Chapter 13. Neurodynamics

Neural Networks and Learning Machines
(Haykin)

Lecture Notes on Self-learning Neural Algorithms

Byoung-Tak Zhang
School of Computer Science and Engineering
Seoul National University

Version 20171109
13.1 Introduction ........................................................................................................................................ 3
13.2 Dynamic Systems .................................................................................................................................. 5
13.3 Stability of Equilibrium States .............................................................................................................. 8
13.4 Attractors .................................................................................................................................................. 14
13.5 Neurodynamic Models .......................................................................................................................... 15
13.6 Attractors and Recurrent Networks ...................................................................................................... 19
13.7 Hopfield Model ...................................................................................................................................... 20
13.8 Cohen-Grossberg Theorem .................................................................................................................. 27
13.9 Brain-State-In-A-Box Model ................................................................................................................. 29
Summary and Discussion ............................................................................................................................... 33
13.1 Introduction (1/2)

- **Time plays a critical role in learning.** Two ways in which time manifests itself in the learning process:
  1. A static neural network (NN) is made into a dynamic mapper by stimulating it via a *memory structure*, short term or long term.
  2. Time is built into the operation of a neural network through the use of *feedback*.

- Two ways of applying *feedback* in NNs:
  1. *Local feedback*, which is applied to a single neuron inside the network;
  2. *Global feedback*, which encompasses one or more layers of hidden neurons—or better still, the whole network.

- **Feedback** is like a double-edged sword in that when it is applied improperly, it can produce harmful effects. In particular, the application of feedback can cause a system that is originally stable to become unstable. Our primary interest in this chapter is in the stability of recurrent networks.

- The subject of neural networks viewed as *nonlinear dynamic systems*, with particular emphasis on the *stability* problem, is referred to as *neurodynamics*. 
13.1 Introduction (2/2)

• An important feature of the stability (or instability) of a nonlinear dynamic system is that it is a property of the whole system.
  – *The presence of stability always implies some form of coordination between the individual parts of the system.*

• The study of neurodynamics may follow one of two routes, depending on the application of interest:
  – *Deterministic neurodynamics*, in which the neural network model has a deterministic behavior. Described by a set of *nonlinear differential equations* ➔ This chapter.
  – *Statistical neurodynamics*, in which the neural network model is perturbed by the presence of noise. Described by *stochastic nonlinear differential equations*, thereby expressing the solution in probabilistic terms.
13.2 Dynamic Systems (1/3)

A dynamic system is a system whose state varies with time.

\[
\frac{d}{dt} x_j(t) = F_j(x_j(t)), \quad j = 1, 2, \ldots, N
\]

\[
\frac{d}{dt} \mathbf{x}(t) = \mathbf{F}(\mathbf{x}(t))
\]

\[
\mathbf{x}(t) = [x_1(t), x_2(t), \ldots, x_N(t)]^T
\]

Figure 13.1  A two-dimensional trajectory (orbit) of a dynamic system.
13.2 Dynamic Systems (2/3)

Figure 13.2  A two-dimensional state (phase) portrait of a dynamic system.

Figure 13.3  A two-dimensional vector field of a dynamic system.
13.2 Dynamic Systems (3/3)

Lipschitz condition

Let $\|x\|$ denote the norm, or Euclidean length, of the vector $x$.

Let $x$ and $u$ be a pair of vectors in an open set $M$ in a normal vector (state) space. Then, according to the Lipschitz condition, there exists a constant $K$ for all $x$ and $u$ in $M$.

$$\|F(x) - F(u)\| \leq K \|x - u\|$$

Divergence Theorem

$$\int_S (F(x) \cdot n) dS = \int_V (\nabla \cdot F(x)) dV$$

If the divergence $\nabla \cdot F(x)$ (which is a scalar) is zero, the system is conservative, and if it is negative, the system is dissipative.
13.3 Stability of Equilibrium States (1/6)

Table 13.1

<table>
<thead>
<tr>
<th>Type of Equilibrium State $\bar{x}$</th>
<th>Eigenvalues of the Jacobian $A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stable node</td>
<td>Real and negative</td>
</tr>
<tr>
<td>Stable focus</td>
<td>Complex conjugate with negative real parts</td>
</tr>
<tr>
<td>Unstable node</td>
<td>Real and positive</td>
</tr>
<tr>
<td>Unstable focus</td>
<td>Complex conjugate with positive real parts</td>
</tr>
<tr>
<td>Saddle point</td>
<td>Real with opposite signs</td>
</tr>
<tr>
<td>Center</td>
<td>Conjugate purely imaginary</td>
</tr>
</tbody>
</table>

$F(\bar{x}) = 0$

$x(t) = \bar{x} + \Delta x(t)$

$F(x) \approx \bar{x} + A \Delta x(t)$

$A = \frac{\partial}{\partial x} F(x) \bigg|_{x = \bar{x}}$

$\frac{d}{dt} \Delta x(t) \approx A \Delta x(t)$
13.3 Stability of Equilibrium States (2/6)

Figure 13.4  (a) Stable node. (b) Stable focus. (c) Unstable node. (d) Unstable focus. (e) Saddle point. (f) Center.
Lyapunov’s Theorems

Theorem 1. The equilibrium state $\bar{x}$ is stable if, in a small neighborhood of $\bar{x}$, there exists a positive-definite function $V(x)$ such that its derivative with respect to time is negative semidefinite in that region.

Theorem 2. The equilibrium state $\bar{x}$ is asymptotically stable if, in a small neighborhood of $\bar{x}$, there exists a positive-definite function $V(x)$ such that its derivative with respect to time is negative definite in that region.
13.3 Stability of Equilibrium States (4/6)

Figure 13.5 Illustration of the notion of uniform stability of a state vector.
Requirement :
The Lyapunov function $V(x)$ to be a positive-definite function

1. The function $V(x)$ has continuous partial derivatives with respect to the elements of the state $x$.
2. $V(\bar{x}) = 0$.
3. $V(x) > 0$ if $x \in u - \bar{x}$ where $u$ is a small neighborhood around $\bar{x}$.

According to Theorem 1,
\[
\frac{d}{dt} V(x) \leq 0 \quad \text{for } x \in u - \bar{x}
\]

According to Theorem 2,
\[
\frac{d}{dt} V(x) < 0 \quad \text{for } x \in u - \bar{x}
\]
13.3 Stability of Equilibrium States (6/6)

Figure 13.6 Lyapunov surfaces for decreasing value of constant $c$, with $c_1 < c_2 < c_3$. The equilibrium state is denoted by the point $\bar{x}$.
13.4 Attractors

- A $k$-dimensional surface embedded in the $N$-dimensional state space, which is defined by the set of equations

$$M_j(x_1, x_2, \ldots, x_N) = 0, \begin{cases} j = 1, 2, \ldots, k \\ k < N \end{cases}$$

- These manifolds are called **attractors** in that they are bounded subsets to which regions of initial conditions of a nonzero state-space volume converge as time $t$ increases.

- Point attractors
- Limit cycle
- Basin (domain) of attraction
- Separatrix
- Hyperbolic attractor

Figure 13.7  Illustration of the notion of a basin of attraction and the idea of a separatrix.
13.5 Neurodynamic Models (1/4)

• General properties of the neurodynamic systems

1. A large number of degrees of freedom
   The human cortex is a highly parallel, distributed system that is estimated to possess about 10 billion neurons, with each neuron modeled by one or more state variables. The system is characterized by a very large number of coupling constants represented by the strengths (efficacies) of the individual synaptic junctions.

2. Nonlinearity
   A neurodynamic system is inherently nonlinear. In fact, nonlinearity is essential for creating a universal computing machine.

3. Dissipation
   A neurodynamic system is dissipative. It is therefore characterized by the convergence of the state-space volume onto a manifold of lower dimensionality as time goes on.

4. Noise
   Finally, noise is an intrinsic characteristic of neurodynamic systems. In real-life neurons, membrane noise is generated at synaptic junctions
13.5 Neurodynamic Models (2/4)

Figure 13.8 Additive model of a neuron, labeled j.
13.5 Neurodynamic Models (3/4)

Additive model

\[
C_j \frac{dv_j(t)}{dt} + \frac{v_j(t)}{R_j} = \sum_{i=1}^{N} w_{ji} x_i(t) + I_j
\]

\[
x_j(t) = \varphi(v_j(t))
\]

\[
C_j \frac{dv_j(t)}{dt} = -\frac{v_j(t)}{R_j} + \sum_{i=1}^{N} w_{ji} x_i(t) + I_j, \quad j = 1, 2, \ldots, N
\]

\[
\varphi(v_j) = \frac{1}{1 + \exp(-v_j)}, \quad j = 1, 2, \ldots, N
\]
13.5 Neurodynamic Models (4/4)

Related model

\[ \frac{dv_j(t)}{dt} = -v_j(t) + \sum_{i} w_{ji} \varphi(v_i(t)) + I_j, \quad j = 1, 2, ..., N \]

\[ \frac{dx_j(t)}{dt} = -x_j(t) + \varphi\left(\sum_{i} w_{ji} x_i(t)\right) + K_j, \quad j = 1, 2, ..., N \]

\[ v_k(t) = \sum_{j} w_{kj} x_j(t) \]

\[ I_k = \sum_{j} w_{kj} K_j \]
A neurodynamic model can have complicated attractor structures and may therefore exhibit useful computational capabilities.

The identification of attractors with computational objects is one of the foundations of neural network paradigms.

One way in which the collective properties of a neural network may be used to implement a computational task is by way of the concept of energy minimization.

The Hopfield network and brain-state-in-a-box model are examples of an associative memory with no hidden neurons; an associative memory is an important resource for intelligent behavior. Another neurodynamic model is that of an input–output mapper, the operation of which relies on the availability of hidden neurons.
13.7 Hopfield Model (1/7)

- The *Hopfield network (model)* consists of a set of neurons and a corresponding set of unit-time delays, forming a *multiple-loop feedback system*.

- To study the dynamics of the Hopfield network, we use the neurodynamic model which is based on the additive model of a neuron.

\[
E = -\frac{1}{2} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} w_{ji} x_i x_j
\]

Figure 13.9  Architectural graph of a Hopfield network consisting of \(N = 4\) neurons.
The Discrete Hopfield Model as a Content-Addressable Memory

- The induced local field $v_j$ of neuron $j$ is defined by

$$v_j = \sum_{i=1}^{N} w_{ji} x_i + b_j$$

where $b_j$ is a fixed bias applied externally to neuron $j$. Hence, neuron $j$ modifies its state $x_j$ according to the deterministic rule

$$x_j = \begin{cases} +1 & \text{if } v_j > 0 \\ -1 & \text{if } v_j < 0 \end{cases}$$

- This relation may be rewritten in the compact form

$$x_j = \text{sgn}(v_j)$$
The Discrete Hopfield Model as a Content-Addressable Memory

Storage of samples (= learning)

According to the outer-product rule of storage—that is, the generalization of Hebb’s postulate of learning—the synaptic weight from neuron $i$ to neuron $j$ is defined by

$$w_{ji} = \frac{1}{N} \sum_{\mu=1}^{M} \xi_{\mu,j} \xi_{\mu,i}$$

$$w_{ii} = 0 \quad \text{for all } i$$

Figure 13.13  Illustration of the encoding–decoding performed by a recurrent network.
The Discrete Hopfield Model as a Content-Addressable Memory

1. Storage Phase

\[ W = \frac{1}{N} \sum_{\mu=1}^{M} \xi_{\mu} \xi_{\mu}^T - MI \]

- The output of each neuron in the network is fed back to all other neurons.
- There is no self-feedback in the network (i.e., \( w_{ij} = 0 \)).
- The weight matrix of the network is symmetric, as shown by the following relation

\[ W^T = W \]

2. Retrieval Phase

\[ y_j = \text{sgn} \left( \sum_{i=1}^{N} w_{ji} y_i + b_j \right), \quad j = 1, 2, \ldots, N \]

\[ y = \text{sgn}(Wy + b) \]
13.7 Hopfield Model (5/7)

TABLE 13.2 Summary of the Hopfield Model

1. **Learning.** Let $\xi_1, \xi_2, \ldots, \xi_N$ denote a known set of $N$-dimensional fundamental memories. Use the outer-product rule (i.e., Hebb’s postulate of learning) to compute the synaptic weights of the network as

$$w_{ji} = \begin{cases} 
\frac{1}{N} \sum_{\mu=1}^{N} \xi_{\mu,i} \xi_{\mu,j}, & j \neq i \\
0, & j = i
\end{cases}$$

where $w_{ji}$ is the synaptic weight from neuron $i$ to neuron $j$. The elements of the vector $\xi_{i}$ equal ±1. Once they are computed, the synaptic weights are kept fixed.

2. **Initialization.** Let $\xi_{\text{probe}}$ denote an unknown $N$-dimensional input vector (probe) presented to the network. The algorithm is initialized by setting

$$x_j(0) = \xi_{j, \text{probe}}, \quad j = 1, \ldots, N$$

where $x_j(0)$ is the state of neuron $j$ at time $n = 0$ and $\xi_{j, \text{probe}}$ is the $j$th element of the probe $\xi_{\text{probe}}$.

3. **Iteration Until Convergence.** Update the elements of state vector $x(n)$ asynchronously (i.e., randomly and one at a time) according to the rule

$$x_j(n+1) = \text{sgn} \left( \sum_{i=1}^{N} w_{ji} x_i(n) \right), \quad j = 1, 2, \ldots, N$$

Repeat the iteration until the state vector $x$ remains unchanged.

4. **Outputting.** Let $x_{\text{fixed}}$ denote the fixed point (stable state) computed at the end of step 3. The resulting output vector $y$ of the network is

$$y = x_{\text{fixed}}$$

Step 1 is the storage phase, and steps 2 through 4 constitute the retrieval phase.
Figure 13.14  (a) Architectural graph of Hopfield network for $N = 3$ neurons. (b) Diagram depicting the two stable states and flow of the network.
Spurious States

- An eigenanalysis of the weight matrix $W$ leads us to take the following viewpoint of the discrete Hopfield network used as a content-addressable memory.
  1. The discrete Hopfield network acts as a vector projector in the sense that it projects a probe onto a subspace spanned by the fundamental memory vectors.
  2. The underlying dynamics of the network drive the resulting projected vector to one of the corners of a unit hypercube where the energy function is minimized.

- Tradeoff between two conflicting requirements:
  1. The need to preserve the fundamental memory vectors as fixed points in the state space.
  2. The desire to have few spurious states.
13.8 The Cohen-Grossberg Theorem (1/2)

\[
\frac{d}{dt} u_j = a_j(u_j) \left[ b_j(u_j) - \sum_{i=1}^{N} c_{ji} \varphi_i(u_i) \right], \quad j = 1, 2, \ldots, N
\]

\[
E = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ji} \varphi_i(u_i) \varphi_j(u_j) - \sum_{j=1}^{N} \int_0^{u_j} b_j(\lambda) \varphi_j'(\lambda) d\lambda
\]

\[
c_{ij} = c_{ji}
\]

\[
a_j(u_j) \geq 0
\]

\[
\varphi_j'(u_j) = \frac{d}{du_j} \varphi_j(u_j) \geq 0
\]
13.8 The Cohen-Grossberg Theorem (2/2)

The Hopfield Model as a Special Case of the Cohen–Grossberg Theorem

\[
E = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ji} \phi_i(v_i) \phi_j(v_j) + \sum_{j=1}^{N} \int_{0}^{v_j} \left( \frac{v_j}{R_j} - I_j \right) \phi'_j(v) dv
\]

<table>
<thead>
<tr>
<th>TABLE 13.3</th>
<th>Correspondences between the Cohen–Grossberg Theorem and the Hopfield Model</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cohen–Grossberg Theorem</strong></td>
<td><strong>Hopfield Model</strong></td>
</tr>
<tr>
<td>(u_j)</td>
<td>(C_j v_j)</td>
</tr>
<tr>
<td>(a_j(u_j))</td>
<td>1</td>
</tr>
<tr>
<td>(b_j(u_j))</td>
<td>(-(v_j/R_j) + I_j)</td>
</tr>
<tr>
<td>(c_{ji})</td>
<td>(-w_{ji})</td>
</tr>
<tr>
<td>(\varphi_i(u_i))</td>
<td>(\varphi_i(v_i))</td>
</tr>
</tbody>
</table>
13.9 Brain-State-in-a-Box Model (1/4)

\[
y(n) = x(n) + \beta W x(n)
\]

\[
x(n + 1) = \varphi(y(n))
\]

\[
x_j(n+1) = \varphi(y_j(n)) = \begin{cases} 
+1 & \text{if } y_j(n) > +1 \\
y_i(n) & \text{if } -1 \leq y_i(n) \leq +1 \\
-1 & \text{if } y_i(n) < -1 
\end{cases}
\]

**Figure 13.15** (a) Block diagram of the brain-state-in-a-box (BSB) model. (b) Signal-flow graph of the linear associator represented by the weight matrix \( W \).

(c) 2017 Biointelligence Lab, SNU
BSB Model

\[ x_j(n+1) = \varphi \left( \sum_{i=1}^{N} c_{ji} x_i(n) \right), \quad j=1,2,\ldots,N \]

\[ c_{ji} = \delta_{ji} + \beta w_{ji} \]

Continuous form

\[ \frac{d}{dt} x_j(t) = -x_j(t) + \varphi \left( \sum_{i=1}^{N} c_{ji} x_i(t) \right), \quad j=1,2,\ldots,N \]

To transform this into the additive model, we introduce

\[ v_j(t) = \sum_{i=1}^{N} c_{ji} x_i(t) \quad x_j(t) = \sum_{i=1}^{N} c_{ji} v_i(t) \]

Equivalent form

\[ \frac{d}{dt} v_j(t) = -v_j(t) + \sum_{i=1}^{N} c_{ji} \varphi(v_i(t)), \quad j=1,2,\ldots,N \]


13.9 Brain-State-in-a-Box Model (3/4)

Lyapunov (Energy) Function of BSB

\[ E = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ji} \varphi(v_i) \varphi(v_j) + \sum_{j=1}^{N} \int_{0}^{v_j} v \varphi'(v) dv \]

\[ E = -\frac{\beta}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ji} x_j x_i = -\frac{\beta}{2} x^T W x \]

Dynamics of the BSB Model

- It can be demonstrated that the BSB model is a gradient descent algorithm that minimizes the energy function \( E \).
- The equilibrium states of the BSB model are defined by certain corners of the unit hypercube and its origin.
- For every corner to serve as a possible equilibrium state of BSB, the weight matrix \( W \) should be diagonal dominant, which means that \( w_{jj} \geq \sum_{i \neq j} |w_{ij}| \) for \( j = 1, 2, \ldots, N \).
- For an equilibrium state to be stable, i.e. for a certain corner of the unit hypercube to be a fixed point attractor, the weight matrix \( W \) should be strongly diagonal dominant: \( w_{jj} \geq \sum_{i \neq j} |w_{ij}| + \alpha \) for \( j = 1, 2, \ldots, N \).

| TABLE 13.4 Correspondences between the Cohen–Grossberg Theorem and the BSB Model |
|---------------------------------|------------------|
| Cohen–Grossberg Theorem         | BSB Model        |
| \( u_j \)                      | \( v_j \)        |
| \( a_j(u_j) \)                  | 1                |
| \( b_j(u_j) \)                  | \(-v_j\)         |
| \( c_{ji} \)                    | \(-c_{ji}\)      |
| \( \varphi_j(u_j) \)            | \( \varphi_j(v_j) \) |
13.9 Brain-State-in-a-Box Model (4/4)

Clustering

Figure 13.17  Illustrative example of a two-neuron BSB model, operating under four different initial conditions:
• the four shaded areas of the figure represent the model’s basins of attraction;
• the corresponding trajectories of the model are plotted in blue;
• the four corners, where the trajectories terminate, are printed as black dots.
Summary and Discussion

- **Dynamic Systems**
  - Stability, Liapunov functions
  - Attractors

- **Neurodynamic Models**
  - Additive model
  - Related model

- **Hopfield Model (Discrete)**
  - Recurrent network (associative memory)
  - Liapunov (energy) function
  - Content-addressable memory

- **BSB Model**
  - Liapunov function
  - Clustering