4-16-2012

General Neural Networks Dynamics are a Superposition of Gradient-like and Hamiltonian-like Systems

Chaouki T. Abdallah

James W. Howse

Gregory L. Heileman

Department of Electrical and
Computer Engineering

College of Engineering

University of New Mexico

General Neural Networks Dynamics are a Superposition of Gradient-like and Hamiltonian-like Systems

James W. Howse, Chaouki T. Abdallah, and Gregory L. Heileman
Department of Electrical and Computer Engineering
University of New Mexico, Albuquerque, NM 87131

UNM Technical Report No. EECE94–001

Initial Draft Completed: January 21, 1994
Major Revision Completed: September 9, 1994
Current Date: October 5, 1994

Abstract

This report presents a formalism that enables the dynamics of a broad class of neural networks to be understood. A number of previous works have analyzed the Lyapunov stability of neural network models. This type of analysis shows that the excursion of the solutions from a stable point is bounded. The purpose of this work is to present a model of the dynamics that also describes the phase space behavior as well as the structural stability of the system. This is achieved by writing the general equations of the neural network dynamics as the sum of gradient-like and Hamiltonian-like systems. In this paper some important properties of both gradient-like and Hamiltonian-like systems are developed and then it is demonstrated that a broad class of neural network models are expressible in this form. [†]

[†]Acknowledgments: This research was supported by a grant from Boeing Computer Services under Contract W–300445.

Chapter 1

Introduction

There are many formalisms that can be used to analyze neural networks in which the connection matrix is symmetric. In such cases it is typical to construct a Lyapunov or energy function and then analyze the network behavior by observing that the system state must travel along the surface defined by the energy function. This approach can establish conditions which guarantee that the node activities and connection weights converge to an equilibrium state of the network. We recently proposed [10] a formalism which allows any system whose dynamics can be put in the proper form, to be analyzed with respect to its Lyapunov stability, phase space behavior, and structural stability. The dynamical systems which have the proper form were called gradient-like systems and several theorems were presented which explain their behavior. We further demonstrated that a large class of neural networks with symmetric connection matrices could be expressed in this form.

The difficulty with this approach that it does not address networks which have asymmetric connection matrices. There are many ideas concerning ways to exploit the asymmetry of the weight matrix. It is suggested in [8] that memory retrieval time is decreased by asymmetric connections because the number of spurious states is decreased. In [14] it is proposed that asymmetry is important for the learning process because it increases the likelihood that the retrieval states, as opposed to the spurious states, will be enhanced by Hebbian type learning mechanisms. The learning of temporal sequences using asymmetric weights is discussed in [7]. Also some weight update rules such as gated learning, which is used in instars, outstars, and ART [1], naturally produce asymmetric weights. For such networks there is in general no energy function, nor can the dynamics be expressed as a gradient-like system. To our knowledge no one has proposed

a formalism which allows the dynamic behavior of such networks to be studied. In this paper we present such a formalism and discuss some ways in which it can be used to analyze network behavior.

Specifically we propose to represent the total system dynamics as the sum of a gradient-like system, a Hamiltonian system, and an external input term. Since most neural network models are autonomous (i.e. time independent) systems, except for the external inputs, the first two terms will be time independent. The behavior of autonomous gradient-like and Hamiltonian systems is a well studied mathematical topic. Intuitively, the behavior of the system can be studied by observing which of the three terms is largest at any given time interval. Some ways to judge the relative sizes of the terms will be suggested and ways to use these methods to determine parameter values in the network will be suggested.

Chapter 2

General Formulation

Consider a neural network with n nodes and m weights. The activation of the ith node is given by x_i, and the value of the weight *to* the ith node, *from* the jth node is given by c_{ij}. Following the form in [11] the dynamics of many such networks can be written as

$$\begin{aligned} \dot{x}_i &= -a_i(x_i)\left[b_i(x_i) - \sum_{j=1}^{n} c_{ij} d_j(x_j)\right], \quad i = 1, \ \ldots \ , n, \\ \dot{c}_{ij} &= f_{ij}(x_i, x_j, c_{ij}), \quad i, j \in \{1, \ \ldots \ , n\}. \end{aligned} \tag{2.1}$$

In [12] it is shown that any dynamical system which can be defined by $(\mathcal{M}, \mathcal{X})$, where $\mathcal{M}$ is an N-dimensional differentiable manifold and $\mathcal{X}$ is an rth order continuously differentiable (i.e. C^r) vector field, can be decomposed into the sum of one gradient system and $N-1$ Hamiltonian systems. Further it was proved that if either the Riemannian metric or the symplectic form can be specifically chosen to match the vector field $\mathcal{X}$, then the system dynamics can be decomposed into one gradient system and *one* Hamiltonian system, regardless of the dimension of $\mathcal{M}$. This will be true for systems in the form of Equation (2.1) for certain choices of the functions $f_{ij}(x_i, x_j, c_{ij})$. Therefore the total dynamics of this class of networks can be represented as the sum of a time independent Gradient-like system, a time independent Hamiltonian-like system, and a time dependent system. The purpose of the time dependent system is *only* to represent the inputs to the network. So the general form of the system is

$$\dot{\mathbf{u}} = -\mathbf{P}(\mathbf{u})\nabla_{\mathbf{u}} V(\mathbf{u}) + \mathbf{Q}(\mathbf{u})\nabla_{\mathbf{u}} H(\mathbf{u}) + \mathbf{R}(\mathbf{u})\nabla_{\mathbf{u}} S(\mathbf{u}, t), \tag{2.2}$$

where the state vector $\boldsymbol{u}$ is given by $\boldsymbol{u} = [\boldsymbol{x}, \boldsymbol{C}_1, \boldsymbol{C}_2, \ \ldots \ , \boldsymbol{C}_n]^T$ with $\boldsymbol{C}_k$ denoting the kth row of the weight matrix. Note that only *variable* weights need to be put into the state vector $\boldsymbol{u}$,

constant weights should not be included. In Equation (2.2) the first term (i.e. leftmost on the right of the equal sign) is the gradient-like term, the second term is the Hamiltonian term, and the third term is the external input term. The potential terms $V(\boldsymbol{u})$, $H(\boldsymbol{u})$, and $S(\boldsymbol{u}, t)$ must all be twice continuously differentiable (i.e. C^2). The matrix $\boldsymbol{P}(\boldsymbol{u})$ must be symmetric (i.e. $\boldsymbol{P} = \boldsymbol{P}^T$) and positive definite, and the matrix $\boldsymbol{Q}(\boldsymbol{u})$ must be anti-symmetric (i.e. $\boldsymbol{Q} = -\boldsymbol{Q}^T$) and satisfy the Jacobi identity, for all values of $\boldsymbol{u}$. The Jacobi identity for the matrix $\boldsymbol{Q}(\boldsymbol{u})$ is

$$q_{li}\frac{\partial q_{jk}}{\partial u_l} + q_{lj}\frac{\partial q_{ki}}{\partial u_l} + q_{lk}\frac{\partial q_{ij}}{\partial u_l} = 0, \tag{2.3}$$

where i, j, k, and l take on all possible values. This description of neural dynamics generalizes the ideas in [13] by including the weight dynamics in the state vector and also by incorporating a term for the inputs.

This approach can be used to analyze asymmetrically connected networks since the behavior of the system in Equation (2.2) is determined by the largest term in the differential equation. This means that if the gradient term of the dynamics is much larger than the Hamiltonian or input terms, then the system will behave like a gradient system, it will always converge to one of the equilibria or go to infinity. Likewise if the Hamiltonian term is much larger than the gradient or input terms then the system will have no attractors and only divergent or periodic behavior is possible. If the gradient and Hamiltonian parts are of comparable size then in general it is very difficult to make any statements about the overall behavior of the network. Because the behavior of gradient and Hamiltonian systems is well understood it is advantageous to formulate the input term as either a gradient term, a Hamiltonian term or some combination of the two. If all inputs are strictly nonincreasing then the input term can easily be cast as a gradient system. If all inputs are periodic then the input term can be written as a Hamiltonian system provided that there are as many variable weights as there are nodes.

One way to judge the relative size of the gradient-like and Hamiltonian terms is to consider the total time derivatives of the potential functions $V(\boldsymbol{u})$ and $H(\boldsymbol{u})$. Ignoring the input term for the moment, these quantities are

$$\begin{aligned}
\dot{V}(\boldsymbol{u}) &= -\left[\nabla_{\boldsymbol{u}} V(\boldsymbol{u})\right]^T \boldsymbol{P}(\boldsymbol{u}) \left[\nabla_{\boldsymbol{u}} V(\boldsymbol{u})\right] + \left[\nabla_{\boldsymbol{u}} V(\boldsymbol{u})\right]^T \boldsymbol{Q}(\boldsymbol{u}) \left[\nabla_{\boldsymbol{u}} H(\boldsymbol{u})\right], \\
\dot{H}(\boldsymbol{u}) &= \left[\nabla_{\boldsymbol{u}} H(\boldsymbol{u})\right]^T \boldsymbol{Q}(\boldsymbol{u}) \left[\nabla_{\boldsymbol{u}} H(\boldsymbol{u})\right] - \left[\nabla_{\boldsymbol{u}} H(\boldsymbol{u})\right]^T \boldsymbol{P}(\boldsymbol{u}) \left[\nabla_{\boldsymbol{u}} V(\boldsymbol{u})\right].
\end{aligned} \tag{2.4}$$

If the system had only gradient-like dynamics then the term $-\left[\nabla_{\boldsymbol{u}} V(\boldsymbol{u})\right]^T \boldsymbol{P}(\boldsymbol{u}) \left[\nabla_{\boldsymbol{u}} V(\boldsymbol{u})\right]$ in the equation for $\dot{V}(\boldsymbol{u})$ would be the only term to occur. Since the matrix $\boldsymbol{P}(\boldsymbol{u})$ is positive definite, this term is always nonincreasing. If, on the other hand, the system had only Hamiltonian dynamics then the term $\left[\nabla_{\boldsymbol{u}} H(\boldsymbol{u})\right]^T \boldsymbol{Q}(\boldsymbol{u}) \left[\nabla_{\boldsymbol{u}} H(\boldsymbol{u})\right]$ in the equation for $\dot{H}(\boldsymbol{u})$ would be the only term.

Because the matrix $\boldsymbol{Q}(\boldsymbol{u})$ is anti-symmetric, this term is zero. This suggests that one way to analyze the behavior of the system is by studying the cross terms in the equations of $\dot{V}(\boldsymbol{u})$ and $\dot{H}(\boldsymbol{u})$.

First consider the cross term of $\dot{H}(\boldsymbol{u})$. For a purely Hamiltonian system there will always be at least one function which is a constant along some subset of the solutions (i.e. trajectories). For the autonomous case, the Hamiltonian potential $H(\boldsymbol{u})$ is guaranteed to be one such function. Call the set of functions which satisfy this criteria $\{\Phi\}$, and for the kth member of $\{\Phi\}$ call the corresponding set of trajectories $\{\Psi^k\}$. For each member ϕ^k of $\{\Phi\}$, integrate the cross term of $\dot{H}(\boldsymbol{u})$ over a trajectory ψ_l^k for which ϕ^k is constant. Look for closed trajectories which satisfy the condition

$$\oint_{\psi_l^k} \left\{ - \left[\nabla_{\boldsymbol{u}} H(\boldsymbol{u})\right]^T \boldsymbol{P}(\boldsymbol{u}) \left[\nabla_{\boldsymbol{u}} V(\boldsymbol{u})\right] \right\} \, dt = 0. \tag{2.5}$$

The set of closed trajectories satisfying this condition for a given constant of the motion ϕ_k will supply information about the dimension and location of stable persistent motions, such as limit cycles, of the system. It will also show how the network parameters effect these motions. This analysis is discussed in greater detail in [2]. It can similarly be claimed that trajectories for which $V(\boldsymbol{u})$ is nonincreasing and which satisfy

$$\oint_{v_l^k} \left\{ \left[\nabla_{\boldsymbol{u}} V(\boldsymbol{u})\right]^T \boldsymbol{Q}(\boldsymbol{u}) \left[\nabla_{\boldsymbol{u}} H(\boldsymbol{u})\right] \right\} \, dt \leq 0 \tag{2.6}$$

will give comparable information about the equilibrium points of the system.

Chapter 3

Review and Extensions

The purpose of this section is to review the properties of gradient-like systems which were formulated in [10] and then show that the present formalism can be used to extend those results. It was shown in [10] that many networks which use Hebbian learning can be written as gradient-like systems. This formulation will be reviewed and then it will be shown that this result can be generalized by introducing a Hamiltonian term into the dynamic equations. Lastly it will be demonstrated that the introduction of the Hamiltonian term allows additional learning rules to be incorporated into the present model.

3.1 Review of Gradient-Like Dynamics

A gradient-like system is one in which the time derivative of the states $\dot{\boldsymbol{u}}$ is equal to the product of the gradient of a scalar function $V(\boldsymbol{u})$ and a symmetric positive definite matrix $\boldsymbol{P}(\boldsymbol{u})$. These dynamics are described by the equation

$$\dot{\boldsymbol{u}} = -\boldsymbol{P}(\boldsymbol{u}) \left[\nabla_{\boldsymbol{u}} V(\boldsymbol{u})\right]. \tag{3.1}$$

The function $V(\boldsymbol{u})$ is a scalar function referred to as the gradient potential function. It is a mapping of the form $V : \mathcal{U} \rightarrow \mathbb{R}$, where $\mathcal{U} \subset \mathbb{R}^N$ is an open set, which is required to be twice continuously differentiable. The matrix $\boldsymbol{P}(\boldsymbol{u})$ must be symmetric and positive definite (i.e. $\boldsymbol{y}^T \boldsymbol{P}(\boldsymbol{x}) \boldsymbol{y} > 0 \ \forall \ \boldsymbol{y} \neq \boldsymbol{0}$) for all values of $\boldsymbol{u}$.

Conceptually the function $V(\boldsymbol{u})$ defines a surface in the phase space of the system. All of the

trajectories of the network must move along this surface. The matrix $\boldsymbol{P}(\boldsymbol{u})$ specifies the "laws of motion" that the trajectories must obey in moving along the surface defined by $V(\boldsymbol{u})$. Since $\boldsymbol{P}(\boldsymbol{u})$ is positive definite for all values of $\boldsymbol{u}$, the trajectories always move downhill along $V(\boldsymbol{u})$ (i.e. toward smaller values of $V(\boldsymbol{u})$). If a trajectory reaches a point where the slope of $V(\boldsymbol{u})$ is zero in *any* direction, then the trajectory remains at that point thereafter.

This intuition was formalized in [10] through a series of proofs which characterize the behavior of gradient-like systems. It was shown that every isolated local minima of $V(\boldsymbol{u})$ is an asymptotically stable equilibrium point of the network. This does *not* guarantee that every trajectory will converge to an equilibrium point. In order for that to occur the set

$$\mathcal{N}_c = \{\boldsymbol{u} \in \mathbb{R}^N : V(\boldsymbol{u}) \leq c\} \tag{3.2}$$

must be compact (i.e. closed and bounded) for every $c \in \mathbb{R}$. This is guaranteed to be true if $V(\boldsymbol{u})$ is bounded below (i.e. $V(\boldsymbol{u}) \geq \delta \ \ \forall \ \ \boldsymbol{u} \in \mathbb{R}^N$), and radially unbounded (i.e. $V(\boldsymbol{u}) \to \infty$ as $\|\boldsymbol{u}\| \to \infty$).

As the intuitive description of gradient-like dynamics implies, the phase space behavior of such systems is quite simple. Since the trajectories can only remain constant at the equilibrium points and must move toward smaller values of $V(\boldsymbol{u})$ at all other points, the only recurrent trajectories are the equilibria themselves. A *recurrent trajectory* is one that returns to within an arbitrarily small neighborhood of its starting point at some later time. Since almost all trajectories of a gradient-like system must move down hill along the surface defined by $V(\boldsymbol{u})$, almost all trajectories end up at a stable equilibrium point or go to infinity. The exception to this is those few trajectories which terminate at a saddle point. Likewise all trajectories must begin at an unstable equilibrium point or at infinity. Furthermore, in gradient-like systems only three types of equilibria are possible, stable points, unstable points, and saddle points. In the next section we will review the formulation of networks which use Hebbian learning as gradient-like systems.

3.2 Review of Hebbian Learning Networks

One of the main results in [10] was that many neural networks which use Hebbian weight update can be written as gradient-like systems. Consider a neural network with n nodes and $m \leq n^2$ *variable* weights. The activation of the ith node is given by x_i, and the value of the weight *to* the ith node, *from* the jth node is given by c_{ij}. Following [11], the general form for this system is the

set of differential equations

$$\dot{x}_i = -a_i(x_i)\left[b_i(x_i) - \sum_{j=1}^{n} c_{ij}\, d_j(x_j)\right], \quad i = 1, \ \ldots\ , n, \tag{3.3a}$$

$$\dot{c}_{ij} = -\gamma_{ij}\, c_{ij} + \lambda_{ij}\, d_i(x_i)\, d_j(x_j), \quad i, j \in \{1, \ \ldots\ , n\}. \tag{3.3b}$$

In equation (3.3a), $a_i(x_i)$ are the elements of an $(n \times n)$ *diagonal* matrix $\boldsymbol{A}(\boldsymbol{x})$, $b_i(x_i)$ are the elements of the n-dimensional vector $\boldsymbol{b}(\boldsymbol{x})$, and $d_i(x_i)$ is the output function of the ith node. In equation (3.3b), the term $-\gamma_{ij}\, c_{ij}$ is a passive decay term where γ_{ij} is a constant which determines the decay rate. The constant λ_{ij} determines the growth rate of the connection weight c_{ij} if the nodes at both ends of the connection are active. The matrices containing all such constants are $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}$ respectively.

In order to instantiate the dynamics described by equation (3.3) into the gradient-like system of equation (3.1), define the state vector $\boldsymbol{u}$ as

$$\boldsymbol{u} = [x_1, \ x_2, \ x_3, \ \ldots\ , \ x_n, \ c_{11}, \ c_{12}, \ c_{13}, \ \ldots\ , \ c_{nn}]^T. \tag{3.4}$$

Note that $\boldsymbol{u}$ is an $(n+m)$-dimensional vector. Let the gradient potential function be given by

$$V(\boldsymbol{u}) = -\frac{1}{2}\boldsymbol{d}(\boldsymbol{x})^T \boldsymbol{C}\, \boldsymbol{d}(\boldsymbol{x}) + \left[\sum_{k=1}^{n} \int_0^{x_k} d_k'(\zeta_k)\, b_k(\zeta_k)\, d\zeta_k\right] + \frac{1}{4}\mathbf{1}^T \left[\boldsymbol{\Gamma} \circ \boldsymbol{\Lambda}^{-1} \circ \boldsymbol{C} \circ \boldsymbol{C}\right] \mathbf{1}. \tag{3.5}$$

In equation (3.5) note that $\mathbf{1}$ is a n-dimensional vector whose elements are all 1. Also the operation $\circ$ denotes the *Schur* product, which is defined as $[\boldsymbol{A} \circ \boldsymbol{B}]_{ij} = a_{ij} b_{ij}$. Since $V(\boldsymbol{u})$ must be twice continuously differentiable, the same requirement must hold for the output functions $d_i(x_i)$. Choose the matrix $\boldsymbol{P}(\boldsymbol{u})$ to be

$$\boldsymbol{P}(\boldsymbol{u}) = \Delta\left[\frac{a_1(x_1)}{d_1'(x_1)}, \ \ldots\ , \ \frac{a_n(x_n)}{d_n'(x_n)}, \ 2\lambda_{11}, \ 2\lambda_{12}, \ 2\lambda_{13}, \ \ldots\ , \ 2\lambda_{nn}\right]. \tag{3.6}$$

The notation $\Delta[h_{11}, \ h_{22}, \ \ldots\ , \ h_{qq}]$ will be used to denote a $(q \times q)$ diagonal matrix with the listed elements along the diagonal. In order for $\boldsymbol{P}(\boldsymbol{u})$ to be positive definite, the functions $a_i(x_i)$ must be positive definite (i.e. $a_i(x_i) > 0 \ \ \forall \ \ x_i \neq 0$), the output functions $d_i(x_i)$ must be monotonically increasing (i.e. $d_i'(x_i) > 0$), and all elements of $\boldsymbol{\Lambda}$ must be positive. From equation (3.5) it is

apparent that the gradient $\nabla_{\boldsymbol{u}} V(\boldsymbol{u})$ is

$$\nabla_{\boldsymbol{u}} V(\boldsymbol{u}) = \begin{pmatrix} d_1'(x_1)\left[b_1(x_1) - \sum_{j=1}^{n} \frac{1}{2}\left(c_{1j} + c_{j1}\right) d_j(x_j)\right] \\ \vdots \\ d_n'(x_n)\left[b_n(x_n) - \sum_{j=1}^{n} \frac{1}{2}\left(c_{nj} + c_{jn}\right) d_j(x_j)\right] \\ \frac{1}{2}\frac{\gamma_{11}}{\lambda_{11}} c_{11} - \frac{1}{2} d_1(x_1)\, d_1(x_1) \\ \vdots \\ \frac{1}{2}\frac{\gamma_{nn}}{\lambda_{nn}} c_{nn} - \frac{1}{2} d_n(x_n)\, d_n(x_n) \end{pmatrix}. \tag{3.7}$$

Note that $\nabla_{\boldsymbol{u}} V(\boldsymbol{u})$ is an $(n+m)$-dimensional vector.

It can be seen from equation (3.7) that there are two classes of networks whose gradient potential function is given by equation (3.5) which have gradient-like dynamics. The first class are those systems in which the weight matrix $\boldsymbol{C}$ learned by the Hebbian rule is symmetric. This will occur if the matrices $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}$ are symmetric, and the initial conditions for c_{ij} and c_{ji} are the same. A reasonable physical interpretation of this situation is that there is a single bidirectional link between any two nodes, rather than two unidirectional ones. The second class are networks in which the learned weight matrix $\boldsymbol{C}$ is asymmetric, but only the symmetric part of the weight matrix is used to calculate the node activations $\boldsymbol{x}$. It is shown in [9] that this model can treat both additive and multiplicative node activation dynamics [5], and can be extended to incorporate anti-Hebbian learning [3], and higher order networks [4, 15] which use Hebbian learning.

3.3 Hebbian Learning

In this section we will extend the results of Section 3.2 by considering the problem of a network with Hebbian learning in which the learned weights are asymmetric and the whole weight matrix (not just the symmetric part) is used to calculated the node activations. In this section we will formulate such a system as a differential equation of the form

$$\dot{\boldsymbol{u}} = -\boldsymbol{P}(\boldsymbol{u})\nabla_{\boldsymbol{u}} V(\boldsymbol{u}) + \boldsymbol{Q}(\boldsymbol{u})\nabla_{\boldsymbol{u}} H(\boldsymbol{u}). \tag{3.8}$$

This is identical to equation (2.2) if the system is assumed to have inputs which are not time varying. In this case the inputs can easily be incorporated into the gradient-like component of the

dynamics. A general form for a network with Hebbian learning is shown in equation (3.3). Any matrix $\boldsymbol{C}$ can be decomposed into a sum of its symmetric and anti-symmetric parts $\boldsymbol{C} = \boldsymbol{C}^S + \boldsymbol{C}^A$, where the components of the symmetric part $\boldsymbol{C}^S$ are $c_{ij}^S = \frac{1}{2}(c_{ij} + c_{ji})$ and those of the anti-symmetric part $\boldsymbol{C}^A$ are $c_{ij}^A = \frac{1}{2}(c_{ij} - c_{ji})$. This is an excellent way to decompose the weight matrix in equation (3.3a) because the part of the activation which involves the symmetric part of the weight matrix can be written as a gradient-like system and the part that involves the anti-symmetric part can be written as a Hamiltonian system. If equation (3.3a) is decomposed as

$$\dot{x}_i = -a_i(x_i)\left[b_i(x_i) - \sum_{j=1}^{n} c_{ij}^S\, d_j(x_j)\right] + a_i(x_i)\left[\sum_{j=1}^{n} c_{ij}^A\, d_j(x_j)\right], \quad i = 1, \ \ldots\ , n, \tag{3.9}$$

then it is natural to select the potential function $V(\boldsymbol{u})$ as in equation (3.5) and $H(\boldsymbol{u})$ as

$$H(\boldsymbol{u}) = \sum_{k=1}^{n} \int_0^{x_k} \frac{d_k(\zeta_k)}{a_k(\zeta_k)}\, d\zeta_k. \tag{3.10}$$

The associated matrix $\boldsymbol{P}(\boldsymbol{u})$ remains as in equation (3.6), while $\boldsymbol{Q}(\boldsymbol{u})$ is defined as

$$\boldsymbol{Q}(\boldsymbol{u}) = \left(\begin{array}{c|c} \boldsymbol{E} & -\boldsymbol{F}^T \\ \hline \boldsymbol{F} & \boldsymbol{G} \end{array}\right). \tag{3.11}$$

Since the network contains n nodes and m variable weights, $\boldsymbol{E}$ is an $(n \times n)$ matrix, $\boldsymbol{F}$ is an $(n \times m)$ matrix, and $\boldsymbol{G}$ is an $(m \times m)$ matrix. The blocks in $\boldsymbol{Q}(\boldsymbol{u})$ are given by

$$\boldsymbol{E} = \begin{pmatrix} a_1(x_1)\, c_{11}^A\, a_1(x_1) & a_1(x_1)\, c_{12}^A\, a_2(x_2) & \cdots & a_1(x_1)\, c_{1n}^A\, a_n(x_n) \\ a_2(x_2)\, c_{21}^A\, a_1(x_1) & a_2(x_2)\, c_{22}^A\, a_2(x_2) & \cdots & a_2(x_2)\, c_{2n}^A\, a_n(x_n) \\ \vdots & \vdots & \ddots & \vdots \\ a_n(x_n)\, c_{n1}^A\, a_1(x_1) & a_n(x_n)\, c_{n2}^A\, a_2(x_2) & \cdots & a_n(x_n)\, c_{nn}^A\, a_n(x_n) \end{pmatrix}, \tag{3.12a}$$

$$\boldsymbol{F} = \boldsymbol{G} = \boldsymbol{O}. \tag{3.12b}$$

For this choice of the potential functions, the gradient $\nabla_{\boldsymbol{u}} V(\boldsymbol{u})$ is shown in equation (3.7), and $\nabla_{\boldsymbol{u}} H(\boldsymbol{u})$ is

$$\nabla_{\boldsymbol{u}} H(\boldsymbol{u}) = \left(\frac{d_1(x_1)}{a_1(x_1)} \ \ \frac{d_2(x_2)}{a_2(x_2)} \ \ \cdots \ \ \frac{d_n(x_n)}{a_n(x_n)} \ \ 0 \ \ 0 \ \ \cdots \ \ 0\right)^T. \tag{3.13}$$

3.4 Gated Learning

There is a weight update rule which is fundamentally different from Hebbian learning in that it is asymmetric and the decay term is not always active. This type of learning rule is called

gated learning in [1, 5]. This type of learning is used in instars, outstars, and in the various ART models. The node activation dynamics for this type of learning are identical to equation (3.3a), while the weight update dynamics in this scheme are

$$\dot{c}_{ij} = -\gamma_{ij}\, d_i(x_i)\, c_{ij} + \lambda_{ij}\, d_i(x_i)\, d_j(x_j). \tag{3.14}$$

Notice that under this learning rule a weight can not decay unless the node which the connection is incident to has a nonzero output. Also notice that the equilibrium value of a weight under this rule is the output value of the node that the weight is incident from. The constants γ_{ij} and λ_{ij} have the same meaning as in equation (3.3b). Networks that use the gated learning rule can be written in the form of equation (3.8) by choosing the potential functions $V(\boldsymbol{u})$ and $H(\boldsymbol{u})$ as in equations (3.5) and (3.10) respectively, defining $\boldsymbol{P}(\boldsymbol{u})$ as in equation (3.6), and selecting $\boldsymbol{Q}(\boldsymbol{u})$ as in equations (3.11) and (3.12) with the $\boldsymbol{F}$ block redefined as

$$\boldsymbol{F} = \begin{pmatrix} -a_1(x_1)\,\gamma_{11}\,c_{11} & 0 & \cdots & 0 \\ 0 & -a_2(x_2)\,\gamma_{21}\,c_{21} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -a_n(x_n)\,\gamma_{n1}\,c_{n1} \\ \vdots & \vdots & \ddots & \vdots \\ -a_1(x_1)\,\gamma_{1n}\,c_{1n} & 0 & \cdots & 0 \\ 0 & -a_2(x_2)\,\gamma_{2n}\,c_{2n} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -a_n(x_n)\,\gamma_{nn}\,c_{nn} \end{pmatrix}. \tag{3.15}$$

Note that a network employing the gated learning rule can *not* be formulated as a gradient-like system.

Chapter 4

Examples

In this section several examples of systems which can be put into the form of equation (2.2) are presented. In each case the necessary potential functions and associated matrices will be presented. The first example is a simple two node, fixed weight network whose activations oscillate given a constant input. The second example is a system designed using the gradient-Hamiltonian decomposition as a tool. The third example is a two node network whose weights are updated using the gated learning rule.

4.1 Example of an Oscillating Network

In [6] a two node fixed weight network is defined whose node activations oscillate given a constant input in a specified range. The network is shown in Figure 4.1. The weights in this

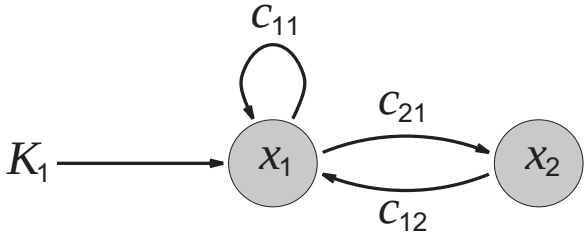

Figure 4.1: Configuration of the oscillating example network

network are constant and the node activation dynamics are given by

$$\begin{aligned} \dot{u}_1 &= -\mathcal{A}_1\, u_1 + (\mathcal{B}_1 - u_1)\,(K_1 + c_{11}\, d_1(u_1)) - u_1\,(c_{12}\, d_1(u_2))\,, \\ \dot{u}_2 &= -\mathcal{A}_2\, u_2 + c_{21}\, d_2(u_1), \end{aligned} \tag{4.1}$$

where the output functions $d_1(\cdot)$ and $d_2(\cdot)$ are defined as

$$d_1(x_i) = \begin{cases} 0 & \text{if } x_i \le \mathcal{T}_1, \\ \dfrac{80}{3}\,\dfrac{(x_i - \mathcal{T}_1)^4}{(\mathcal{T}_2 - \mathcal{T}_1)^2} & \text{if } \mathcal{T}_1 < x_i \le \dfrac{\mathcal{T}_2 + 3\,\mathcal{T}_1}{4}, \\ -\dfrac{80}{3}\,\dfrac{(x_i - \frac{\mathcal{T}_2+\mathcal{T}_1}{2})^4}{(\mathcal{T}_2 - \mathcal{T}_1)^2} + 20\,x_i^2 - 10(\mathcal{T}_2 + 3\,\mathcal{T}_1)\,x_i + & \\ \left[\dfrac{5}{24}(\mathcal{T}_2 - \mathcal{T}_1)^2 + \dfrac{5}{4}(\mathcal{T}_2 + 3\,\mathcal{T}_1)^2\right] & \text{if } \dfrac{\mathcal{T}_2 + 3\,\mathcal{T}_1}{4} < x_i \le \dfrac{3\,\mathcal{T}_2 + \mathcal{T}_1}{4}, \\ \dfrac{80}{3}\,\dfrac{(x_i - \mathcal{T}_2)^4}{(\mathcal{T}_2 - \mathcal{T}_1)^2} + x_i + \left[10(\mathcal{T}_2 - \mathcal{T}_1)^2 - \dfrac{3\,\mathcal{T}_2 + \mathcal{T}_1}{4}\right] & \text{if } \dfrac{3\,\mathcal{T}_2 + \mathcal{T}_1}{4} < x_i \le \mathcal{T}_2, \\ x_i + \left[10(\mathcal{T}_2 - \mathcal{T}_1)^2 - \dfrac{3\,\mathcal{T}_2 + \mathcal{T}_1}{4}\right] & \text{if } x_i > \mathcal{T}_2, \end{cases} \tag{4.2}$$

$$d_2(x_i) = x_i.$$

The function $d_1(x_i)$ defined in Equation (4.2) is a C^2 approximation to the function $\max(0, x - \frac{\mathcal{T}_2+\mathcal{T}_1}{2})$, as shown in Figure 4.2. The output functions $d_i(x_i)$ must be C^2 in order to guarantee that

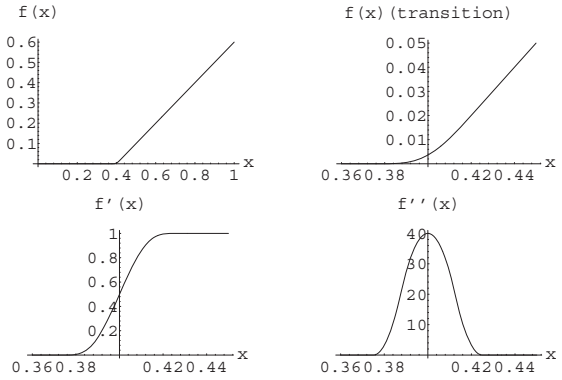

Figure 4.2: A C^2 function which approximates $\max[0, (x - 0.4)]$.

both of the potential functions $V(\boldsymbol{u})$ and $H(\boldsymbol{u})$ are also C^2. There are several valid choices for the potential functions $V(\boldsymbol{u})$ and $H(\boldsymbol{u})$ that put equation (4.1) into the form in equation (2.2). One possible choice for the potential functions is

$$V(\boldsymbol{u}) = \frac{1}{2}\,\mathcal{A}_2 u_2^2 - \int_0^{u_1} (\mathcal{B}_1 - \xi_1)\,(K_1 + c_{11}\, d_1(\xi_1))\; d\xi_1, \tag{4.3a}$$

$$H(\boldsymbol{u}) = c_{21}\, d_2(u_1) + \left[\mathcal{A}_1\, u_2 + \int_0^{u_2} c_{12}\, d_1(\xi_2)\; d\xi_2\right]. \tag{4.3b}$$

For this choice of the potential functions, the associated matrix $\boldsymbol{P}(\boldsymbol{u})$ is just the (2×2) identity matrix, and the matrix $\boldsymbol{Q}(\boldsymbol{u})$ is $\left(\begin{smallmatrix} 0 & -u_1 \\ u_1 & 0 \end{smallmatrix}\right)$.

For the simulations which follow, the constants in Equation (4.1) were chosen to be

$$\begin{aligned} &\mathcal{A}_1 = 1, \qquad \mathcal{A}_2 = 0.025, \qquad \mathcal{B}_1 = 1, \\ &c_{11} = 20, \qquad c_{11} = 33.3, \qquad c_{21} = 0.025, \\ &K_1 = 1. \end{aligned} \tag{4.4}$$

These are identical to the values used in [6]. The time evolution of the activations u_1 and u_2 for this oscillating network are shown in Figure 4.3. The behavior of this system will be explained

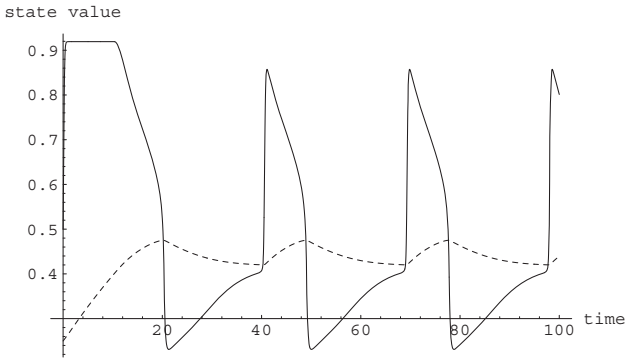

Figure 4.3: Time evolution of u_1 (solid line) and u_2 (dashed line) for the network shown in Figure 4.1

by examining the behavior of the gradient and Hamiltonian vector fields separately, and then considering the nature of the sum of these two vector fields. The Hamiltonian vector field described by Equation (4.3b) is shown in Figure 4.4. Notice that all trajectories converge to the line $u_1 = 0$,

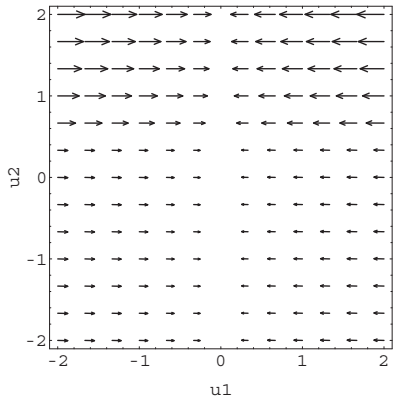

Figure 4.4: Hamiltonian vector field defined by Equation (4.3b)

which is a non-isolated equilibrium. Also note that trajectories started at initial conditions where $u_2 > 0.4$ will converge much more quickly than those where $u_2 \leq 0.4$. The gradient vector

field described by Equation (4.3a) is shown in Figure 4.5. Notice that all trajectories eventually

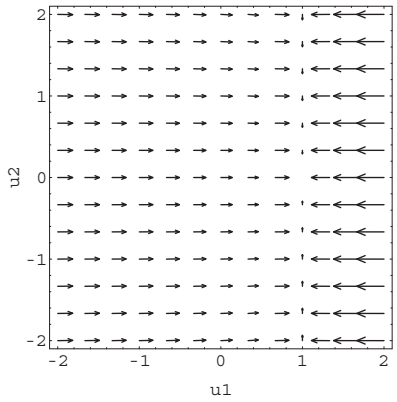

Figure 4.5: Gradient vector field defined by Equation (4.3a)

converge to the point $u_1 = 1, u_2 = 0$, however the convergence is much faster to the line $u_1 = 1$ than it is along this line. Inspection of Figures 4.4 and 4.5 makes it seem reasonable to expect that somewhere between $u_1 = 0$ and $u_1 = 1$ a region occurs where the gradient and Hamiltonian vectors cancel out. Furthermore it seems reasonable to expect this cancelation to occur somewhere in the region where $u_2 > 0$ since the two vector fields seem more equal in magnitude in this region. The actual behavior of this system is shown in Figure 4.6, which is a combination of the total

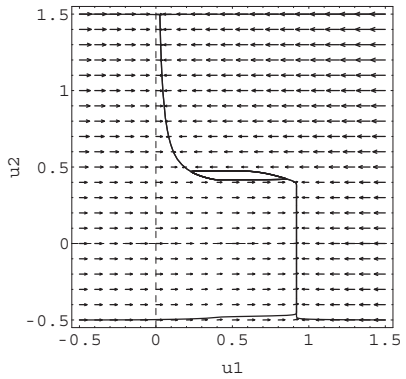

Figure 4.6: Vector field and phase plot of the system defined by Equation (4.1)

vector field and several phase space trajectories of the system defined by Equation (4.1). Notice that the oscillation occurs in a region where the gradient and Hamiltonian vector fields cancel in such a way as to allow a closed orbit.

It seems reasonable to state that a system which relies on this mechanism to oscillate would be extremely sensitive to the network parameters. Numerous simulations verify that this network oscillates only for a small range of parameter values. This example shows that the decomposition of dynamical systems into gradient and Hamiltonian components can be a useful tool in analyzing

the behavior of such systems.

4.2 Designing an Oscillating Network

The previous example suggests that the decomposition of a dynamical system into gradient and Hamiltonian portions can be used as a design tool. In this section, a system which oscillates will be designed by using the gradient portion of the system to define the basin of attraction and the Hamiltonian portion to define the orbit within that basin. The system will be globally stable in that all trajectories will converge to the same limit cycle. Also the qualitative behavior of the system will be unaffected by small parameter changes. Choose the gradient potential function $V(\boldsymbol{u})$ to be

$$\begin{aligned} V(\boldsymbol{u}) &= \left(\sqrt{u_1^2+u_2^2}+1\right)^2 \left(\sqrt{u_1^2+u_2^2}-1\right)^2, \\ &= \left(u_1^2+u_2^2\right)^2 - 2\left(u_1^2+u_2^2\right)+1. \end{aligned} \tag{4.5}$$

This potential, which is shown in Figure 4.7, has a circular non-isolated minimum of radius 1,

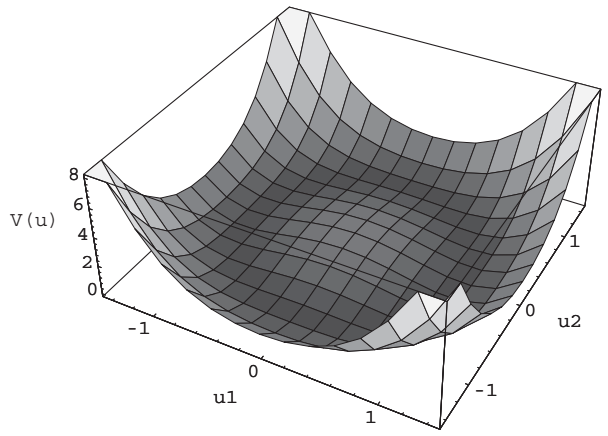

Figure 4.7: The gradient potential versus the state space of the potential defined in Equation (4.5)

centered at the origin. The surface rises at a quadratic rate away from this minimum, and the origin is a local maximum. The Hamiltonian potential $H(\boldsymbol{u})$ is chosen to be

$$H(\boldsymbol{u}) = -\left(u_1^2 + 30\,u_2^2\right). \tag{4.6}$$

The vector field defined by this equation, shown in Figure 4.8, follows the level surfaces of an

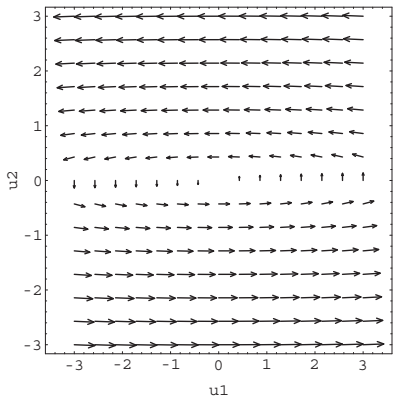

Figure 4.8: The Hamiltonian vector field defined by Equation (4.6)

ellipse centered at the origin. The equations for the overall system dynamics are

$$\begin{aligned} \dot{u}_1 &= -4\,u_1\left(u_1^2+u_2^2\right)+4\,u_1-60\,u_2, \\ \dot{u}_2 &= -4\,u_2\left(u_1^2+u_2^2\right)+4\,u_2+2\,u_1. \end{aligned} \tag{4.7}$$

These dynamic equations can be rewritten to look more "neural". In this form they are

$$\begin{pmatrix}\dot{u}_1\\ \dot{u}_2\end{pmatrix} = -4\begin{pmatrix}u_1\\ u_2\end{pmatrix} - (-4)\begin{pmatrix}u_1 & 0\\ 0 & u_2\end{pmatrix}\begin{pmatrix}1 & 1\\ 1 & 1\end{pmatrix}\begin{pmatrix}u_1^2\\ u_2^2\end{pmatrix} - 2\begin{pmatrix}0 & 30\\ -1 & 0\end{pmatrix}\begin{pmatrix}u_1\\ u_2\end{pmatrix} \tag{4.8}$$

This similar to the multiplicative equations defined by Grossberg. Note that each node has two channels with separate weights and output functions. The major difference is that Grossberg defines the one channel as having an excitatory effect on the system while the other has an inhibitory effect. In this example that is not the case. Other than that Equation (4.8) is simply Grossberg's multiplicative equations with certain parameters chosen as zero. In any case the phase plot for this system appears in Figure 4.9. The shape of the attractor can be changed by

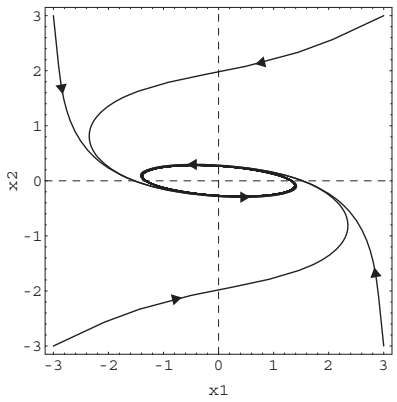

Figure 4.9: A phase plot of the system defined by Equation (4.7)

selecting a different Hamiltonian potential. For instance the Hamiltonian

$$H(\boldsymbol{u}) = -\left(u_1^4+15\,u_2^4\right), \tag{4.9}$$

leads to the system phase plot seen in Figure 4.10.

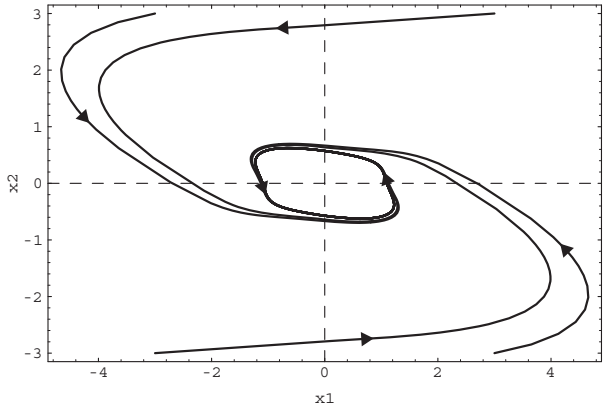

Figure 4.10: A phase plot of the system using the Hamiltonian potential defined in Equation (4.9)

4.3 An Example Using Gated Learning

To illustrate some properties of systems that use a gated learning rule the network in Figure 4.11 will be used. The network consists of two nodes, two weights, and an external input. If the node

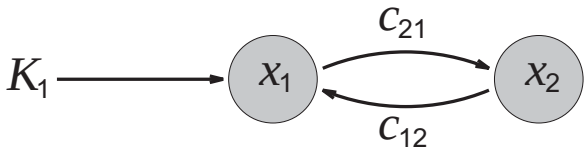

Figure 4.11: Configuration of the gated learning example network

activation dynamics of this network are additive, the weight update rule is gated, and the input is a constant, then the dynamic equations for the system in Figure 4.11 are

$$\begin{aligned}
\dot{x}_1(t) &= -\Upsilon_V\,\epsilon_1\,\left(\mathcal{A}_1\,x_1(t) - K_1 - 0.5\,\left(c_{12}(t) + c_{21}(t)\right)\,\tanh(\mathcal{S}_2\,x_2(t))\right) \\
&\quad + \Upsilon_H\,\epsilon_1\,\left(0.5\,\left(c_{12}(t) - c_{21}(t)\right)\,\tanh(\mathcal{S}_2\,x_2(t))\right) \\
\dot{x}_2(t) &= -\Upsilon_V\,\epsilon_2\,\left(\mathcal{A}_2\,x_2(t) - 0.5\,\left(c_{21}(t) + c_{12}(t)\right)\,\tanh(\mathcal{S}_1\,x_1(t))\right) \\
&\quad + \Upsilon_H\,\epsilon_2\,\left(0.5\,\left(c_{21}(t) - c_{12}(t)\right)\,\tanh(\mathcal{S}_1\,x_1(t))\right) \\
\dot{c}_{12}(t) &= \mathcal{G}_{12}\,\tanh(\mathcal{S}_1\,x_1(t))\,\tanh(\mathcal{S}_2\,x_2(t)) - \mathcal{L}_{12}\,\tanh(\mathcal{S}_1\,x_1(t))\,c_{12}(t) \\
\dot{c}_{21}(t) &= \mathcal{G}_{21}\,\tanh(\mathcal{S}_2\,x_2(t))\,\tanh(\mathcal{S}_1\,x_1(t)) - \mathcal{L}_{21}\,\tanh(\mathcal{S}_2\,x_2(t))\,c_{21}(t)
\end{aligned} \tag{4.10}$$

The dynamic equations are written in this unusual way to indicate which part of the dynamics is gradient-like and which is Hamiltonian. The first term in each equation is the gradient-like part

while the second term is the Hamiltonian part. Since the input is time independent, it can be incorporated into the gradient-like part of the system. In this case, the example system can be cast into the form of Equation (2.2) by defining the gradient potential $V(\boldsymbol{u})$ as

$$\begin{aligned} V(\boldsymbol{u}) = &-\frac{1}{2}\left[\tanh\left(\mathcal{S}_1\, x_1\right) c_{12} \tanh\left(\mathcal{S}_2\, x_2\right) + \tanh\left(\mathcal{S}_2\, x_2\right) c_{21} \tanh\left(\mathcal{S}_1\, x_1\right)\right] \\ &+ \int_0^{x_1} \mathcal{S}_1\ \left(\zeta_1 - K_1\right) \operatorname{sech}^2\left(\mathcal{S}_1\, \zeta_1\right)\, d\zeta_1 + \int_0^{x_2} \mathcal{S}_2\, \zeta_2\ \operatorname{sech}^2\left(\mathcal{S}_2\, \zeta_2\right)\, d\zeta_2, \end{aligned} \tag{4.11}$$

and the its associated matrix $\boldsymbol{P}(\boldsymbol{u})$ is given by

$$\boldsymbol{P}(\boldsymbol{u}) = \begin{pmatrix} \dfrac{\epsilon_1}{\mathcal{S}_1\ \operatorname{sech}^2\left(\mathcal{S}_1\, x_1\right)} & 0 & 0 & 0 \\ 0 & \dfrac{\epsilon_2}{\mathcal{S}_2\ \operatorname{sech}^2\left(\mathcal{S}_2\, x_2\right)} & 0 & 0 \\ 0 & 0 & 2\,\mathcal{G}_{12} & 0 \\ 0 & 0 & 0 & 2\,\mathcal{G}_{21} \end{pmatrix}, \tag{4.12}$$

while the Hamiltonian $H(\boldsymbol{u})$ is

$$H(\boldsymbol{u}) = \int_0^{x_1} \frac{\tanh\left(\mathcal{S}_1\, \zeta_1\right)}{\epsilon_1}\, d\zeta_1 + \int_0^{x_2} \frac{\tanh\left(\mathcal{S}_2\, \zeta_2\right)}{\epsilon_2}\, d\zeta_2, \tag{4.13}$$

and its associated matrix $\boldsymbol{Q}(\boldsymbol{u})$ is given by

$$\boldsymbol{Q}(\boldsymbol{u}) = \begin{pmatrix} 0 & \epsilon_1 \dfrac{1}{2}\left(\, c_{12} - c_{21}\,\right) \epsilon_2 & \epsilon_1\, \mathcal{L}_{12}\, c_{12} & 0 \\ \epsilon_2 \dfrac{1}{2}\left(\, c_{21} - c_{12}\,\right) \epsilon_1 & 0 & 0 & \epsilon_2\, \mathcal{L}_{21}\, c_{21} \\ -\epsilon_1\, \mathcal{L}_{12}\, c_{12} & 0 & 0 & 0 \\ 0 & -\epsilon_2\, \mathcal{L}_{21}\, c_{21} & 0 & 0 \end{pmatrix}, \tag{4.14}$$

where the state vector $\boldsymbol{u}$ is

$$\boldsymbol{u} = \left[x_1,\ x_2,\ c_{12},\ c_{21}\right]^T. \tag{4.15}$$

It is deceptively difficult to analyze the behavior of this apparently simple system.

Chapter 5

Conclusion

In this work we have proposed that the dynamics of many neural networks can be written as the sum of a gradient-like component, a Hamiltonian component, and some input component. We then proposed a method that allows this decomposition to be used to gain some insight into the behavior such systems and also gauge the effect of the network parameters on that behavior. We showed that this decomposition extends our work in [10] by allowing the node activations to be computed using both symmetric and anti-symmetric parts of the weight matrix, and also by allowing additional learning rules to be formulated. We then put forward some examples of networks which were amenable to this formulation.

Bibliography

[1] G.A. Carpenter. Neural network models for pattern recognition and associative memory. *Neural Networks*, 2(4):243–257, 1989.

[2] J.T. Duarte and R.V. Mendes. Deformation of Hamiltonian dynamics and constants of motion in dissipative systems. *Journal of Mathematical Physics*, 24(7):1772–1778, 1983.

[3] P. Földiák. Forming sparse representaions by local anti–Hebbian learning. *Biological Cybernetics*, 64(2):165–170, 1990.

[4] C.L. Giles and T. Maxwell. Learning, invariance, and generalization in high–order neural networks. *Applied Optics*, 26(23):4972–4978, 1987.

[5] S. Grossberg. Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural Networks*, 1(1):17–61, 1988.

[6] S. Grossberg and D. Somers. Synchronized oscillations during cooperative feature linking in a cortical model of visual perception. *Neural Networks*, 4(4):453–466, 1991.

[7] H. Gutfreund and M. Mezard. Processing of temporal sequences in neural networks. *Physical Review Letters*, 61(2):235–238, 1988.

[8] J.A. Hertz, G. Grinstein, and S.A. Solla. Irreversible spin glasses and neural networks. In J.L. van Hemmen and I. Morgenstern, editors, *Heidelberg colloquium on glassy dynamics*, pages 538–546. Springer-Verlag, Inc., Berlin, Germany, 1987. Volume 275 of *Lecture Notes in Physics*.

[9] J.W. Howse, C.T. Abdallah, and G.L. Heileman. Gradient-like dynamics in neural networks. Technical Report EECE93–001, University of New Mexico, January 1993.

[10] J.W. Howse, C.T. Abdallah, G.L. Heileman, and M. Georgiopoulos. A gradient-like formulation of neural network dynamics. Submitted to *Neural Networks*.

[11] B. Kosko. *Neural networks and fuzzy systems: A dynamical systems approach to machine intelligence*. Prentice Hall, Inc., Englewood Cliffs, NJ, 1992.

[12] R.V. Mendes and J.T. Duarte. Decomposition of vector fields and mixed dynamics. *Journal of Mathematical Physics*, 22(7):1420–1422, 1981.

[13] R.V. Mendes and J.T. Duarte. Vector fields and neural networks. *Complex Systems*, 6(1):21–30, 1992.

[14] G. Parisi. Asymmetric neural networks and the process of learning. *Journal of Physics A: Mathematical and General*, 19(11):L675–L680, 1986.

[15] D. Psaltis, C.H. Park, and J. Hong. Higher order associative memories and their optical implementations. *Neural Networks*, 1(2):149–163, 1988.