ANN ee543

CHAPTER II

Recurrent Neural Networks

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

Introduction

- In this chapter first the dynamics of the continuous space recurrent neural networks will be examined in a general framework.
- Then, the Hopfield Network as a special case of this kind of networks will be introduced.

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.1. Dynamical Systems

- The dynamics of a large class of neural network models, may be represented by a set of first order differential equations in the form

$$\frac{d}{dt}x_j(t) = F_j(\mathbf{x}\ (t)) \quad j = 1..N \qquad (2.1.1)$$

where F_j is a nonlinear function of its argument.

- In a more compact form it may be reformulated as

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{F}(\mathbf{x}(t)) \qquad (2.1.2)$$

where the nonlinear function $\mathbf{F}$ operates on elements of the state vector $\mathbf{x}(t)$ in an autonomous way, that is $\mathbf{F}(\mathbf{x}(t))$ does not depend explicitly on time t.

- $\mathbf{F}(\mathbf{x})$ is a vector field in an N-dimensional state space. Such an equation is called state space equation and $\mathbf{x}(t)$ is called the state of the system at particular time t.

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.1. Dynamical Systems Existence and Uniqueness

For the state space equation (2.1.2) to have a solution and for the solution to be unique, we have to impose certain restrictions on the vector function $\mathbf{F}(\mathbf{x}(t))$.

- For a solution to exist, it is sufficient that $\mathbf{F}(\mathbf{x})$ to be continuous in all of its arguments.
- For uniqueness of the solution, $\mathbf{F}(\mathbf{x})$ should satisfy Lipschitz condition.

Let $||\mathbf{x}||$ denote a norm, which may be the Euclidean length, Hamming distance or any other one, depending on the purpose. Let $\mathbf{x}$ and $\mathbf{y}$ be a pair of vectors in an open set **S,** in vector space. Then according to the Lipschitz condition, there exists a constant κ such that

$$|| \mathrm{F(x) - F(y)} || \le \kappa \, || \mathrm{x - y} || \qquad (2.1.3)$$

for all $\mathbf{x}$ and $\mathbf{y}$ in **S**. A vector $\mathbf{F}(\mathbf{x})$ that satisfies equation (2.1.3) is said to be ***Lipschitz***. In particular, if all partial derivatives $\partial F_i(\mathbf{x})/\partial \mathrm{x}_j$ are finite everywhere, then the function $\mathbf{F}(\mathbf{x})$ satisfies the Lipschitz condition [Haykin 94].

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.1. Dynamical Systems Phase Space

The phase space of a dynamical system describes the global characteristics of the motion rather than the detailed aspects of analytic or numeric solutions of the equation.

- At a particular instant of time *t*, a single point in the n-dimensional phase space represents the observed state of the state vector, that is $\mathbf{x}(t)$.
- Changes in the state of the system with time t are represented as a curve in the phase space, each point on the curve carrying (explicitly or implicitly) a label that records the time of observation.
- This curve is called a trajectory or orbit of the system. Figure 2.1.a. illustrates a trajectory in a two dimensional system.

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.1. Dynamical Systems Phase Space

The family of trajectories each for a different initial condition $\mathbf{x}(0)$ is called the phase portrait of the system (Figure 2.1.b).

- The phase portrait includes all those points in the phase space where the field vector $\mathbf{F}(\mathbf{x})$ is defined.
- For an autonomous system, there will be one and only one trajectory passing through an initial state..
- The tangent vector, that is $d\mathbf{x}(t)/dt$, represents the instantaneous velocity $\mathbf{F}(\mathbf{x}(t))$ of the trajectory.

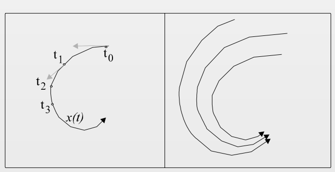


Figure 2.1. a) A two dimensional trajectory b) Phase portrait

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.3. Major forms of Dynamical Systems

We distinguish three major forms dynamical system, for fixed weights and inputs (Figure 2.2):

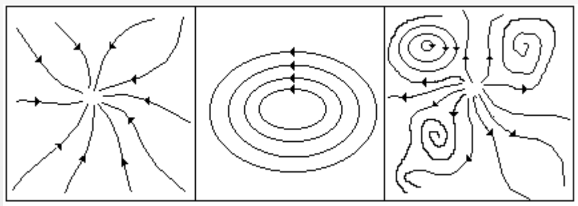


Figure 2.2. Three major forms of dynamical systems
a) Convergent b) Oscillatory c) Chaotic

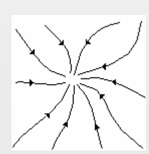
ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.3. Major forms of Dynamical Systems

a) ***Convergent:*** every trajectory $\mathbf{x}(t)$ converges to some fixed point, which is a state that does not change over time (Figure 2.2.a).

- These fixed points are called the attractors of the system.
- The set of initial states $\mathbf{x}(0)$ that evolves to a particular attractor is called the basin of attraction.
- The locations of the attractors and the basin boundaries change as the dynamical system parameters change. For example, by altering the external inputs or connection weights in a recurrent neural network the basin attraction of the system can be adjusted.

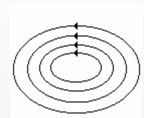


ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.3. Major forms of Dynamical Systems

b) ***Oscillatory:*** every trajectory converges either to a cycle or to a fixed point. A cycle of period T satisfies $\mathbf{x}(t+T)=\mathbf{x}(t)$ for all times t (Figure 2.2.b)



b) ***Chaotic:*** most trajectories do not tend to cycles or fixed points. One of the characteristics of chaotic systems is that the long-term behavior of trajectories is extremely sensitive to initial conditions. That is, a slight change in the initial state $\mathbf{x}(0)$ can lead to very different behaviors, as t becomes large. (Figure 2.2.c)

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2. 4. Gradient, Conservative and Dissipative Systems Gradient

- For a vector field $\mathbf{F}(\mathbf{x})$ on state space $\mathbf{x}(t) \in R^N$, the ∇ operator helps in formal description of the system. In fact, ∇ is an operational vector defined as:

$$\nabla=[\partial/\partial x_1 \quad \partial/\partial x_2 \quad \partial/\partial x_N]. \qquad (2.4.1)$$

- If the ∇ operator applied on a scalar function E of vector $\mathbf{x}(t)$, that is

$$\nabla E=[\partial E/\partial x_1 \quad \partial E/\partial x_2 \quad \ldots \quad \partial E/\partial x_N]. \qquad (2.4.2)$$

is called the gradient of the function E and extends in the direction of the greatest rate of change of E and has that rate of change for its length.

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2. 4. Gradient, Conservative and Dissipative Systems *Level surfaces*

- If we set $E(\mathbf{x})=c$, we obtain a family of surfaces known as ***level surfaces*** of E, as $\mathbf{x}$ takes on different values. On the assumption that E is single valued at each point, one and only one level surface passes through any given point P. The gradient of $E(\mathbf{x})$ at any point P is perpendicular to the level surface of E, which passes through that point. (Figure 2.3)

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2. 4. Gradient, Conservative and Dissipative Systems

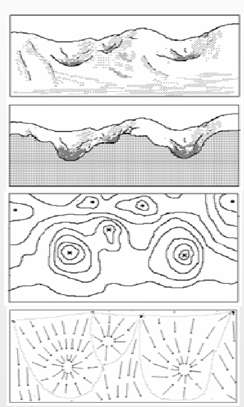


Figure 2.3 a) Energy landscape b) a slice c) level surfaces d) - gradient

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2. 4. Gradient, Conservative and Dissipative Systems *Divergence*

For a vector field

$$\mathbf{F}(\mathbf{x})=[F_1(\mathbf{x}) \quad F_2(\mathbf{x}) \quad \ldots \quad F_N(\mathbf{x})]^T \qquad (2.4.3)$$

the inner product

$$\nabla.\mathbf{F}= \frac{\partial F_1}{\partial x_1}+\frac{\partial F_2}{\partial x_2}+..+\frac{\partial F_N}{\partial x_N}. \qquad (2.4.4)$$

is called the *divergence* of $\mathbf{F}$, and it has a scalar value.

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2. 4. Gradient, Conservative and Dissipative Systems *Dissipative ve and conservative systems*

- Consider a region of volume V and surface S in the phase space of an autonomous system, and assume a flow of points from this region.
- Let $\mathbf{n}$ denote a unit vector normal to the surface at dS pointing outward from the enclosed volume.
- Then, according to the divergence theorem, the relation

$$\int_s (\mathbf{F}(\mathbf{x}).\mathbf{n})dS = \int_v (\nabla.\mathbf{F}(\mathbf{x}))dV \qquad (2.4.5)$$

holds between the volume integral of the divergence of $\mathbf{F}(\mathbf{x})$ and the surface integral of the outwardly directed normal component of $\mathbf{F}(\mathbf{x})$.

ANN ee543

CHAPTER II : ***Recurrent Neural Networks***

2. 4. Gradient, Conservative and Dissipative Systems
Dissipative ve and conservative systems

$$\int_s (\mathbf{F}(\mathbf{x}).\mathbf{n})dS = \int_v (\nabla.\mathbf{F}(\mathbf{x}))dV \qquad (2.4.5)$$

- The quantity on the left-hand side of Eq. (2.4.5) is recognized as the net flux flowing out of the region surrounded by the closed surface S. If the quantity is zero (or equivalently $\nabla \cdot \mathbf{F}(\mathbf{x}) = 0$ in V), the system is ***conservative***; if it is negative (or $\nabla \cdot \mathbf{F}(\mathbf{x}) < 0$ in V), the system is ***dissipative***.
- If the system is dissipative, this guarantees the stability of the system.

ANN ee543

CHAPTER II : ***Recurrent Neural Networks***

2.5. Equilibrium States

Remember

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{F}(\mathbf{x}(t)) \qquad (2.1.2)$$

•A constant vector **x*** satisfying the condition

$$\mathbf{F}(\mathbf{x}^*) = \mathbf{0}, \qquad (2.5.1)$$

is called an ***equilibrium state (stationary state or fixed point)*** of the dynamical system defined by Eq. (2.1.2).

• Since it results in

$$\left.\frac{dx_i}{dt}\right|_{x^*} = 0 \quad for \quad i = 1..N, \qquad (2.5.2)$$

the constant function $\mathbf{x}(t)=\mathbf{x}^*$ is a solution of the dynamical system.

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.5. Equilibrium States

Remember

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{F}(\mathbf{x}(t)) \qquad (2.1.2)$$

$$\mathbf{F}(\mathbf{x}^*) = \mathbf{0}\,, \qquad (2.5.1)$$

- If the system is operating at an equilibrium point, then the state vector stays constant, and the trajectory with an initial state $\mathbf{x}(0)=\mathbf{x}^*$ degenerates to a single point.
- We are frequently interested in the behavior of the system around the equilibrium points, and try to investigate if the trajectories around the equilibrium points are converging to the equilibrium point, diverging from it or staying in an orbit around the point or combination of these.
- The use of a linear approximation of the nonlinear function $\mathbf{F}(\mathbf{x})$ makes it easier to understand the behavior of the system around the equilibrium points.

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.5. Equilibrium States

Remember

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{F}(\mathbf{x}(t)) \qquad (2.1.2)$$

- Let $\mathbf{x}=\mathbf{x}^*+\Delta\mathbf{x}$ be a point around $\mathbf{x}^*$. If the nonlinear function $\mathbf{F}(\mathbf{x})$ is smooth and if the disturbance $\Delta\mathbf{x}$ is small enough then it can be approximated by the first two terms of its Taylor expansion around $\mathbf{x}^*$ as:

$$\mathbf{F}(\mathbf{x}^*+\Delta\mathbf{x}) \cong \mathbf{F}(\mathbf{x}^*) + \mathbf{F}'(\mathbf{x}^*)\Delta\mathbf{x} \qquad (2.5.3)$$

where

$$\mathbf{F}'(\mathbf{x}^*) = \frac{\partial}{\partial\,\mathbf{x}}\mathbf{F}\Big|_{\mathbf{x}=\mathbf{x}^*} \qquad (2.5.4)$$

that is, in particular:

$$F'_{ij}(\mathbf{x}^*) = \frac{\partial F_j(\mathbf{x})}{\partial x_i}\Big|_{\mathbf{x}=\mathbf{x}^*}\,. \qquad (2.5.5)$$

- Notice that $\mathbf{F}(\mathbf{x}^*)$ and $\mathbf{F}'(\mathbf{x}^*)$ in Eq. (2.5.3) are constant, therefore it is a linear equation in terms of $\Delta\,\mathbf{x}$.

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.5. Equilibrium States

Remember

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{F}(\mathbf{x}(t)) \quad (2.1.2)$$

$$\mathbf{F}(\mathbf{x}^*) = \mathbf{0}\,, \quad (2.5.1)$$

$$\mathbf{F}(\mathbf{x}^* + \Delta\mathbf{x}) \cong \mathbf{F}(\mathbf{x}^*) + \mathbf{F}'(\mathbf{x}^*)\Delta\mathbf{x} \quad (2.5.3)$$

•Since an equilibrium point satisfies Eq. (2.5.1), we obtain

$$\mathbf{F}(\mathbf{x}^* + \Delta\mathbf{x}) \cong \mathbf{F}'(\mathbf{x}^*)\Delta\mathbf{x} \quad (2.5.6)$$

On the other hand, since

$$\frac{d}{dt}(\mathbf{x}^* + \Delta\mathbf{x}) = \frac{d}{dt}\Delta\mathbf{x} \quad (2.5.7)$$

the Eq. (2.1.2) becomes

$$\frac{d}{dt}\Delta\mathbf{x} = \mathbf{F}'(\mathbf{x}^*)\Delta\mathbf{x} \quad (2.5.8)$$

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.5. Equilibrium States

Remember

$$\frac{d}{dt}\Delta\mathbf{x} = \mathbf{F}'(\mathbf{x}^*)\Delta\mathbf{x} \quad (2.5.8)$$

•Since Eq. (2.5.8) defines a homogenous differential equation with constant real coefficient, the eigenvalues of the matrix $\mathbf{F}'(\mathbf{x}^*)$ determines the behavior of the system.

• In order to have $\Delta\mathbf{x}(t)$ to diminish as $t\rightarrow\infty$, we need the real parts of all the eigenvalues to be negative.

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.6. Stability

- An equilibrium state $\mathbf{x}^*$ of an autonomous nonlinear dynamical system is called ***stable***, if for any given positive ε, there exists a positive δ satisfying,

$$||\mathbf{x}(0)-\mathbf{x}^*|| < \delta \Rightarrow ||\mathbf{x}(t)-\mathbf{x}^*|| < \varepsilon \textit{ for all } t>0. \qquad (2.6.1)$$

- If $\mathbf{x}^*$ is a stable equilibrium point, it means that any trajectory described by the state vector $\mathbf{x}(t)$ of the system can be made to stay within a small neighborhood of the equilibrium state $\mathbf{x}^*$ by choosing an initial state $\mathbf{x}(0)$ close enough to $\mathbf{x}^*$.
- An equilibrium point $\mathbf{x}^*$ is said to be ***asymptotically stable*** if it is also convergent, where convergence requires the existence of a positive such δ that

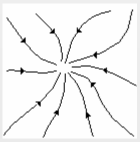
$$||\mathbf{x}(0)-\mathbf{x}^*|| < \delta \Rightarrow \lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}^*. \qquad (2.6.2)$$

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.6. Stability

- If the equilibrium point is convergent, the trajectory can be made approaching to $\mathbf{x}^*$ as t goes to infinity, by choosing again an initial state $\mathbf{x}(0)$ close enough to $\mathbf{x}^*$.
- Notice that asymptotically stable states correspond to attractors of the system.
- For an autonomous nonlinear dynamic system the asymptotic stability of an equilibrium $\mathbf{x}^*$ can be decided by the existence of energy functions, which are also called Liapunov functions.



ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.6. Stability Liapunov Function

- A continuous function $L(\mathbf{x})$ with a continuous time derivative $L'(\mathbf{x})=dL(\mathbf{x})/dt$ is a definite Liapunov function if it satisfies:

  a) $L(\mathbf{x})$ is bounded

  b) $L'(\mathbf{x})$ is negative definite, that is:

$$L'(x) < 0 \text{ for } x \neq x^* \qquad (2.6.3)$$

and

$$L'(x) = 0 \text{ for } x = x^* \qquad (2.6.4)$$

- If the condition (2.6.3) is in the form

$$f(\mathbf{x}) \leq 0 \text{ for } \mathbf{x} \neq \mathbf{x}^* \qquad (2.6.5)$$

  the Liapunov function is called semidefinite.

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.6. Stability Liapunov's Theorem

- The stability of an equilibrium point can be decided by using the following theorem:

  The equilibrium state $\mathbf{x}^*$ is stable (asymptotically stable), if there exists a semidefinite (definite) Liapunov function in a small neighborhood of $\mathbf{x}^*$.

- The use of Liapunov functions makes it possible to decide the stability of equilibrium points without solving the state-space equation of the system.
- Unfortunately there is not a formal way to find a Liapunov function, mostly it is determined in a trial and error fashion. If we are able to find a Liapunov function, then we state the stability of the system. However, the inability to find a Liapunov function, does not imply the instability of the system.
- Often convergence of neural networks is guaranteed by an introduction of an energy function together with the network itself.

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.6. Stability Liapunov's Theorem

- In fact the energy functions are Liapunov functions, so non-increasing along trajectories. Therefore the dynamics of the network can be visualized in terms of some multidimensional 'energy landscapes' as given previously in Figure 2.3.
- The attractors of the dynamic system are the local minima of the energy function surrounded with 'valleys' corresponding to the basins of attraction (Figure 2.4).

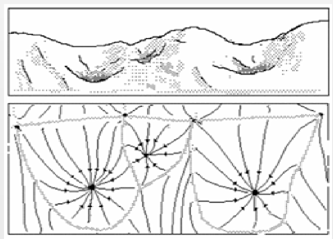


Figure 2.4. Energy landscape and basin attractions

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.7. Effect of input and initial state on the attraction

- The convergence of a network to an attractor of the activation dynamics may be viewed as a retrieval process in which the fixed point is interpreted as the output of the neural network.
- As an example consider the following network dynamic:

$$\frac{d}{dt}x_i(t) = -x_i(t) + f_i\left(\sum_j w_{ji}x_j + \theta_i\right) \qquad (2.7.1)$$

Assume that the weight matrix **W** is fixed and the network is specified through θ and initial state x(0). Both θ and **x**(0) are ways of introducing an input pattern into the network, although they play distinct dynamical roles

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.7. Effect of input and initial state on the attraction

Remember

$$\frac{d}{dt}x_i(t) = -x_i(t) + f_i(\sum_j w_{ji}x_j + \theta_i) \qquad (2.7.1)$$

- We then distinguish two modes of operation, depending on whether
  - network has fixed $\mathbf{x}(0)$ and input is applied as $\theta=\mathbf{u}$
  - network has fixed θ and $\mathbf{x}(0)=\mathbf{u}$ is chosen.

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.7. Effect of input and initial state on the attraction

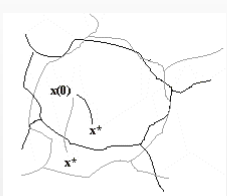


Figure 2.5. a) The same initial value $\mathbf{x}(0)$ may result in different fixed points as final value for different $\mathbf{u}$

case 1: network has fixed $\mathbf{x}(0)$ and input is applied as $\theta=\mathbf{u}$

The vector $\mathbf{u}$ acts as input and the initial is state set to some constant vector for all inputs. In general, the value of the attractors vary smoothly as the vector $\mathbf{u}$ is varied, hence the network provides a continuous mapping between the input and the output spaces (Figure 2.5.a).

ANN ee543

CHAPTER II : ***Recurrent Neural Networks***

2.7. Effect of input and initial state on the attraction

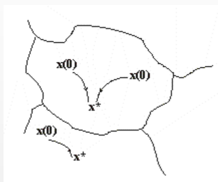


Figure 2.5. b) Different $\mathbf{x}(0)$ may converge to different fixed values although $\mathbf{u}$ is the same

Case 2: network has fixed θ and $\mathbf{x}(0)=\mathbf{u}$ is chosen.

In this case, the input pattern is presented to the network through the initial state $\mathbf{x}(0)$ while having fixed θ. The attractors of the dynamics may be used to represent items in a memory while the initial states are the stimulus to remember the stored memory items. The initial states that contain incomplete or erronous information may be considered as queries to the memory. The network then converges to the complete memory items that best fits the stimulus. (Figure 2.5.b)

ANN ee543

CHAPTER II : ***Recurrent Neural Networks***

2.8 Cohen-Grossberg Theorem

Cohen-Grossberg theorem is useful in deciding the stability of a certain class of neural networks.

Theorem: Given a neural network with *N* processing elements having bounded output signals $f_i(a_i)$ and transfer functions of the form

$$\frac{d}{dt}a_i = \alpha_i(a_i)\left(\beta_i(a_i) - \sum_{j=1}^{n} w_{ji} f_j(a_j)\right) \quad i = 1..N \qquad (2.8.1)$$

satisfying constraints:

a) Symmetry:

$$w_{ji} = w_{ij} \quad i, j = 1..N \qquad (2.8.2)$$

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.8 Cohen-Grossberg Theorem

b) Nonnegativity:

$$\alpha_i(a) \geq 0 \quad i = 1..N \tag{2.8.3}$$

c) Monotonocity:

$$f'(a) = \frac{d}{da}(f(a)) \geq 0 \quad for \quad a \geq 0 \tag{2.8.4}$$

Then the network will converge to some stable point and there will be at most a countable number of such stable points.

The function

$$E = +\tfrac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ji} f_i(a_i) f_j(a_j) - \sum_{j=1}^{N}\int_0^{a_j} \beta_j(s) f'(s) ds \tag{2.8.5}$$

is an energy function of the system. That is, E has negative time derivative on every possible trajectory that the network's state can follow.

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.8 Cohen-Grossberg Theorem

Proof:

Due to condition (1) W is symmetric. The time derivative of the energy function can be written as

$$\frac{dE}{dt} = -\sum_{i=1}^{N} \alpha_i(a_i) f_i'(a_i)(\beta_i(a_i) - \sum_{j=1}^{N} w_{ji} f_j(a_j))^2 \tag{2.8.6}$$

and it has negative value for $\mathbf{a} \neq \mathbf{a}^*$ whenever conditions (2) and (3) are satisfied.

Since

$$\frac{dE}{dt} < 0 \quad for \quad a_i \neq a_i^* \tag{2.8.7}$$

the global system is therefore asymptotically stable.

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.9 Hopfield Network

- The ***continuous deterministic Hopfield model*** which is based on continuous variables and responses, is proposed in [Hopfield 84] to extend their discrete-memory model [Hopfield 82] for the processing elements to resemble actual neurons more closely.
- In this model the neurons are modeled as amplifiers in conjunction with feedback circuits made up of wires, resistors and capacitors which suggests the possibility of building these circuits using VLSI technology (Figure 2.6).

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.9 Hopfield Network

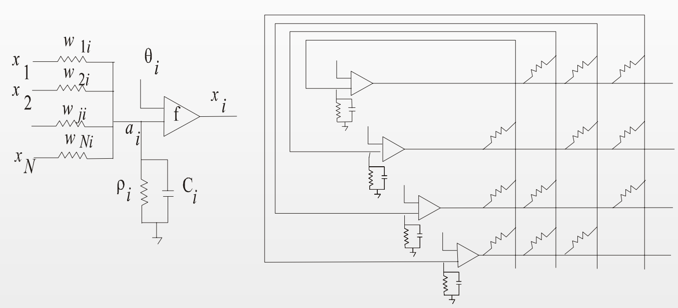


Figure 2.6 Hopfield Network made of electronical components

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.9 Hopfield Network

The output of the amplifier, *xi*, is a continuous, monotonically increasing function of the instantaneous input *ai* to the *i*th amplifier. The input-output relation of the i^{th} amplifier is given by

$$f_i(a) = \tanh(\kappa_i a) \qquad (2.9.1)$$

where κ_i is a constant called the ***gain parameter***.

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.9 Hopfield Network

Notice that since

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \qquad (2.9.2)$$

the amplifier transfer function is in fact a sigmoid function

$$f_i(a) = \frac{1 - e^{-\kappa_i a}}{1 + e^{-\kappa_i a}} = \frac{2}{1 + e^{-2a\kappa_i}} - 1 \qquad (2.9.3)$$

as given in equation (1.2.8) with $\kappa' = 2\kappa$, but shifted so that to have values between -1 and +1.

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.9 Hopfield Network

In Figure 2.7, the transfer function is illustrated for several values of κ. This function is differentiable at each point and always has positive derivative. In particular, its derivative at origin gives the gain *i*, that is

$$\kappa_i = \frac{df_i}{da}\Big|_{a=0} \qquad (2.9.4)$$

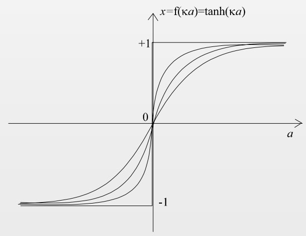


Figure 2.7 Output function used in Hopfield network

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.9 Hopfield Network

The amplifiers in the Hopfield circuit correspond to the neurons. A set of nonlinear differential equations describes the dynamics of the network.

The input voltage a_i of the amplifier *i* and it is determined by the equation

$$C_i \frac{da_i(t)}{dt} = -\frac{1}{R_i} a_i(t) + \sum_j w_{ji} f_j(a_j(t)) + \theta_i \qquad (2.9.5)$$

while

$$x_i = f_j(a_i) \qquad (2.9.6)$$

corresponds to the output voltage. In Eq. (2.9.5) R_i is determined as

$1/R_i = \rho_i + \Sigma_j w_{ji}$

ANN ee543

CHAPTER II : ***Recurrent Neural Networks***

2.9 Hopfield Network

Remember

$$C_i \frac{da_i(t)}{dt} = -\frac{1}{R_i} a_i(t) + \sum_j w_{ji} f_j(a_j(t)) + \theta_i \qquad (2.9.5)$$

The state of the network is described by an *N* dimensional state vector where *N* is the number of neurons in the network. The i^{th} component of the state vector is given by the output value of the i^{th} amplifier taking real values between -1 and 1. The state of the network moves in the state space in a direction determined by the above nonlinear dynamic equation (2.9.5).

ANN ee543

CHAPTER II : ***Recurrent Neural Networks***

2.9 Hopfield Network

Remember

$$C_i \frac{da_i(t)}{dt} = -\frac{1}{R_i} a_i(t) + \sum_j w_{ji} f_j(a_j(t)) + \theta_i \qquad (2.9.5)$$

Given the neuron characteristics by (2.9.5), Hopfield network can be represented by a neural network as shown in Figure 2.8.

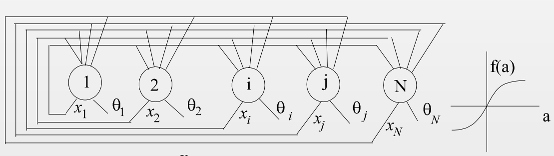


Figure 2.8 Hopfield Network made of neurons

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.9 Hopfield Network

The energy function for the continuous Hopfield model is given by the formula

$$E = -\tfrac{1}{2}\sum_i \sum_j w_{ji} x_j x_i + \sum_i \tfrac{1}{R_i}\int_0^{x_i} f_i^{-1}(x)dx - \sum_i \theta_i x_i \qquad (2.9.7)$$

where f_i^{-1} is the inverse of the function *fi*, that is

$$f_i^{-1}(x_i) = a_i \qquad (2.9.8)$$

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.9 Hopfield Network

Remember

$$f_i(a) = \frac{1-e^{-\kappa_i a}}{1+e^{-\kappa_i a}} = \frac{2}{1+e^{-2a\kappa_i}} - 1 \qquad (2.9.3)$$

In particular, for the transfer function defined by the equation (2.9.3), we have

$$f_i^{-1}(x) = -\ln\frac{1-x}{1+x} \qquad (2.9.9)$$

which is shown in Figure 2.9.

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.9 Hopfield Network

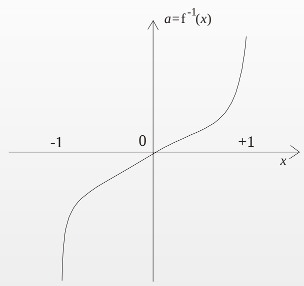


Figure 2.9 Inverse of the output function

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.9 Hopfield Network : Stability

One way to show the stabilty of Hopfield network is to show that its energy function is a Liapunov function.

For energy E of the Hopfield network to be a Lyapunov function, it should satisfy the following constraints:

a) $E(\mathbf{x})$ is bounded

b) $\frac{dE}{dt} \leq 0$

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.9 Hopfield Network: *stability*

Remember:

$$E = -\tfrac{1}{2}\sum_i \sum_j w_{ji} x_j x_i + \sum_i \tfrac{1}{R_i}\int_0^{x_i} f_i^{-1}(x)dx - \sum_i \theta_i x_i \qquad (2.9.8)$$

Because the function tanh(a) is used in the system as the output function, it limits the state variable to take value between $-1<x_i<1$. Furthermore, because the integral of the inverse of this function is bounded if $-1<x_i<1$, the energy function given by Eq. (2.9.8) is bounded.

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.9 Hopfield Network: *stability*

It can be easily shown that the derivative of the energy function is equivalent to:

$$\frac{dE}{dt} = -\sum_i C_i \frac{df_i^{-1}(x)}{dx}\left(\frac{dx_i}{dt}\right)^2 \qquad (2.9.15)$$

Due to equation (2.9.9) we have

$$\frac{df_i^{-1}(x)}{dx} \geq 0 \qquad (2.9.16)$$

for any value of x. So Eq. (2.9.15) implies that,

$$\frac{dE}{dt} \leq 0 \qquad (2.9.17)$$

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.9 Hopfield Network: *stability*

Remember

$$E=-\tfrac{1}{2}\sum_i \sum_j w_{ji}x_j x_i+\sum_i \tfrac{1}{R_i}\int_0^{x_i} f_i^{-1}(x)dx-\sum_i \theta_i x_i \qquad (2.9.8)$$

Therefore the energy function described by equation (2.9.8) is a Lyapunov function for the Hopfield network when the connection weights are symmetrical.

This means that, whatever the initial state of the network is, it will converge to one of the equilibrium states depending on the basin attraction in which the initial state lies.

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.9 Hopfield Network: *stability using C-G theorem*

Remember:

$$C_i\frac{da_i(t)}{dt}=-\tfrac{1}{R_i}a_i(t)+\sum_j w_{ji}f_j(a_j(t))+\theta_i \qquad (2.9.5)$$

Another way to show that the Hopfield network is stable is to apply the Cohen-Grossberg theorem given in section 2.8. For this purpose we reorganize the Eq. (2.9.5) as:

$$\frac{da_i(t)}{dt}=\tfrac{1}{C_i}((-\tfrac{1}{R_i}a_i(t)+\theta_i)-\sum_j(-w_{ji})f_j(a_j(t))) \qquad (2.9.18)$$

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.9 Hopfield Network

Rem.

$$\frac{d}{dt}a_i = \alpha_i(a_i)(\beta_i(a_i) - \sum_{j=1}^{n} w_{ji} f_j(a_j)) \quad i = 1..N \qquad (2.8.1)$$

$$\frac{da_i(t)}{dt} = \frac{1}{C_i}((-\frac{1}{R_i}a_i(t) + \theta_i) - \sum_j (-w_{ji}) f_j(a_j(t))) \qquad (2.9.18)$$

If we compare Eq. (2.9.18) with Eq. (2.8.2) we recognize that in fact Hopfield network is a special case of the system defined in Cohen-Grossberg theorem:

$$w_{ij} \Leftrightarrow -w_{ij} \qquad (2.9.19)$$

and

$$\alpha_i(a_i) \leftrightarrow \frac{1}{C_i} \qquad (2.9.20)$$

and

$$\beta(a_i(t)) \leftrightarrow -\frac{a_i(t)}{R_i} + \theta_i \qquad (2.9.21)$$

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.9 Hopfield Network

It satisfies the conditions on

a) symmetry because $w_{ij}=w_{ij}$ implies

$$-w_{ij}=-w_{ji} \qquad (2.9.22)$$

b) nonnegativity because

$$\alpha_i(a_i) = \frac{1}{C_i} > 0 \qquad (2.9.23)$$

c) monotonocity because of

$$f'(a) = \frac{d}{dt}\tanh(\kappa a) \geq 0 \qquad (2.9.24)$$

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.9 Hopfield Network

Therefore, according to the Cohen-Grossberg theorem, the energy function defined as

$$E = \tfrac{1}{2}\sum_i\sum_j (-w_{ij}) f(a_i) f(a_j) - \sum_i \int_0^{a_i} (-\tfrac{1}{R_i} a + \theta_i) f'(a)da \qquad (2.9.25)$$

is a Lyapunov function of the Hopfield network and the network is globally asymptotically stable.

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.9 Hopfield Network

Remember:

$$E = -\tfrac{1}{2}\sum_i \sum_j w_{ji} x_j x_i + \sum_i \tfrac{1}{R_i}\int_0^{x_i} f_i^{-1}(x)dx - \sum_i \theta_i x_i \qquad (2.9.7)$$

$$E = \tfrac{1}{2}\sum_i\sum_j (-w_{ij}) f(a_i) f(a_j) - \sum_i \int_0^{a_i} (-\tfrac{1}{R_i} a + \theta_i) f'(a)da \qquad (2.9.25)$$

In fact, the energy equation defined by equation (2.2.25) can be easily reorganized as the one given in equation (2.9.7) (see lecture notes)

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.9 Hopfield Network

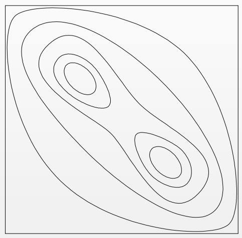


Figure 2.10 Energy contour map for a two neuron two stable system

As the time derivative of the Energy function is negative, the change in the state value of the network is in a direction where the energy decreases.

The behavior of a Hopfield network of two neurons is demonstrated in the Figure 2.10 [Hopfield 84]. In the figure the ordinate and absisca are the outputs of each neuron. The network has two stable states and they are located near the upper left and lower right corners.

ANN ee543

CHAPTER II : *Recurrent Neural Networks*

2.9 Hopfield Network

The second term of the energy function in Eq. (2.9.7), which is

$$\sum_{i=1}^{N} \frac{1}{N} \int_{0}^{x_i} f^{-1}(x)dx \qquad (2.9.30)$$

alters the energy landscape.

The value of the gain parameter determines how close the stable points come to the hypercube corners.

In the limit of very high gain, $\kappa \rightarrow \infty$, this term approaches to zero and the stable points of the system lie just at the corners of the Hamming hypercupe where the value of each state component is either -1 or 1.

For finite gain, the stable points move towards the interior of the hypercube. As the gain becomes smaller, these stable points gets closer. When $\kappa=0$, only a single stable point exists for the system Therefore the choice of the gain parameter is quite important for the success of the operation