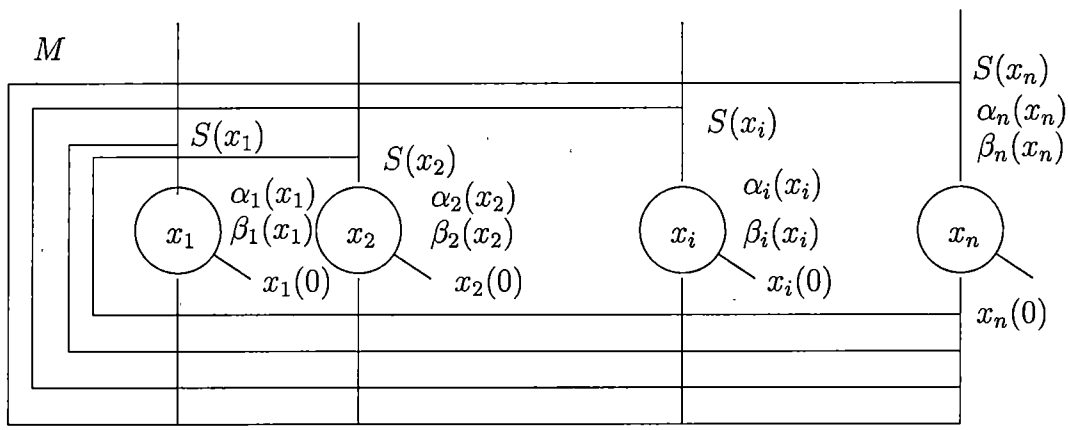
Cohen-Grossberg's Model A one-layer non - adaptive autoassociator possesses Cohen-Grossberg[1983] activation dynamics if its activation equations have the form

$$\frac{dx_i}{dt} = \alpha_i(x_i)[\beta_i(x_i) - \sum_{j=1}^{n} m_{ij} S_j(x_j)]$$

where $1 \leq i \leq n$. In a one-layer network with n neurons, described as in Fig.2.11, x_i represents the i-th neuron state, $S_i(\xi)$ is the activation function of the i-th neuron, $\alpha_i(x_i)$ and $\beta_i(x_i)$ are typical functions for the i-th neuron and each m_{ij} represents the weight of the connection between the i-th neuron and the j-th neuron of the one layer neural network. The nonnegative functions $\alpha_i(x_i) \geq 0$ represent abstract amplification functions, assumed upper bounded. The functions β_i can be of any type. To assure the stability theorem they must satisfy the conditions of bounded integral from [24]. The model is called an autoassociator because the updated state of i-th neuron depends of

Cohen-Grossberg's Model

$$\frac{dx_i}{dt} = \alpha_i(x_i)[\beta_i(x_i) - \sum_{j=1}^{n} m_{ij} S_j(x_j)]$$



One Layer Recurrent Neural Network

Fig. 2.11. Cohen-Grossberg's Model

present state and output of itself $\alpha_i(x_i), \beta_i(x_i), S_i(x_i)$ and the other neurons from the same layer, $\forall j \neq i, S_j(x_j)$.

Adaptive Autoassociators Kosko [4] has extended the Cohen-Grossberg model [24] by replacing the constant weight matrix m_{ij} with the adaptive learning rules described in sections 2.16 and 2.17.

Cohen-Grossberg-Kosko Model A one-layer adaptive autoassociator possesses Cohen-Grossberg-Kosko [1990] activation dynamics if its system equations have the form

$$\frac{dx_i}{dt} = \alpha_i(x_i)[\beta_i(x_i) - \sum_{j=1}^{n} m_{ij} S_j(x_j)] \tag{2.60}$$

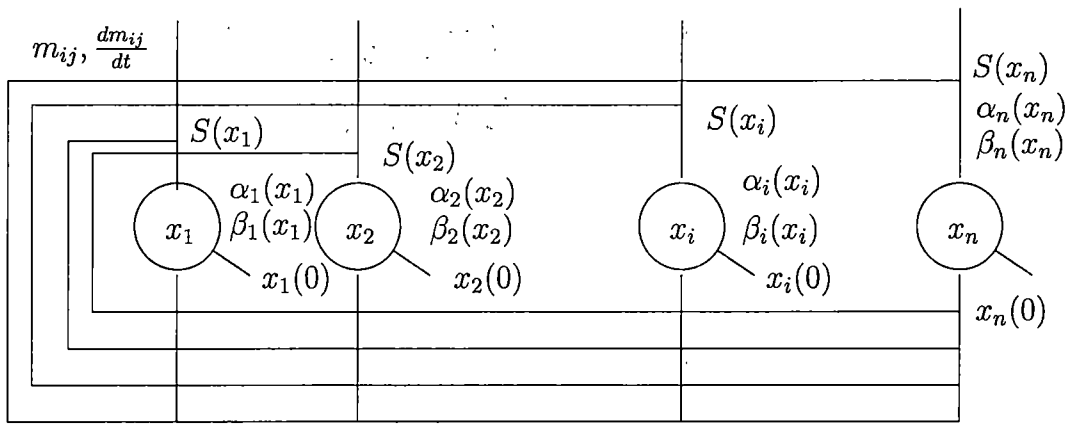
$$\frac{dm_{ij}}{dt} = -m_{ij} + S_i(x_i) S_j(x_j) \tag{2.61}$$

with the Hebbian learning rule defined in section 2.16, $1 \leq i \leq n$. In the one-layer adaptive network with n neurons, x_i represents the i-th neuron state, $S_i(\xi)$ is the activation function of the i-th neuron, $\alpha_i(x_i)$ and $\beta_i(x_i)$ are typical functions for the i-th neuron and each m_{ij} is the adaptive (2.61) weight connection between the i-th neuron and the j-th neuron of the one layer neural network, governed by Hebbian learning rule defined in section 2.16. Figure 2.12 describes this model. It is called an associator because the updated state of i-th neuron depends of present state and output of itself $\alpha_i(x_i), \beta_i(x_i), S_i(x_i)$ and the other neurons from the same layer, $\forall j \neq i, S_j(x_j)$.

Cohen-Grossberg-Kosko Model

$$\frac{dx_i}{dt} = \alpha_i(x_i)[\beta_i(x_i) - \sum_{j=1}^{n} m_{ij} S_j(x_j)]$$

$$\frac{dm_{ij}}{dt} = -m_{ij} + S_i(x_i) S_j(x_j)$$



One Layer Adaptive Recurrent Neural Network

Fig. 2.12. Cohen-Grossberg-Kosko Model

2.20 Heteroassociators

In [4] Kosko calls BAM networks heteroassociators and extends them to adaptive systems by ABAM Model. The ABAM Model [4], [2], [3] describes in three general equations two - layer evolution synchronized with adaptive learning.

ABAM Model [4] A two-layer heteroassociator possesses Kosko [1990] activation dynamics if its activation and learning equations have the form

$$\frac{dx_i}{dt} = a_i(x_i)[b_i(x_i) - \sum_{j=1}^{p} m_{ij} S_j(y_j)], \tag{2.62}$$

and

$$\frac{dy_j}{dt} = c_j(y_j)[d_j(x_j) - \sum_{i=1}^{n} m_{ij} S_i(x_i)], \tag{2.63}$$

with the Hebbian learning rule defined in section 2.16

$$\frac{dm_{ij}}{dt} = -m_{ij} + S_i(x_i) S_j(x_j) \tag{2.64}$$

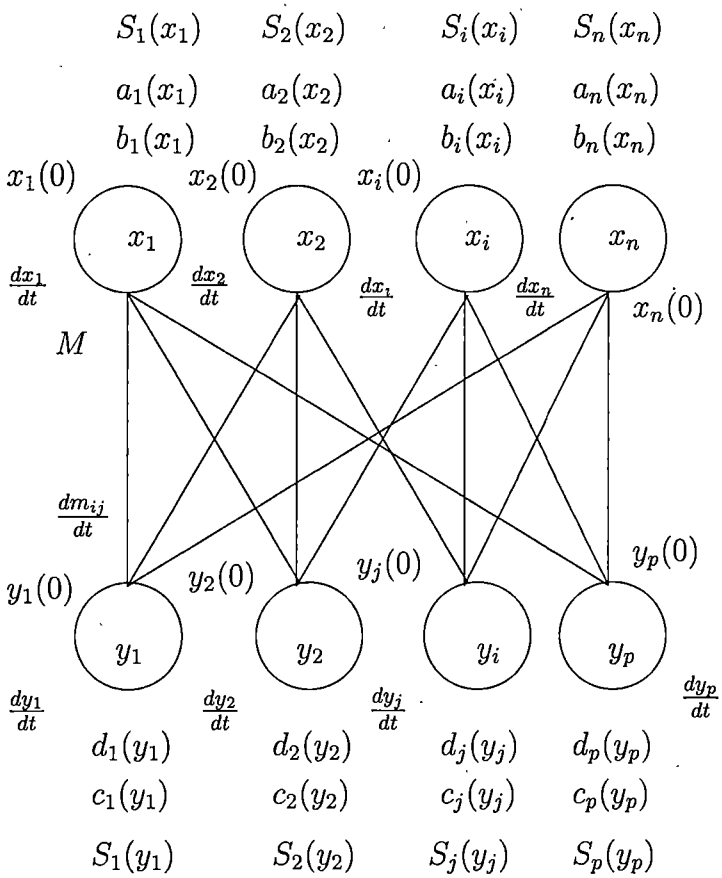
In the two-layer network with n neurons in the first layer, F_X, and p neurons in the second layer, F_Y, x_i represents the i-th neuron state of the F_X layer, $S_i(\xi)$ is the activation function of the i-th neuron of the F_X layer, $a_i(x_i)$ and $b_i(x_i)$ are typical functions for the i-th neuron of the F_X layer, y_j represents the j-th neuron state of the F_Y layer, $S_j(\xi)$ is the activation function of the j-th neuron of the F_Y layer, $c_j(x_j)$ and $d_j(x_j)$ are typical functions for the j-th neuron of the F_Y layer and each m_{ij} is the adaptive (2.64) weight connection between the i-th neuron of the first layer and the j-th neuron of the F_Y layer. Heteroassociators compute the updated state of i-th neuron from the F_X layer based on

the present states and outputs of itself, $x_i, a_i(x_i), b_i(x_i), S_i(x_i)$ and the neurons from the other layer $F_Y; \forall j, x_j, S_j(x_j)$. For a neuron in layer F_X just itself and the neurons from the F_Y layer can contribute to its updated state. Figure 2.13 describes this model.

ABAM Model

Learning equation

$$\frac{dm_{ij}}{dt} = -m_{ij} + S_i(x_i)S_j(x_j)$$



System Equations

$$\frac{dx_i}{dt} = a_i(x_i)[b_i(x_i) - \sum_{j=1}^{p} m_{ij}S_j(y_j)]$$

$$\frac{dy_j}{dt} = c_j(y_j)[d_j(x_j) - \sum_{i=1}^{n} m_{ij}S_i(x_i)]$$

Fig. 2.13. ABAM Model

Chapter 3

Stability and Convergence

3.1 Review of Literature

Neural network processing is typically governed by two mathematical concepts: stability and convergence. The first is the stability of the PE activation over time given any initial input. The second is convergence of the error between the desired and computed PE outputs to a minimum. Stability usually is associated with feedback recall, and convergence usually is associated with supervised learning. Convergence and stability theories have long histories. Liapunov stability theory is the foundation of the stability of several neural network architectures [23], [26]. The one-layer recurrent neural network was investigated and solved by Cohen and Grossberg [24]. The two layer neural network, bidirectional associative memory BAM case, was initiated and analyzed by Kosko [1], [2], [4]. Sections 3.13-3.16 present some previous results in order to understand the original results for N layer recurrent neural networks analyzed in chapters 4 and 5.

3.2 A Definition of Convergence

A convergent sequence is defined as an infinite sequence of numbers x_n that tends to a limit, x. A more precise definition is: For any given error, $e > 0$, there exists a positive integer n_0 such that $|x_n - x| < e$ for every $n > n_0$.

3.3 A Definition of Global Stability

Globally stable neural networks are defined as nonlinear dynamic systems that bring all inputs to fixed points. These fixed points (limit points, convergence points, equilibria...) are places where information can be deliberately stored. Although global stability guarantees all inputs are mapped to a fixed point, it does not guarantee that it will map an input onto the desired fixed point.

3.4 Stability Definitions

A vector differential equation is described as

$$\frac{dx}{dt} = f(t, x(t)), t \geq 0 \tag{3.1}$$

where $x(t) \in R^n$, and $f : R_+ \times R^n \longrightarrow R^n$ are continuous. It is assumed that the equation has a unique solution corresponding to each initial condition.

$s(t, t_0, x_0)$ denotes the solution of the equation (3.1) corresponding to the initial conditions $x(t_0) = x_0$, evaluated at time t_0. In other words, s satisfies the equation

$$\frac{ds(t, t_0, x_0)}{dt} = f[t, s(t, t_0, x_0)], \forall t \geq t_0; s(t_0, t_0, x_0) = x_0 \tag{3.2}$$

The solution, s maps $R_+ \times R^n$ into R^n, and satisfies the following properties:

$$s(t_0, t_0, x_0) = x_0, \forall x_0 \in R^n; s(t, t_1, s(t_1, t_0, x_0)) = s(t, t_0, x_0), \forall t \geq t_1 \geq t_0, \forall x_0 \in R^n \tag{3.3}$$

We define a vector $x_0 \in R^n$ as an equilibrium of the above system if

$$f(t, x_0) = 0, \forall t \geq 0 \tag{3.4}$$

or equivalently

$$s(t, t_0, x_0) = x_0, \forall t \geq t_0 \geq 0 \tag{3.5}$$

In other words, if the system starts at an equilibrium, it remains there. If the equilibrium under study is not the origin, one can always redefine the coordinates in the state space R^n so that the equilibrium of interest becomes the new origin. Thus without loss of generality, it is assumed that at equilibrium

$$f(t, 0) = 0, \forall t \geq 0 \tag{3.6}$$

This is equivalent to

$$s(t, t_0, 0) = 0, \forall t \geq t_0 \tag{3.7}$$

3.5 Definition - Stable Equilibrium

The equilibrium 0 is stable if, for each $\epsilon > 0$ and each $t_0 \in R_+$, there exists a $\delta(\epsilon, t_0) \in R$ such that

$$\|x_0\| < \delta(\epsilon, t_0) \Longrightarrow \|s(t, t_0, x_0)\| < \epsilon, \forall t \geq t_0 \tag{3.8}$$

The equilibrium 0 is uniformly stable if, for each $\epsilon > 0$, there exist a $\delta(\epsilon)$ such that

$$\|x_0\| < \delta(\epsilon), t_0 \geq 0 \Longrightarrow \|s(t, t_0, x_0)\| < \epsilon, \forall t \geq t_0 \tag{3.9}$$

The equilibrium is unstable if it does not satisfy either stability criterion.

3.6 Definition - Attractive Equilibrium

The equilibrium 0 is attractive if, for each $t_0 \in R_+$, there exists an $\eta(t_0) > 0$ such that

$$\|x_0\| < \eta(t_0) \Longrightarrow s(t_0 + t, t_0, x_0) \longrightarrow 0 \tag{3.10}$$

as $t \longrightarrow \infty$.

The equilibrium 0 is uniformly attractive if there is a real number $\eta > 0$ such that

$$\|x_0\| < \eta, t_0 \geq 0 \Longrightarrow s(t_0 + t, t_0, x_0) \longrightarrow 0 \tag{3.11}$$

as $t \longrightarrow \infty$ uniformly in x_0, t_0.

3.7 Definition - Global Uniform Asymptotic Equilibrium

The equilibrium 0 is globally uniformly asymptotically stable if,

1. It is uniformly stable, and
2. For each pair of positive numbers M, ϵ with M arbitrarily large and ϵ arbitrarily small, there exists a finite real number $T(M, \epsilon)$ such that

$$\|x_0\| < M, t_0 \geq 0 \Longrightarrow \|s(t_0 + t, t_0, x_0)\| < \epsilon, \forall t \geq T(M, \epsilon) \tag{3.12}$$

3.8 Liapunov Function Discrete Case

Discrete systems are described by the recursive relationship

$$x_{k+1} = f_k(x_k) \tag{3.13}$$

where $x_k \in R^n$, and $f_k : R^n \to R^n$ for all $k \geq 0$. The equation (3.13) has one solution, given an initial condition of the form $x(k_0) = x_0$. This solution evaluated at the k-th time ($k \geq k_0$), is denoted by $s(k, k_0, .)$ being continuous for each pair (k, k_0). A point $x_0 \in R^n$ is called an equilibrium of the system (3.13) if

$$f_k(x_0) = x_0, \forall k \geq 0 \tag{3.14}$$

and

$$s(k, k_0, x_0) = x_0, \forall k \geq k_0 \geq 0 \tag{3.15}$$

We can assume, without loss of generality, that the equilibrium of interest is the origin,

$$f_k(0) = 0, \forall k \geq 0 \tag{3.16}$$

and there is a function $V : Z_+ \times R^n \to R$ such that

$$V_k^* = V[k, s(k, k_0, x_0)] \tag{3.17}$$

The forward difference of the sequence V_k^* is

$$\Delta V_k^* = V_{k+1}^* - V_k^* = V[k+1, s(k+1, k, x_k)] - V(k, x_k). \tag{3.18}$$

where ΔV depends on both the function V and the system (3.13). The evolution along the trajectory of (3.13) is given by

$$V_k^* = V_j^* + \Sigma_{i=j}^{k-1} \Delta V(i, x_i) \tag{3.19}$$

Definition The equilibrium 0 of the system (3.13) is stable if, for each $\epsilon > 0$ and each integer $k_0 \geq 0$, there exists a $\delta = \delta(\epsilon, k_0)$ such that

$$||x_k|| \leq \delta(\epsilon, k_0) \Rightarrow ||s(k, k_0, x_0)|| < \epsilon, \forall k \geq k_0 \tag{3.20}$$

3.9 Liapunov Function

Definition A function $V : R_+ \times R^n \rightarrow R$ is a locally positive definite function (lpdf) if

1. it is continuous
2. $V(t, 0) = 0, \forall t \geq 0$, and
3. there exists a small constant $r > 0$ and a function α of class C^1 such that

$$\alpha(||x||) \leq V(t, x), \forall t \geq 0, \forall x \in B_r. \tag{3.21}$$

Definition V is a positive definite function (pdf) if the above conditions are satisfied for all $x \in R^n$.

From [23] we select the following conclusions:

A continuous function $V : R_+ \times R^n \rightarrow R$ is an lpdf if and only if it satisfies the following two conditions:

1. $V(0) = 0$
2. there exists a constant $r > 0$ such that $V(x) > 0, \forall x \in B_r - 0$.

A continuous function $V : R_+ \times R^n \rightarrow R$ is an pdf if it satisfies the following three conditions:

1. $V(0) = 0$

2. $V(x) > 0, \forall x \in R^n - 0$

3. there exists a constant $r > 0$ such that $\inf_{||x|| \geq r} V(x) > 0$.

3.10 Theorems of Stability

The equilibrium 0 of the system (3.1) is locally stable if there exist a C^1 (lpdf) function $V : R_+ \times R^n \longrightarrow R$ and a small constant $r > 0$ such that

$$\frac{dV(t, x(t))}{dt} \leq 0, \forall t \geq t_0 \tag{3.22}$$

where $\frac{dV}{dt}$ is evaluated along the trajectories of the system (3.1).

The equilibrium 0 of the system (3.1) is globally stable if there exist a C^1 (pdf) function $V : R_+ \times R^n \longrightarrow R$ such that

$$\frac{dV(t, x(t))}{dt} \leq 0, \forall t \geq t_0 \tag{3.23}$$

where $\frac{dV}{dt}$ is evaluated along the trajectories of the system. For discrete system (3.13) the above definitions hold by replacing (3.22) and (3.23) with

$$\Delta V = V(t+1) - V(t) \leq 0$$

For the proofs and further explanations, [23] is recommend.

3.11 Liapunov's Direct Method

The direct method of Liapunov [5] studies dynamic system stability by finding certain functions and studying their time derivatives. This method is in direct contrast

with the alternative method of motion integration. If the conditions are satisfied, Liapunov's direct method proves global or local system stability. The specific properties needed for Liapunov's direct method to prove the global stability of the system

$$\frac{dx_j}{dt} = F(t, x_1, x_2, ..., x_n) \tag{3.24}$$

are as follows:

$\frac{dx_j}{dt} = 0$ if $x_i = 0,\ \forall i = 1, 2, ..., n$ (i.e. 0 only at the origin)

$$\frac{dV}{dt} = \sum_{i=1}^{n} \frac{\partial V}{\partial x_i} \frac{dx_i}{dt} \leq 0,\ \forall x_i \tag{3.25}$$

where V is a Liapunov system energy function (pdf). Liapunov's direct method states that if the above properties are satisfied for a given energy function V, then for all possible system inputs $X = (x_1, x_2, ..., x_n)$, the system converges to a globally stable solution.

3.12 Discussion of the Liapunov Stability

In the previous sections several results in Liapunov stability theory were summarized. The favorable aspects of these are:

1. They enable us to draw conclusions about the stability properties of an equilibrium without solving the system equations.
2. Especially for stability and asymptotic stability, the Liapunov function V has an intuitive interpretation as the system energy.

The unfavorable aspects of these theorems are:

1. They represent only sufficient conditions for the system stability. Thus, if a particular Liapunov function candidate V fails to satisfy the Liapunov theorem conditions, then no conclusion can be drawn, and a new analysis should begin with another Liapunov function candidate.
2. In a general nonlinear system there is no systematic procedure to generate a Liapunov function candidate and the investigator's creativity is needed.

The problem now is how to prove dynamic system stability. The first or direct approach is solving the equations and then studying the system evolution by differential equation theory or something similar. The second approach is to find a Liapunov function candidate as in section 3.11. The Liapunov method offers a short way to prove the global stability of a dynamic system. If a Liapunov function candidate can not be found, other mathematical strategies must be applied. A dynamic system may or may not be stable. But if there is a Liapunov function meeting the Liapunov theorem conditions, then stability is guaranteed. In general the Liapunov approach reveals only the existence of stable points, not their number or nature.

3.13 Three General Stability Theorems

There are three general theorems that describe the stability of many recurrent neural networks [4]. The first theorem is the Cohen-Grossberg theorem [24]. It is used to show the stability of non-adaptive autoassociators. The second theorem is the Cohen-Grossberg-Kosko theorem [3]. It is used to show the stability of adaptive autoassocia-

tors. A third theorem, the ABAM theorem [22], is used to show the stability of adaptive BAM heteroassociators. These are explained in the following three sections.

3.14 Cohen-Grossberg's Theorem

In the following theorem for one-layer network with n neurons, x_i represents the i-th neuron state, $S_i(\xi)$ is the activation function of the i-th neuron, $\alpha_i(x_i)$ and $\beta_i(x_i)$ are typical functions for the i-th neuron and each m_{ij} represents the weight of the connection between the i-th neuron and the j-th neuron of the one layer neural network. The nonnegative functions $\alpha_i(x_i) \geq 0$ represent abstract amplification functions, assumed upper bounded. The functions β_i can be of any type. To assure the stability theorem they must satisfy the conditions of bounded integral from [24]. The model is called an autoassociator because the updated state of i-th neuron depends of present state and output of itself $\alpha_i(x_i), \beta_i(x_i), S_i(x_i)$ and the other neurons from the same layer, $\forall j \neq i, S_j(x_j)$.

For any one layer nonlinear dynamic system with n neurons of the form

$$\frac{dx_i}{dt} = \alpha_i(x_i)[\beta_i(x_i) - \sum_{j=1}^{n} m_{ji} S_j(x_j)] \tag{3.26}$$

where $1 \leq i \leq n$ and

1. the matrix $||m_{ij}||$ is symmetric and all $m_{ii} \geq 0$
2. the functions $\alpha_i(\xi)$ are continuous for all ξ
3. the functions $\alpha_i(\xi) \geq 0$ for all ξ, and the functions $S_i(\xi) \geq 0$ for all ξ
4. the functions $S_i(\xi)$ are differentiable and nondecreasing for all ξ
5. the equation (3.26) describes the time network activation

the function

$$V = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} m_{ji}S_i(x_i)S_j(x_j) - \sum_{i=1}^{n}\int_0^{x_i} S_i'(\Theta_i)\beta_i(\Theta_i)d\Theta_i \tag{3.27}$$

is the Liapunov system energy function and the system is globally stable. The proof can be found in [24]. Usually the Liapunov Candidate function is also called the "energy" of the system.

3.15 Cohen-Grossberg-Kosko Theorem

In the following theorem for one-layer adaptive network with n neurons, x_i represents the i-th neuron state, $S_i(\xi)$ is the activation function of the i-th neuron, $\alpha_i(x_i)$ and $\beta_i(x_i)$ are typical functions for the i-th neuron and each m_{ij} is the adaptive (2.61) weight connection between the i-th neuron and the j-th neuron of the one layer neural network, governed by Hebbian learning rule defined in section 2.16. The model is called associator because the updated state of i-th neuron depends of present state and output of itself $\alpha_i(x_i), \beta_i(x_i), S_i(x_i)$ and the other neurons from the same layer, $\forall j \neq i, S_j(x_j)$.

For any nonlinear adaptive one layer dynamic system of the form

$$\frac{dx_i}{dt} = \alpha_i(x_i)[\beta_i(x_i) - \sum_{j=1}^{n} m_{ij}S_j(x_j)] \tag{3.28}$$

$$\frac{dm_{ij}}{dt} = -m_{ij} + S_i(x_i)S_j(x_j) \tag{3.29}$$

with the Hebbian learning rule given in (3.29), $1 \leq i \leq n$ and

1. the matrix $||m_{ij}||$ is symmetric and all $m_{ii} \geq 0$
2. the functions $\alpha_i(\xi)$ are continuous for all ξ

3. the functions $\alpha_i(\xi) \geq 0$ for all ξ, and the function $S_i(\xi) \geq 0$ for all ξ
4. the functions $S_i(\xi)$ are differentiable and nondecreasing for all ξ
5. the equation (3.28) describes the time network activation
6. the dynamic equation (3.29) describes the changes in the interconnection network strengths m_{ji}

the function

$$V = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} m_{ij}S_i(x_i)S_j(x_j) - \sum_{i=1}^{n}\int_0^{x_i} S_i'(\Theta_i)\beta_i(\Theta_i)d\Theta_i - \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n} m_{ij}^2 \quad (3.30)$$

is a Liapunov function and the system is globally stable. The proof can be found in [4]. The network with one layer has n neurons. Usually the Liapunov Candidate function is also called the "energy" of the system.

3.16 ABAM Theorem

In the following theorem for two-layer network with n neurons in the first layer, F_X, and p neurons in the second layer, F_Y, x_i represents the i-th neuron state of the F_X layer, $S_i(\xi)$ is the activation function of the i-th neuron of the F_X layer, $a_i(x_i)$ and $b_i(x_i)$ are typical functions for the i-th neuron of the F_X layer, y_j represents the j-th neuron state of the F_Y layer, $S_j(\xi)$ is the activation function of the j-th neuron of the F_Y layer, $c_j(x_j)$ and $d_j(x_j)$ are typical functions for the j-th neuron of the F_Y layer and each m_{ij} is the adaptive (2.64) weight connection between the i-th neuron of the F_X layer and the j-th neuron of the F_Y layer.

In every two layer dynamic system of the form

$$\frac{dx_i}{dt} = a_i(x_i)[b_i(x_i) - \sum_{j=1}^{p} m_{ij} S_j(y_j)], \tag{3.31}$$

and

$$\frac{dy_j}{dt} = c_j(y_j)[d_j(x_j) - \sum_{i=1}^{n} m_{ij} S_i(x_i)], \tag{3.32}$$

with Hebbian learning rule

$$\frac{dm_{ij}}{dt} = -m_{ij} + S_i(x_i) S_j(x_j) \tag{3.33}$$

such that $1 \leq i \leq n, 1 \leq j \leq p$ and

1. the functions $b_i(x)$ and $d_j(x)$ are continuous for all x
2. the functions $a_i(x)$ and $c_j(x) \geq 0$ are continuous for all x
3. the functions $S_i(x)$ and $S_j(x) \geq 0$ for all x
4. the functions $S_i(x)$ and $S_j(x)$ are differentiable and nondecreasing for all x
5. the above (3.31), (3.32) equations describe the time activation of the layer $F_X \in R^n$ and $F_Y \in R^p$, respectively
6. the dynamic equation (3.33) of m_{ij} describes the changes in the interconnection strengths between F_X and F_Y PEs when $M \in R^n \times R^p$

the function

$$V = -\sum_{i=1}^{n}\sum_{j=1}^{p} m_{ij} S_i(x_i) S_j(y_j) + \sum_{i=1}^{n} \int_0^{x_i} S_i'(\Theta_i) b_i(\Theta_i) d\Theta_i \tag{3.34}$$

$$+ \sum_{j=1}^{p} \int_0^{y_j} S_j'(\zeta_j) d_j(\zeta_j) d\zeta_j + \frac{1}{2} \sum_{i=1}^{n}\sum_{j=1}^{p} m_{ij}^2 \tag{3.35}$$

is a Liapunov function and the system is globally stable.

The proof can be found in [4], [27], [21]. For all these systems there is a reciprocal equilibrium relationship. The system is at equilibrium ($\frac{dV}{dt} = 0$) if and only if

$$(\frac{dx_i}{dt} = 0)(\frac{dy_j}{dt} = 0)(\frac{dm_{ij}}{dt} = 0) \tag{3.36}$$

for all i and j, and the stability is achieved in exponential time [4]. As a point of interest, the ABAM theorem [4] can be placed in an algebraic framework of Cohen-Grossberg-Kosko theorem, [24], and vice versa [4].

Chapter 4

Generalization in Recurrent Neural Networks

4.1 Continuous Associative Memories

The original presentation begins by recalling the Hopfield and BAM models [24], [4] and defining the multilayer associative memory equations. A system of $n+p$ coupled first order differential equations defines an additive activation model that interconnects layers F_X and F_Y through the constant synaptic matrices M and M^T:

$$\frac{dx_i}{dt} = -a_i x_i + \sum_{j=1}^{p} S_j(y_j) m_{ij} + I_i \tag{4.1}$$

$$\frac{dy_j}{dt} = -b_j y_j + \sum_{i=1}^{n} S_i(x_i) m_{ij} + J_j \tag{4.2}$$

where $1 \leq i \leq n, 1 \leq j \leq p$. This model is called "additive" because the total activation received by a neuron

$$\sum_{j=1}^{p} S_j(y_j) m_{ij}$$

has an additive effect on the future excitation of that neuron. The additive autoassociative $F_x = F_y$ model, the Hopfield model, corresponds to a system of n coupled first - order differential equations:

$$\frac{dx_i}{dt} = -a_i x_i + \sum_{j=1}^{n} S_j(x_j) w_{ij} + I_i \tag{4.3}$$

where $1 \leq i \leq n$. The multiplicative model is defined as

$$\frac{dx_i}{dt} = -a_i(x_i)[b_i(x_i) - \sum_{j=1}^{p} S_j(y_j)m_{ij}] \tag{4.4}$$

$$\frac{dy_j}{dt} = -c_j(y_j)[d_j(y_j) - \sum_{i=1}^{n} S_i(x_i)m_{ij}] \tag{4.5}$$

where $1 \leq i \leq n, 1 \leq j \leq p$. This model is called "multiplicative" because the total activation received by a neuron

$$\sum_{j=1}^{p} S_j(y_j)m_{ij}$$

is multiplied by $a_i(x_i)$ in order to compute the future excitation of that neuron. The terms are explained in section 2.20. The multiplicative autoassociative $F_x = F_y$ model corresponds to a system of n coupled first - order differential equations:

$$\frac{dx_i}{dt} = -a_i(x_i)[b_i(x_i) - \sum_{j=1}^{n} S_j(x_j)w_{ij}] \tag{4.6}$$

where $1 \leq i \leq n$. In [24], Cohen and Grossberg proved that if the system (4.6) satisfies the conditions of $w_{ij} = w_{ji}; w_{ii} \geq 0$ (symmetric, nonnegative diagonal weights), $S_j()$ is a differentiable, nonnegative, nondecreasing function, $a_i()$ and $b_i()$ are continuous functions and $a_i(x) \geq 0$, then there is a Liapunov function, V, bounded below, that satisfies

$$\frac{d}{dt}V(x) \leq 0 \tag{4.7}$$

on the system evolution trajectory discussed in section 3.14. The system (4.3) or (4.6) corresponds to a one - layer symmetric autoassociative case. It includes just the intra-connections between the neurons from the same layer. The BAM insists on the inter-

connection between neurons from different layers. One layer activates the other and so on.

Autoassociative and heteroassociative were defined in sections 2.19 and 2.20. Mathematically heteroassociative neural networks can be generalized to work with the cartesian product $R_{n_1} \times R_{n_2} \times R_{n_3}...$, where $n_1, n_2, n_3...$ are the numbers of neurons in the first layer, second layer, third layer, and so on..., for an arbitrary number of layers. Autoassociative networks work with the space $R_{n_1+n_2+n_3...}$, where $n_1 + n_2 + n_3 + ...$ is the number of neurons in a associative network as will be discussed in section 4.4. The mathematical equations are homologous, but the ranges are different. Kosko [4] points out that the Cohen-Grossberg Theorem [Grossberg, 1988], [24], ensures the global stability of (4.3) and (4.6) which corresponds (in the sense that $R_{n_1} \times R_{n_2}$ is isomorphic with $R_{n_1+n_2}$, so they keep the transformation proprieties) to the continuous BAM theorem for the nonadaptive heteroassociative networks. Also it represents a special case of the ABAM and RABAM [4], [27]. Mathematically there is an isomorphism between the heteroassociative multilayer network and an autoassociative network with one layer composed of all the layers of the heteroassociative network. It can be proved mathematically that a BAM has an isomorphic relationship with an autoassociative recurrent neural network where the weights of neurons' intraconnections of the same layer are zero [4]. The main block diagonal in the connection matrix of the autoassociative network is zero. All BAM demonstrations can be reduced to autoassociatve network correspondents. The generalization can be made for continuous (chapter 4) and discrete (chapter 5) systems. The following multilayer generalized models are initiated.

4.2 Definition of the N - Layer Associative Memory

Given N layers, every neuron state is dependent on the activations of the neurons from the other layers. Each neuron has a connection to itself but not to any other neurons in the same layer. Every two layers are connected exactly as in a BAM. Every neuron from one layer receives the activated signals from all other neurons from other layers but not its own layer. The mathematical equations can be written as follows,

The Additive Activation Model

$$\frac{dx_i^k}{dt} = -A_i^k x_i^k + \sum_{l\neq k, l=1}^{N} \sum_{j=1}^{n_l} S_j(x_j^l) w_{ij}^{kl} + I_i^k \tag{4.8}$$

where $1 \leq i \leq n^k, 1 \leq k \leq N$

The Multiplicative Activation Model

$$\frac{dx_i^k}{dt} = -A_i^k(x_i^k)[B_i^k(x_i^k) - \sum_{l\neq k, l=1}^{N} \sum_{j=1}^{n_l} S_j(x_j^l) w_{ij}^{kl}] \tag{4.9}$$

where $1 \leq i \leq n^k, 1 \leq k \leq N$ (the notation $W^{lk} = (W^{kl})^T$ represents the connection between the layers l and k, and vice versa). The following notations are used: the upper index k is used for everything that belongs to the layer k, where $1 \leq k \leq N$; i corresponds to the i-th neuron, $(1 \leq i \leq n_k)$, from the layer k with n_k neurons; j corresponds to the j-th neuron, $(1 \leq j \leq n_l)$, from the layer l which has n_l neurons, w_{ij}^{kl} is the weight of connection between two neurons from different layers, which in this notation denotes the weight of connection between the neuron i-th of the layer k and the j-th neuron of the layer l (when $k \neq l$ for heteroassociative model); A_i^k are the coefficients from the additive model with different values corresponding to each neuron i in layer k; $A_i^k(x_i^k)$

and $B_i^k(x_i^k)$ are the multiplicative model functions with different forms corresponding to the i-th neuron from layer k.

These generalized models are difficult to understand, so some examples are provided for N=2 and N=3. The additive system given by the equations (4.1) and (4.2) can be written for two layers as

$$\frac{dx_i^1}{dt} = -A_i^1 x_i^1 + \sum_{j=1}^{n_2} S_j(x_j^2) w_{ij}^{12} + I_i^1$$
$$\frac{dx_i^2}{dt} = -A_i^2 x_i^2 + \sum_{j=1}^{n_1} S_j(x_j^1) w_{ij}^{21} + I_i^2$$

where $1 \leq i \leq n^k, 1 \leq k \leq 2$, as references [2], [4].

The multiplicative system given by the equations (4.4) and (4.5) can be written for two layers as

$$\frac{dx_i^1}{dt} = -A_i^1(x_i^1)[B_i^1(x_i^1) - \sum_{j=1}^{n_2} S_j(x_j^2) w_{ij}^{12}]$$
$$\frac{dx_i^2}{dt} = -A_i^2(x_i^2)[B_i^2(x_i^2) - \sum_{j=1}^{n_1} S_j(x_j^1) w_{ij}^{21}]$$

where $1 \leq i \leq n^k, 1 \leq k \leq 2$ and n_1 and n_2 are the number of neurons in the two layers. This was explaned in section 2.20. For N=3, the additive and multiplicative model can be written as follows: A system of $n_1 + n_2 + n_3$ coupled first - order differential equations defines the additive activation model that interconnects layers F_{X^1}, F_{X^2} and F_{X^3} through constant synaptic matrices (by notation $W^{lk} = (W^{kl})^T$ which represent the connection between the layers l and k).

The Additive Model for three-layers

$$\frac{dx_i^1}{dt} = -A_i^1 x_i^1 + \sum_{l \neq 1, l=1}^{N} \sum_{j=1}^{n_l} S_j(x_j^l) w_{ij}^{1l} + I_i^1$$

$$\frac{dx_i^2}{dt} = -A_i^2 x_i^2 + \sum_{l\neq 2, l=1}^{N} \sum_{j=1}^{n_l} S_j(x_j^l) w_{ij}^{2l} + I_i^2$$
$$\frac{dx_i^3}{dt} = -A_i^3 x_i^3 + \sum_{l\neq 3, l=1}^{N} \sum_{j=1}^{n_l} S_j(x_j^l) w_{ij}^{3l} + I_i^3$$

where $1 \leq i \leq n^k, 1 \leq k \leq 3$

The Multiplicative Model for three layers

$$\frac{dx_i^1}{dt} = -A_i^1(x_i^1)[B_i^1(x_i^1) - \sum_{l\neq 1, l=1}^{N} \sum_{j=1}^{n_l} S_j(x_j^l) w_{ij}^{1l}]$$
$$\frac{dx_i^2}{dt} = -A_i^2(x_i^2)[B_i^2(x_i^2) - \sum_{l\neq 2, l=1}^{N} \sum_{j=1}^{n_l} S_j(x_j^l) w_{ij}^{2l}]$$
$$\frac{dx_i^3}{dt} = -A_i^3(x_i^3)[B_i^3(x_i^3) - \sum_{l\neq 3, l=1}^{N} \sum_{j=1}^{n_l} S_j(x_j^l) w_{ij}^{3l}]$$

where $1 \leq i \leq n^k, 1 \leq k \leq 3$ and n_1, n_2, n_3 are the neuron numbers of the three layers. The model is represented in Fig.4.1.

The case $F_{x^1} = F_{x^2} = F_{x^3}$ is identified with an autoassociative network. One of the following cases, $F_{x^1} = F_{x^2}$ or $F_{x^1} = F_{x^3}$ or $F_{x^3} = F_{x^2}$, is identified as a BAM network.

4.3 Theorems for N - Layer Associative Memory

Theorem 4.1 (The Continuous N Layer Associative Memory Theorem - for Additive Model).

Every additive dynamic system of the form

$$\frac{dx_i^k}{dt} = -A_i^k x_i^k + \sum_{l\neq k, l=1}^{N} \sum_{j=1}^{n_l} S_j(x_j^l) w_{ij}^{kl} + I_i^k \tag{4.10}$$

where $1 \leq i \leq n^k, 1 \leq k \leq N$ *and*

a. the matrices $w_{ij}^{kl} = w_{ji}^{lk}$ *(by the notation convention and by the chosen heteroassocia-*

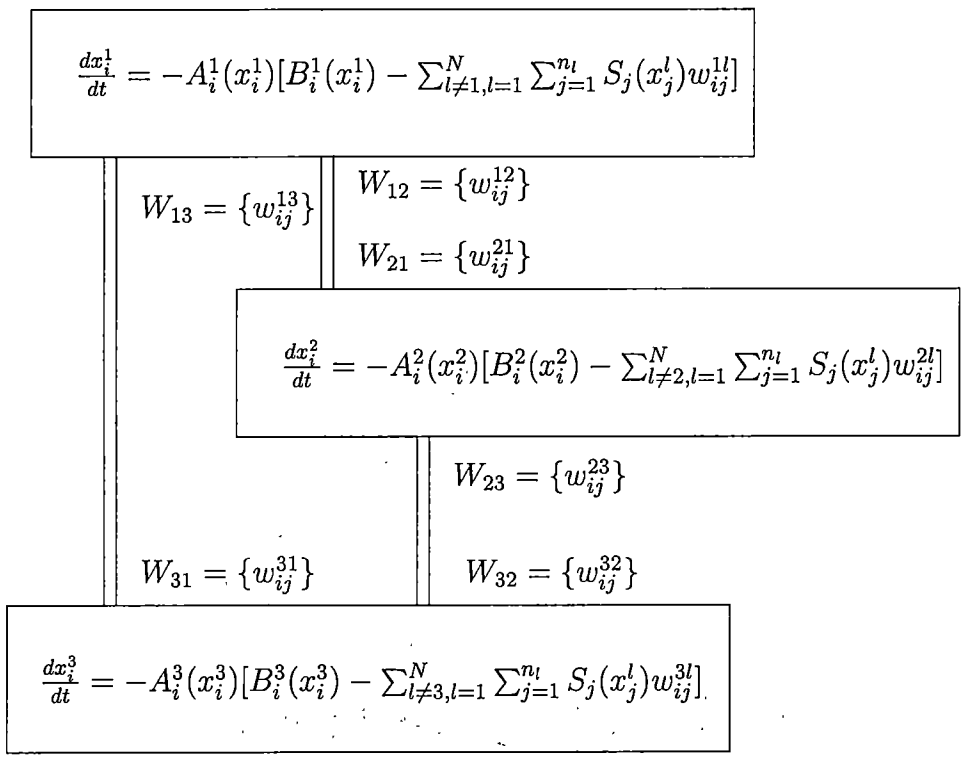


Fig. 4.1. Three Layer Recurrent Neural Network - Multiplicative Model

tive model)

b. the functions $S_i(\xi)$ are bounded continuous functions

c. the functions $S_i(\xi)$ are differentiable and nondecreasing

e. the above (4.10) equations describe the time activation of the layer $F_{X^k} \in R^n$ for every layer, $k \leq N$

f. it is assumed the boundedness of the integrals used in the below Liapunov function (for example $x_i^k \geq 0$ assures the boundedness of the integrals used)

then the function

$$V = -\frac{1}{2}\sum_{k=1}^{N}\sum_{k\neq l,l=1}^{N}\sum_{i=1}^{n_k}\sum_{j=1}^{n_l} w_{ij}^{kl} S_i(x_i^k) S_j(x_j^l) + \sum_{k=1}^{N}\sum_{i=1}^{n_k} A_i^k \int_0^{x_i} S_i'(\Theta_i)\Theta_i d\Theta_i \qquad (4.11)$$

$$-\sum_{k=1}^{N}\sum_{i=1}^{n_k} S(x_i^k) I_i^k$$

is a Liapunov system energy function and the system is globally stable.

Proof. The proof can be done in two ways. One is based on Liapunov theorem and the other is based on the isomorphic equivalence between heteroassociative and autoassociative networks discussed in section 4.4. The first proof is given below starting with the boundedness of the function

$$V = -\frac{1}{2}\sum_{k=1}^{N}\sum_{k\neq l,l=1}^{N}\sum_{i=1}^{n_k}\sum_{j=1}^{n_l} w_{ij}^{kl} S_i(x_i^k) S_j(x_j^l) + \sum_{k=1}^{N}\sum_{i=1}^{n_k} A_i^k \int_0^{x_i} S_i'(\Theta_i)\Theta_i d\Theta_i$$

$$-\sum_{k=1}^{N}\sum_{i=1}^{n_k} S(x_i^k) I_i^k.$$

The boundedness of the function follows from the boundedness of the signal functions S_i stated in assumption (b) and from the hypothesis (f) of the theorem regarding the

boundedness of the integrals used (for example $x_i^k \geq 0$ assures the boundedness of the integrals used). The boundedness of the integral terms requires additional technical hypotheses to avoid pathologies. For the thesis' purpose, the boundedness of the integral terms is assumed. For every term in part

$$\sum_{k=1}^{N}\sum_{i=1}^{n_k} A_i^k \int_0^{x_i} S_i'(\Theta_i)\Theta_i d\Theta_i \tag{4.12}$$

the above integral sum is low bounded as well as

$$-\frac{1}{2}\sum_{k=1}^{N}\sum_{k\neq l,l=1}^{N}\sum_{i=1}^{n_k}\sum_{j=1}^{n_l} w_{ij}^{kl} S_i(x_i^k) S_j(x_j^l) \geq -\frac{1}{2}\sum_{k=1}^{N}\sum_{l=1}^{N}\sum_{i=1}^{n_k}\sum_{k\neq l,j=1}^{n_l} ||w_{ij}^{kl}|| \tag{4.13}$$

and

$$-\sum_{k=1}^{N}\sum_{i=1}^{n_k} S(x_i^k) I_i^k \geq -\sum_{k=1}^{N}\sum_{i=1}^{n_k} ||I_i^k|| \tag{4.14}$$

By time-differentiating V, the time-derivative operator is distributed across the right-hand side of the above function. Then each term can be time-differentiated separately. The chain rule of differentiation then gives

$$\frac{d}{dt}F(x_i(t)) = \frac{dF}{dx_i}\frac{dx_i}{dt} \tag{4.15}$$

$$\frac{d}{dt}\int_0^x f(u)du = \frac{dx}{dt}f(x) \tag{4.16}$$

The term $S_i'(x_i)\frac{dx_i}{dt}$ multiplies each additive term of the layer k.

$$\frac{dV}{dt} = -\sum_{k=1}^{N}\sum_{k\neq l,l=1}^{N}\sum_{i=1}^{n_k}\sum_{j=1}^{n_l} w_{ij}^{kl} S_i'(x_i^k)\frac{dx_i^k}{dt} S_j(x_j^l) \tag{4.17}$$

$$+\sum_{k=1}^{N}\sum_{i=1}^{n_k} A_i^k \frac{dx_i^k}{dt} S_i'(x_i^k) x_i^k$$

$$-\sum_{k=1}^{N}\sum_{i=1}^{n_k} S'(x_i^k)\frac{dx_i^k}{dt} I_i^k$$

$$\frac{dV}{dt} = -\sum_{k=1}^{N}\sum_{i=1}^{n_k} S'(x_i^k)\frac{dx_i^k}{dt}\frac{dx_i^k}{dt} \tag{4.18}$$

but $S'(x_i^k) \geq 0$, which concludes $\frac{dV}{dt} \leq 0$. After computing derivatives and grouping, the proof reaches the conclusion $\frac{dV}{dt} \leq 0$ which proves that V is a Liapunov function so the system is globally stable. For the above system there is a reciprocal equilibrium relationship. The system is at equilibrium

$$(\frac{dV}{dt} = 0)$$

if and only if

$$(\frac{dx_i^k}{dt} = 0) \tag{4.19}$$

for all k, i and the stability is achieved in exponential time. As a point of interest, the theorem can be placed in an algebraic framework of the Cohen-Grossberg-Kosko theorem [4].

Theorem 4.2 (The Continuous N Layer Associative Memory Theorem-for Multiplicative Model).

Every multiplicative dynamic system of the form

$$\frac{dx_i^k}{dt} = -A_i^k(x_i^k)[B_i^k(x_i^k) - \sum_{l\neq k, l=1}^{N}\sum_{j=1}^{n_l} S_j(x_j^l)w_{ij}^{kl}] \tag{4.20}$$

where $1 \leq i \leq n^k, 1 \leq k \leq N$ *and*

a. the matrices $w_{ij}^{kl} = w_{ji}^{lk}$ *(by the notation convention and by the chosen heteroassociative model)*

b. the functions $S_i(\xi)$ *are bounded continuous functions*

c. the functions $S_i(\xi)$ *are differentiable and nondecreasing*

d. the equations (4.20) describe the time activation of the layer $F_{X^k} \in R^n$ *for every* k *layer,* $k \leq N$

e. the functions $A_i^k(x_i^k)$ *and* $B_i^k(x_i^k)$ *are continuous for all* $1 \leq i \leq n_k$, *and* $1 \leq k \leq N$, *and* $A_i^k() \geq 0$ *(they are typical multiplicative functions. They correspond to the* i*-th neuron from layer* k*)*

f. it is assumed the boundedness of the integrals used in the Liapunov function below (The boundedness of the integral terms requires additional technical hypotheses to avoid pathologies, as Cohen and Grossberg[1983] discuss in [24])

then the function

$$V = -\frac{1}{2}\sum_{k=1}^{N}\sum_{k\neq l,l=1}^{N}\sum_{i=1}^{n_k}\sum_{j=1}^{n_l} w_{ij}^{kl} S_i(x_i^k) S_j(x_j^l) + \sum_{k=1}^{N}\sum_{i=1}^{n_k}\int_0^{x_i^k} S_i'(\Theta_i) B_i^k(\Theta_i) d\Theta_i \quad (4.21)$$

is a Liapunov system energy function and the system is globally stable.

Proof. The proof is similar to the above (Theorem 4.1) proof. The boundedness of the above function follows from the boundedness of the signal functions S_i stated in assumption (b) and from hypothesis (f) regarding the boundedness of the integrals used. The boundedness of the integral terms requires additional technical hypotheses to avoid pathologies, as Cohen and Grossberg [1983] discuss in [24]. For the thesis' purpose, the boundedness of the integral terms is assumed. By time-differentiating V, the time-derivative operator is distributed across the right-hand side of the above function. Then each term can be time-differentiated separately by computing $\frac{dV}{dt}$ and grouping after $S_i'(x_i^k)\frac{dx_i^k}{dt}$, so

$$\frac{dV}{dt} = \sum_{k=1}^{N}\sum_{i=1}^{n_k} S_i'(x_i^k)\frac{dx_i^k}{dt}[B_i^k(x_i^k) - \sum_{l\neq k,l=1}^{N}\sum_{j=1}^{n_l} S_j(x_j^l) w_{ij}^{kl}] \quad (4.22)$$

$$\frac{dV}{dt} = -\sum_{k=1}^{N}\sum_{i=1}^{n_k} S_i'(x_i^k)A_i^k(x_i^k)[B_i^k(x_i^k) - \sum_{l\neq k,l=1}^{N}\sum_{j=1}^{n_l} S_j(x_j^l)w_{ij}^{kl}]^2 \tag{4.23}$$

After computing the time differentiation and grouping, the conclusion

$$\frac{dV}{dt} \leq 0$$

proves that V is a Liapunov function so the system is globally stable. For the above system there is a reciprocal equilibrium relationship. The system is at equilibrium

$$(\frac{dV}{dt} = 0)$$

if and only if

$$(\frac{dx_i^k}{dt} = 0) \tag{4.24}$$

for all k, i and the stability is achieved in exponential time. As a point of interest, the theorem can be placed in an algebraic framework of the Cohen-Grossberg-Kosko theorem [4].

4.4 The Relationship Between Recurrent Neural Networks

As described above, equations (4.8) or (4.9) equations of the heteroassociative N layer network can be seen as a one layer, autoassociative network. We can consider it as a virtual layer which contains all the N layers, where its size is the sum of all the neurons from all the N layers of the heteroassociative network, $n_1 + n_2 + ... + n_N$. This autoassociative network has a matrix of connections between neurons formed from the heteroassociative W^{kl} matrices. This construction follows the convention $W^{kl} = (W^{kl})^T$ and $W^{kk} = 0$. The connection matrix of the autoassociative network is symmetric with the block main diagonal zero. The following lemma appears necessary:

Lemma 1. *There is a relationship between the two above described multilayer recurrent neural networks: any heteroassociative network described by the equations (4.8) or (4.9) has a corresponding one layer, autoassociative network (the virtual layer which contains all the N layers of the heteroassociative recurrent neural network) and vice versa.*

Motivation

The difference between autoassociative and heteroassociative networks is given by the network architecture highlighted in the multilayer case. Of course, mathematically the heteroassociative neural networks work with the Cartesian product $R_{n_1} \times R_{n_2} \times R_{n_3}...$, where $n_1, n_2, n_3...$ are the numbers of neurons in the first layer, second layer, third layer and so on..., but the autoassociative neural networks work with the space $R_{n_1+n_2+n_3...}$, where $n_1 + n_2 + n_3 + ...$ are the numbers of neurons in the autoassociative network. It is known that $R_{n_1} \times R_{n_2} \times R_{n_3}...$ is isomorphic with $R_{n_1+n_2+n_3...}$, so there is an isomorphism equivalent relationship between autoassociative and heteroassociative networks. The autoassociative network has the matrix of connections between neurons formed by the heteroassociative W^{kl} matrices. By construction there is the convention $W^{kl} = (W^{kl})^T$ and $W^{kk} = 0$. The connection matrix of the autoassociative network is symmetric with the block main diagonal zero. By means of this lemma, there is a second way to prove the above theorems based on the Cohen-Grossberg theorems [24], and vice versa.

Chapter 5

Discrete Case

5.1 General Additive Discrete Memory Model

Discrete activation models for recurrent neural networks have signal activation functions that take the form of threshold functions

$$S(x(t+1)) = \begin{cases} 1, if, x(t+1) > T \\ S(x(t)), if, x(t+1) = T \\ 0, if, x(t+1) < T \end{cases} \tag{5.1}$$

for an arbitrary real-valued threshold T. The index t indicates the discrete time step and $x(t), x(t+1)$ are the states at time t and $t+1$. The threshold function allows one to model complex asynchronous state-change patterns as defined in Kosko [4]. Different neurons can randomly choose any time whether to compare their current activation to their threshold or not. Not all the neurons from the same layer are activated in the same time. In BAM model any of the 2^n subsets of F_X neurons, or the 2^p subsets of F_Y neurons, can decide to change state. Randomly each neuron may decide whether to check the threshold conditions. Generally the network behaves as a vector stochastic process. Kosko [2] defines synchronous state change of a BAM as a deterministic update of an entire layer of neurons at a time. Then all the neurons from a layer are updated synchronously, they synchronously choose to compare their current activation to their

threshold. Another case is simple asynchrony defined in [4] as the case when only one neuron makes a stare-change decision at a time. In general state-change decisions are subset asynchronous, one subset of neurons per layer makes state-change decisions at a time. When the subset has just one neuron we have the simple asynchrony and when the set is all the neurons of the layer we have synchronous state change. As defined in [4] a BAM system (F_X, F_Y, M) is bidirectionally stable if all inputs converge to fixed-point equilibria. Further bidirectional stability appears as an example of global or absolute stability and as a dynamic equilibrium. The signal information flows back and forth in a bidirectional fixed point. Kosko describes the bidirectional equilibrium as a resonant state. Grossberg [1982] alternatively refers to this joint equilibration of neurons and synapses as adaptive resonance. Kosko proved [4] that every matrix M is bidirectionally stable for synchronous or asynchronous state change, for the discrete additive BAM with threshold signal functions, governed by the difference equations

$$x_i(t+1) = \sum_{j=1}^{p} S_j(y_j(t))m_{ij} + I_i \tag{5.2}$$

$$y_j(t+1) = \sum_{i=1}^{n} S_i(x_i(t))m_{ij} + J_j$$

where for two layer neural network with n neurons in the first layer, F_X and p neurons in the second layer F_Y, $x_i(t+1), x_i(t)$ are the states of i-th neuron of the layer F_X at time $t+1$ and t, $y_j(t+1), y_j(t)$ are the states of j-th neuron of the layer F_Y at time $t+1$ and t, m_{ij} are the weights of connection between i-th neuron of layer F_X and j-th neuron of layer F_Y. The above theorem can be generalized for any discrete neural system of the

form

$$x_i(t+1) = -A_i(x_i(t)) + \sum_{j=1}^{p} S_j(y_j(t))m_{ij} + I_i \tag{5.3}$$

$$y_j(t+1) = -B_j(y_j(t)) + \sum_{i=1}^{n} S_i(x_i(t))m_{ij} + J_j \tag{5.4}$$

for any connection matrix M. This model is inspired by the continuous additive model discussed in chapter 4.

Theorem 5.1 (General Additive Discrete BAM Theorem).

Given a discrete system of $n+p$ coupled equations which defines the additive activation model that interconnects layers F_x and F_y through the constant synaptic matrices M and M^T:

$$x_i(t+1) = -A_i(x_i(t)) + \sum_{j=1}^{p} S_j(y_j(t))m_{ij} + I_i \tag{5.5}$$

$$y_j(t+1) = -B_j(y_j(t)) + \sum_{i=1}^{n} S_i(x_i(t))m_{ij} + J_j \tag{5.6}$$

where $S_i()$ are the threshold activation functions described by

$$S(x(t+1)) = \begin{cases} 1, \, if, \, x(t+1) > U \\ S(x(t)), if, x(t+1) = U \\ 0, \, if, \, x(t+1) < U \end{cases} \tag{5.7}$$

and where U is the vector threshold for X and V is the vector threshold for Y, then the system is bidirectionally stable, for synchronous or asynchronous state changes, for all matrices M (under the assumption of some boundedness conditions).

Proof. This theorem is a particular case of Theorem 5.2, for N=2.

5.2 Definition of Discrete N - Layer Associative Memory

We now consider the following multilayer generalized discrete model. A discrete N layer associative memory, recurrent neural network can be described:

Given N layers, every neuron state is dependent on the activations of the neurons from the other layer. Each neuron has a connection to itself but not to any other neurons in the same layer. Every two layers are connected exactly as in BAM. Every neuron from one layer receives the activated signals of other neurons from other layers but not its own layer. The mathematical equations can be described as follows:

The Additive Activation Model

$$x_i^k(t+1) = -A_i^k(x_i^k(t)) + \sum_{l \neq k, l=1}^{N} \sum_{j=1}^{n_l} S_j(x_j^l(t)) w_{ij}^{kl} + I_i^k \tag{5.8}$$

where t is the discrete time; the system state at time $t+1$, $x_i^k(t+1)$ is computed based on the parameters of the system at time t; the upper index k is used for the layer k, where $1 \leq k \leq N$; i corresponds to the i-th neuron of layer k, which has n_k neurons, $(1 \leq i \leq n_k)$; j corresponds to the j-th neuron of the layer l, which has n_l neurons, $(1 \leq j \leq n_l)$; $x_i^k(t)$ is the state of i-th neuron from layer k at time t, w_{ij}^{kl} are the connection weights between two neurons from different layers, which in this notation denotes the weight of connection between the i-th neuron of layer k and the j-th neuron of layer l (when $k \neq l$ for the heteroassociative model); $A_i^k(x_i^k(t))$ are the functions which have different forms corresponding to the i-th neuron from layer k, and $S()$ are threshold function as defined in (5.1). As a remark it is necessary to point out that in

(5.8) the $A_i^k(x_i^k(t))$ are functions rather than coefficients as in the continuous model; this makes the above model distinct.

5.3 Theorem for Discrete N - Layer Associative Memory

Theorem 5.2 (The Discrete N Layer Associative Memory Theorem-for Additive Model).

Every discrete additive system of the form

$$x_i^k(t+1) = -A_i^k(x_i^k(t)) + \sum_{l\neq k, l=1}^{N} \sum_{j=1}^{n_l} S_j(x_j^l(t)) w_{ij}^{kl} + I_i^k \tag{5.9}$$

where

a. the matrices $w_{ij}^{kl} = w_{ji}^{lk}$ (by the notation convention and by the chosen heteroassociative model)

b. the functions $S_i(\xi)$ are threshold signal functions as described in (5.1), with U_i^k the threshold of the neuron i-th from the layer k

c. the equations (5.9) describe the time activation of every k layer, $F_{X^k} \in R^n$, $k \leq N$

d. it is assumed the boundedness of the second sum of the below Liapunov function (The boundedness of the second sum in (5.10) requires additional technical hypotheses to avoid pathologies)

then the function

$$V(t) = -\frac{1}{2} \sum_{k=1}^{N} \sum_{k\neq l, l=1}^{N} \sum_{i=1}^{n_k} \sum_{j=1}^{n_l} w_{ij}^{kl} S_i(x_i^k(t)) S_j(x_j^l k(t)) \tag{5.10}$$

$$+ \sum_{k=1}^{N} \sum_{i=1}^{n_k} \sum_{q=1}^{t} A_i^k(x_i^k(q-1))(S_i(x_i^k(q) - S_i(x_i^k(q-1))$$

$$-\sum_{k=1}^{N}\sum_{i=1}^{n_k} S(x_i^k(t))(I_i^k - U_i^k)$$

is a Liapunov system energy function and the system is globally stable, for every class of matrices W which verifies (a).

Proof. The proof uses the bounded Liapunov function given above. The boundedness of the function V follows from the boundedness of the threshold signal functions S_i and also from the hypothesis made at point d regarding the boundedness of the sum used. The boundedness of the second sum in (5.10) requires additional technical hypotheses to avoid pathologies. For the thesis' purpose, the second sum in (5.10) is assumed bounded. Every term is analyzed:

$$\sum_{k=1}^{N}\sum_{i=1}^{n_k}\sum_{q=1}^{t} A_i^k(S_i(x_i^q(t)) - S_i(x_i^{q-1}(t))x_i^{q-1}(t) \tag{5.11}$$

The above integral sum is assumed low bounded.

$$-\frac{1}{2}\sum_{k=1}^{N}\sum_{k\neq l,l=1}^{N}\sum_{i=1}^{n_k}\sum_{j=1}^{n_l} w_{ij}^{kl}S_i(x_i^k(t))S_j(x_j^l(t)) \geq -\frac{1}{2}\sum_{k=1}^{N}\sum_{l=1}^{N}\sum_{i=1}^{n_k}\sum_{k\neq l,j=1}^{n_l} ||w_{ij}^{kl}|| \tag{5.12}$$

$$-\sum_{k=1}^{N}\sum_{i=1}^{n_k} S(x_i^k(t))(I_i^k - U_i^k) \geq -\sum_{k=1}^{N}\sum_{i=1}^{n_k} ||I_i^k - U_i^k|| \tag{5.13}$$

We assume at least one signal neuron change, activation, from time k to time $k+1$, which means there is i for which $\Delta S_i(x_i) \neq 0$. This allows us to model the synchronous, simple asynchronous, and subset asynchronous changes. Only the neurons from the same layer can be active, changing the state t to $t+1$. The variation of the function V, $\Delta V = V(t+1) - V(t)$ differs from zero because of the changes in one of the layers. Suppose that the change occurs only in layer F_{X_f}, the layer f is activated, $1 \leq f \leq N$, then the

above function can be written in the following way:

$$V(t) = -\frac{1}{2}\sum_{k=1}^{N}\sum_{k\neq l,l=1}^{N}\sum_{i=1}^{n_k}\sum_{j=1}^{n_l} w_{ij}^{kl} S_i(x_i^k(t)) S_j(x_j^l k(t)) \tag{5.14}$$

$$+\sum_{k=1}^{N}\sum_{i=1}^{n_k}\sum_{q=1}^{t} A_i^k(x_i^k(q-1))(S_i(x_i^k(q) - S_i(x_i^k(q-1))$$

$$-\sum_{k=1}^{N}\sum_{i=1}^{n_k} S(x_i^k(t))(I_i^k - U_i^k)$$

$$V(t+1) = -\frac{1}{2}\sum_{f\neq l,l=1}^{N}\sum_{active,layer,f,i=1}^{n_f}\sum_{j=1}^{n_l} w_{ij}^{fl} S_i(x_i^f(t+1)) S_j(x_j^l f(t)) \tag{5.15}$$

$$-\frac{1}{2}\sum_{k=1,k\neq f}^{N}\sum_{k\neq l,f\neq l,l=1}^{N}\sum_{i=1}^{n_k}\sum_{j=1}^{n_l} w_{ij}^{kl} S_i(x_i^k(t)) S_j(x_j^l k(t))$$

$$+\sum_{active,layer,f,i=1}^{n_f}\sum_{q=1}^{t+1} A_i^f(x_i^f(q-1))(S_i(x_i^f(q) - S_i(x_i^f(q-1))$$

$$+\sum_{k\neq f,k=1}^{N}\sum_{i=1}^{n_k}\sum_{q=1}^{t+1} A_i^k(x_i^k(q-1))(S_i(x_i^k(q) - S_i(x_i^k(q-1))$$

$$-\sum_{k=1}^{N}\sum_{i=1}^{n_k} S(x_i^k(t+1))(I_i^k - U_i^k)$$

We wrote above the general case when the layer f is activated, $1 \leq f \leq N$.

$$\Delta V = -\sum_{f\neq l,l=1}^{N}\sum_{i=1}^{n_f}\sum_{j=1}^{n_l} w_{ij}^{fl}(S_i(x_i^f(t+1)) - S_i(x_i^f(t)))S_j(x_j^l f(t))) \tag{5.16}$$

$$+\sum_{active,layer,f,i=1}^{n_f}\sum_{q=1}^{t+1} A_i^f(x_i^f(t))(S_i(x_i^f(t+1)) - S_i(x_i^f(t)))$$

$$-\sum_{i=1}^{n_f}(S(x_i^f(t+1)) - S(x_i^f(t)))(I_i^f - U_i^f)$$

$$\Delta V = -\sum_{i=1}^{n_f}[S_i(x_i^f(t+1)) - S_i(x_i^f(t))](x_i^f(t+1) - U_i^f) \leq 0 \qquad (5.17)$$

along trajectories. It will be proven that all the terms of the above sum are nonpositive. We need to check only the cases (for every i, $1 \leq i \leq n_f$) :

First, $\Delta S_i(x_i) > 0$, then

$$\Delta S_i(x_i) = S_i^{k+1} - S_i^k = 1 - 0 \qquad (5.18)$$

From the threshold law this implies $x_i^f(t+1) > U_i^f$, so the above product is

$$\Delta S_i(x_i^f(t))(x_i^f(t+1) - U_i^f) > 0$$

Second, suppose $\Delta S_i(x_i) < 0$, then

$$\Delta S_i(x_i) = S_i^{k+1} - S_i^k = 0 - 1 \qquad (5.19)$$

From the threshold law this implies $x_i^f(t+1) < U_i^f$, so the above product is positive

$$\Delta S_i(x_i^k)(x_i^{k+1} - U_i) > 0$$

When $\Delta S_i(x_i) = 0$ this means the neuron i-th of the layer f is not active. Every activation (the state change from t to $t+1$) can be considered analogously. This example is for the threshold signal functions described in (5.1) for any $2^{n_f} - 1$ possible subsets of neurons from layer F_{X_f} which activate, or change state, at time $t+1$. The above argument was made for an arbitrary layer f. The same argument produces the proof for the case in which the activation is from any other layer of the N - layer network. So $\Delta V \leq 0$ for every state change, which proves that the above function is a Liapunov function so the system is globally stable.

Chapter 6

Conclusions and Recommendations

This thesis develops a new point of view about multilayer recurrent neural networks. The multilayer recurrent neural networks are mathematically described by the dynamic equations found in chapters 4 and 5. The thesis states and proves a chain of original theorems concerning multilayer neural networks for different architectures. Historically, the idea of recurrent neural network appeared first with Hopfield's single additive architecture [16]. Around 1985, Cohen and Grossberg developed the multiplicative single layer structure known as the Grossberg model [24] and proved the Cohen-Grossberg Theorem regarding global stability. The next step was made by Kosko with his BAM model as well as with a set of studies of two layer recurrent neural networks. This thesis introduces and develops the original study of the N - layer recurrent neural networks from different points of view.

Some observations regarding the N - layer structure are as follows: Section 4.4 proves the isomorphic relationship between one layer and N - layer neural networks. This leads to following question: Why are we studying N - layer neural networks? The differentiation of N -layers or N - groups of neurons allows adaptive and highly correlated structures which are not equivalent to single layer, autoassociative models. The differentiation by layers is very important in adaptive and discrete structures because

the neurons from the same layer are not modifying each other; only a neuron from a different layer can affect a given neuron's activation state. Theorems from chapters 4 and 5 prove different system global stabilities based on the system Liapunov energy function. However they do not show the exact convergent solutions or the stable points. These theorems only prove the global stability of the given systems. It is just a beginning of a long study which can be developed based on N - layer recurrent neural networks. This subject will be investigated in future work. Intuitively, these dynamic systems have analogies to the short-term and long-term memories of biological neural networks:

1. In N layer recurrent neural networks short term memory is represented by the activation state vector X^k of each layer k,
2. Long term memory is represented by the matrix of synapses, W.

Some activation patterns may represent auditory, olfactory, tactile or visual patterns as Kosko speculated in [4]. As exemplified when learning poetry, it is hard to remember after a few days all the words completely, although some fragments are still remembered. This might conceivably be modelled by updated activation patterns of neural structure competing and evolving, conform to their dynamics. Further, dynamic neural systems could conceivably model the process of thought, or human intelligence [4]. The N layer recurrent neural generalized model could conceivably be applied in this way to the development of models of biological neural networks. Another observation is that all the neuronal dynamic systems analyzed in chapters 4 and 5 do not include time as an individual variable. As a result, all of these models can be classified as autonomous dynamic systems. It is known that autonomous systems are usually easier to analyze

than nonautonomus systems.

All the studies presented in chapters 4 and 5 prove global stability. Global stability means convergence to fixed points for all initial conditions and parameters. It is important to mention that global stability is a joint equilibrium of the neuronal activation and synaptic dynamics. As it is known that neural activation changes faster than synaptic weights [4] this means a reciprocal balance between the two dynamics. The stabilization at the neuronal level is faster than at the synaptic level; neuronal dynamic activations stabilize faster than synaptic learning. This leads to a stability dilemma in most adaptive recurrent structures. The neuronal activation is based on activity patterns while the synaptic sequence weights learn from themself. Learning destabilizes the neural activation by moving the state in the global stability space.

There is a relationship between additive and multiplicative N layer recurrent neural networks. If in the additive model the constant terms, I_i^k are zero, it becomes a particular case of the multiplicative model obtained by taking $A_i^k(x_i^k) = 1$ and $B_i^k(x_i^k) = b_i^k x_i^k$. So the symmetry of results and statements for both cases is motivated. However the proofs are based on different Liapunov functions.

The boundedness proofs are based on well defined hypotheses (e.g. the signal to be positive). An important topic for future study is the extension of the above theorems to stochastic neural processing under noise. This thesis establishes many theoretical results for continuous and discrete multilayer recurrent neural networks, for different architectures. Other topics that need to be investigated are random adaptive learning laws. This

study also provides opportunities to further develop Hopfield's optimization of recurrent neural networks.

References

References

1. Kosko, B., "Adaptive Bidirectional Associative Memories" *Applied Optics*, vol. 26, no. 23. 4947-4960, December 1987.
2. Kosko, B.,"Bidirectional Associative Memories," *IEEE Transaction on Systems, Man, and Cybernetics*, vol. SMC-18, 49-60, January 1988.
3. Kosko, B., "Feedback Stability and Unsupervised Learning," *Proceeding of the 1988 IEEE International Conference on Neural Networks*, vol. I, no. 1, 141-152, July 1988.
4. Kosko, B. (1992), *Neural Networks and Fuzzy Systems*, Prentice-Hall, Englewood Cliffs, NJ 07632.
5. Cichochi and Unbehauen (1993), *Neural Network for Optimization and Signal Processing*, Teubner-Verlag: Trinity, ISBN 0 471 93010 5.
6. Demuth, H. and Beale, M. (1998), *Neural Network TOOLBOX User's Guide. For use with MATLAB*, Natick, Mass: The MathWorks Inc.
7. Freeman J.A. and Skapura D.M. (1991), *Neural Networks. Algorithms, Applications, and Programming Techniques*, Reading, Mass: Addison-Wesley, ISBN 0201513765.
8. Hassoun, Mohamad H. (1995), *Fundamentals of Artificial Neural Networks*, Cambridge: The MIT Press, ISBN 0-262-08239-X.
9. Haykin, Simon (1999), *Neural Networks - a Comprehensive Foundation*, New Jersey:Prentice Hall, 2nd edition, ISBN 0-13-273350-1.

10. Hagan, Martin T., Demuth, H. and Beale, M. (1996), *Neural Network Design*, Boston: PWS Publishing.

11. Hertz, John A., Krogh, Anders S. and Palmer, Richard G. (1991), *Introduction to the Theory of Neural Computation*, Redwood City, Calif: Addison-Wesley Pub. Co., ISBN 0-201-51560-1.

12. Kosko, B. (1991), *Neural Networks for Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, ISBN 0-13-617390-X.

13. Sarle, W.S. (1998), *Neural Network FAQ* [online] Newsgroup: comp.ai.neural-nets, Available from: ftp://ftp.sas.com/pub/neural/FAQ.html [Accessed 1 June 2001].

14. Hoppensteadt, Frank and Izhikevich, Eugene (1997), *Weakly Connected Neural Networks,* NJ: Springer, ISBN 0-387-94948-8.

15. Kosko, B. "Differential Competitive Learning for Centroid Estimation and Phoneme Recognition", *IEEE Trans on Neural Networks*, vol. 2, no. 1, pp 118-124, January 1991.

16. Hopfield, J.J "Neural networks and physical systems with emergent collective computational abilities", in *Proc. Nat. Acad. Sci.*, USA, vol. 79, pp 2554-2558, 1982.

17. Xu Z., Leung Y. and He X., "Asymmetric bidirectional associative memories", *IEEE Trans, Syst. Man, Cybern.*, vol. 24, pp. 1558-1564, 1994.

18. Boyd, S., Ghaoui, L., Feron, E. and Balakrishnan, V. (1994), *Linear Matrix Inequalities in Systems and Control Theory*, Philadelphia, PA: SIAM, 1994.

19. Wu, Y. and Pados, D. A., "A feedforward bidirectional associative memory", *IEEE Trans Neural Networks*, vol. 11, pp. 859-866, 2000.

20. Fang, Y. and Kincaid, T.G., "Stability analysis of dynamic neural networks", *IEEE Transactions on Neural Networks*, vol. 7, no. 4, July 1996.

21. Kosko, B., "Global Stability of Generalized Additive Fuzzy Systems," *IEEE Trans Systems, Man, and Cybernetics-PartC Applications and Reviews*, vol. 28, no. 3, August 1998.

22. Kosko, B., "Unsupervised Learning in Noise," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 44-57, March 1990.

23. Vidyasagar, M. (1993), *Nonlinear Systems Analysis*, Englewood Cliffs, NY: Prentice Hall, ISBN 0136234631.

24. Cohen, M.A. and Grossberg, S., "Absolute stability of global pattern formation and parallel memory storage by competitive neural networks", *IEEE Trans. Syst., Man. Cybern.*, vol. 13, pp. 815-826, 1983.

25. Chen, Tianping and Amari, Shun Ichi, "Stability in Asymmetric Hopfield Networks", *IEEE Trans. Neural Networks*, vol. 12, no.1, January 2001.

26. La Salle, Joseph and Lefschetz, Solomon (1961), *Stability by Liapunov's Direct Method,* New York: Academic Press Inc.

27. Kosko, B., "Structural Stability of Unsupervised Learning in Feedback Neural Networks," *IEEE Trans on Automatic Control*, vol. 36, No. 7, July 1991.

28. Kosko, B., "Stochastic competitive Learning", *IEEE Trans Neural Network*, vol. 2, no. 5, Sept 1991.

Vita

Rodica Ion Waivio was born in Romania on August 4, 1969. She attended schools in the public systems of Bucharest, the Computer Science College, were she graduated in June, 1988. She entered the Department of Automation and Computers, Bucharest Polytechnic University, Romania, based on an competitive examination in October 1988, from where in June 1993 she received the Diploma. Her desire of wide study was fulfill by continuation of computer study research at Department of Informatics, Bucharest University where in 1996 she received the Diploma of study in Informatics. Further in 1996 she was admitted for a Master in Management of Databases at The Bucharest Economic Academy, Faculty of Informatics System for Financial and Accounting Administration, from where she graduated in June 1997.

She is presently following a Ph.D Program and Research in Neural Networks.