The Generalized Brain-State-in-a-Box (gBSB) Neural Network: Model, Analysis, and Applications

Cheolhwan Oh---Purdue
Stefen Hui---SDSU
Stanisław H. Żak---Purdue
October 18, 2005
Outline

- Modeling
  - Linear associative memory
  - Brain-State-in-a-Box (BSB) neural net
  - The generalized BSB (gBSB)

- Analysis
  - Stability definitions
  - Stability tests

- Applications
  - Associative memory design using gBSB net
  - Large scale gBSB nets
  - Storing and retrieving images
Neural Associative Memory

- Associative memory: a memory that can be accessed by contents---Content Addressable Memory (CAM)

- Autoassociative memory: stored patterns are retrieved from their distorted versions
Heteroassociative memory: a set of input patterns is paired with a different set of output patterns.

Two stages of neural associative memory operation:
- **Storage phase**: Patterns are stored by the neural network.
- **Recall phase**: Memorized patterns are retrieved in response to given initial patterns.
Linear Associative Memory (LAM)

- \( W = \sum_{j=1}^{r} v^{(j)} v^{(j)T} \)

- \( v^{(1)}, \ldots, v^{(r)} \) are mutually orthogonal pattern vectors

- LAM recovers stored pattern from the partial input

Let \( v^{(k)} = v^{(k1)} + v^{(k2)} \), \( v^{(k1)} \perp v^{(k2)} \), \( v^{(k1)} \perp \{v^{(1)}, \ldots v^{(k-1)}, v^{(k+1)}, \ldots, v^{(r)}\} \)

Then

\[
y = Wv^{(k1)} = \sum_{j=1}^{r} v^{(j)} v^{(j)T} v^{(k1)} = v^{(k)} v^{(k)T} v^{(k1)}
\]

\[
= (v^{(k1)} + v^{(k2)})(v^{(k1)} + v^{(k2)})^T v^{(k1)}
\]

\[
= (v^{(k1)} + v^{(k2)}) v^{(k2)T} v^{(k1)} = v^{(k)} v^{(k2)T} v^{(k1)} = c v^{(k)}
\]
Limitations of Linear Associative Memory (LAM)

- Patterns must be orthogonal to each other
- When the input vector is a noisy version of the prototype pattern, the LAM cannot recall the corresponding prototype pattern
Brain-State-in-a-Box (BSB) neural network

- BSB model---a nonlinear dynamical system
- Proposed by Anderson, Silverstein, Ritz and Jones (1977) as a memory model based on neurophysiological considerations
- Used to model effects and mechanisms seen in psychology and cognitive science
Brain-State-in-a-Box (BSB) neural network (continued)

- Can be used to recognize a pattern from its given noisy version

- The network trajectory is constrained to be in the hypercube $H_n = [-1, 1]^n$
Dynamics of the BSB neural model

\[ x(k+1) = g(x(k) + \alpha Wx(k)), \quad x(0)=x_0 \]

- \( x_0 \): initial condition
- \( x(k) \in \mathbb{R}^n \): state of the BSB net at time \( k \)
- \( \alpha \in \mathbb{R} \): step size
- \( W \in \mathbb{R}^{n \times n} \): symmetric weight matrix
- \( g: \mathbb{R}^n \to \mathbb{R}^n \): activation function
  - (standard linear saturation function)
### Activation function

- **Standard linear saturation function (used in BSB model)**

\[
(g(x))_i = (sat(x))_i = \begin{cases} 
1 & \text{if } x_i \geq 1 \\
x_i & \text{if } -1 < x_i < 1 \\
-1 & \text{if } x_i \leq -1 
\end{cases}
\]

where \(x_i\) is the \(i\)-th element of the vector \(x\)
Super stable equilibrium state

Equilibrium state:
A point $x_e$ is an equilibrium state of the dynamical system $x(k+1) = T(x(k))$ if $x_e = T(x_e)$

Super stable equilibrium state:
An equilibrium state $x_e$ is super stable if there exists a neighborhood of $x_e$, denoted $N(x_e)$, such that for any initial state $x_0 \in N(x_e)$, the trajectory starting from $x_0$ reaches $x_e$ in a finite number of steps
Basin of attraction

A basin of attraction of an equilibrium state of the BSB net---the set of points such that the trajectory of the BSB net emanating from any point in the set converges to the equilibrium state.
Characteristics of BSB based neural associative memories

- If an equilibrium state is a vertex of the hypercube $H_n = [-1, 1]^n$, then belongs to the set $\{-1, 1\}^n$
- Pattern vectors are stored as super stable equilibrium vertices
- If an initial state (noisy pattern) is located in the basin of attraction of a certain stored pattern, the network trajectory starting from the given initial state converges to this pattern---we say that recall is successful
Generalized Brain-State-in-a-Box (gBSB) neural network

- Proposed by Hui and Žak (1992)
- Weight matrix can be nonsymmetric
  - → easier to implement
- Offers more control of the volume of the basin of attraction of an equilibrium state
Dynamics of gBSB model

\[ x(k+1) = g((I_n + \alpha W)x(k) + \alpha b), \quad x(0) = x_0 \]

\( W \in \mathbb{R}^{n \times n} \): weight matrix (not necessarily symmetric)
\( b \in \mathbb{R}^n \): bias vector
\( I_n \in \mathbb{R}^n \): identity matrix
\( g : \mathbb{R}^n \rightarrow \mathbb{R}^n \): standard linear saturation function
Stability condition for gBSB neural model

- Let \( \mathbf{v} = [v_1 \ v_2 \ \cdots \ v_n]^T \in \{-1, 1\}^n \)
  
  (\( \mathbf{v} \) is a vertex of \( H_n \))

- \( \mathbf{v} \) is an equilibrium state of the gBSB model if and only if
  
  \[(W \mathbf{v} + \mathbf{b})_i v_i \geq 0 \quad \text{for} \quad i=1,\ldots,n\]

- \( \mathbf{v} \) is a super stable equilibrium state of the gBSB model if
  
  \[(W \mathbf{v} + \mathbf{b})_i v_i > 0 \quad \text{for} \quad i=1,\ldots,n\]

- The above stability conditions are also called the vertex stability conditions
Synthesis of gBSB neural associative memory

- Weight matrix construction algorithm of Lillo, Miller, Hui and Žak (1994)

- Given pattern vectors to be stored:
  \( \mathbf{v}^{(j)} \in \{-1, 1\}^n, \quad j = 1, 2, \ldots, r \)

- Form a matrix \( \mathbf{B} = [\mathbf{b} \, \mathbf{b} \, \ldots \, \mathbf{b}]^T \in \mathbb{R}^{n \times r} \), where

\[
\mathbf{b} = \sum_{j=1}^{n} \mathbf{\varepsilon v}^{(j)}, \quad \mathbf{\varepsilon}_j \geq 0, \quad j = 1, 2, \ldots, r
\]
Choose $D \in \mathbb{R}^{n \times n}$ such that

$$d_{ii} > \sum_{j=1, j \neq i}^{n} |d_{ij}|, \quad \text{and}$$

$$d_{ii} < \sum_{j=1, j \neq i}^{n} |d_{ij}| + |b_i|, \quad i = 1, 2, ..., n$$

Choose $\Lambda \in \mathbb{R}^{n \times n}$ such that

$$\lambda_{ii} < -\sum_{j=1, j \neq i}^{n} |\lambda_{ij}| - |b_i|, \quad i = 1, 2, ..., n$$
Synthesis of gBSB neural associative memories (continued)

- Determine the weight matrix $W$, 
  $$W = (DV - B)V^\dagger + \Lambda(I_n - VV^\dagger),$$
  where $V = [v^{(1)} \ v^{(2)} \ldots \ v^{(r)}] \in \{-1, 1\}^{n \times r}$ and $V^\dagger$ is a pseudo-inverse matrix of $V$

- The above implementation procedure guarantees that the given patterns are stored as super stable vertices of $H_n$
Simulation experiments

- Prototype patterns

- Stacking operator

\[ X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{m \times n}, \quad \text{where} \quad x_i \in \mathbb{R}^m, \ i = 1, 2, \ldots, n \]

\[ s(X) = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} : \text{column vector} \]
Examples

- Example 1

- Example 2
Large scale neural associative memory design

- Neural associative memory: a memory that can be accessed by content and is constructed using an artificial neural network
- Difficulty with neural associative memory design: quadratic growth of the number of interconnections with the problem size
- Large scale patterns
  → Large scale neural network
  → Heavy computational overhead
- Proposed approach---apply pattern decomposition
Large scale neural associative memory design using pattern decomposition

- Advantages:
  Computationally efficient
  (Smaller size of weight matrices)

- Disadvantages of pattern decomposition:
  Small size patterns
  → Small size neural networks
  → Deterioration of recall performance---reduced capacity, more spurious states
Associative memory design using overlapping pattern decomposition

- **Purpose**
  - Take the advantage of decomposed neural associative memory
  - Enhance recall performance

- **Design**
  Each neural subnetwork is constructed independently of other subnetworks
Ring overlapping pattern decomposition

- **Disjoint decomposition**

  \[
  \begin{array}{cccccc}
  v_1 & v_k & v_{k+1} & v_{2k} & v_{2k+1} & v_{pk}
  \end{array}
  \]

- **Overlapping decomposition**

  \[
  \begin{array}{cccccc}
  v_1 & v_k & v_{k+1} & v_{2k} & v_{pk}
  \end{array}
  \]

- **Overlapping decomposition with a ring structure**

  \[
  \begin{array}{cccccc}
  v_1 & v_k & v_{k+1} & v_{2k} & v_{pk} & v_1
  \end{array}
  \]
Toroidal overlapping pattern decomposition
Proposed associative memory system

- **Storage phase**
  - Decompose prototype patterns into subpatterns with overlapping pattern decomposition structure
  - Construct subnetworks independently of each other
Proposed associative memory system (continued)

- Recall phase
  - Initial pattern decomposed into subpatterns
  - Subnetworks process corresponding initial subpatterns (retrieval processes)
  - Check the overlapping portions and apply error correction procedure
  - Recombine the output subpatterns
Error correction for vector patterns

Input subpatterns

\[ \mathbf{x}^{(i)} = [x_{(i-1)k+1} \cdots x_{ik} \ x_{ik+1}] \]

\[ \mathbf{x}^{(i+1)} = [x_{ik+1} \cdots x_{(i+1)k} \ x_{(i+1)k+1}] \]

Output subpatterns of the retrieval processes

\[ \mathbf{v}^{(i)} = [v_{(i-1)k+1}^{(i)} \cdots v_{ik}^{(i)} \ v_{ik+1}^{(i)}] \]

\[ \mathbf{v}^{(i+1)} = [v_{ik+1}^{(i+1)} \cdots v_{(i+1)k}^{(i+1)} \ v_{(i+1)k+1}^{(i+1)}] \]
Error correction for vector patterns (continued)

- Check if there are mismatches in the overlapping portions between neighboring subpatterns, i.e.,
  \[ v_{ik+1}^{(i)} = v_{ik+1}^{(i+1)} \]?

- If \( v_{ik+1}^{(i)} \neq v_{ik+1}^{(i+1)} \), recall error occurred

- Error correction algorithm
  - Form a modified input subpattern and go through retrieval process again with the modified input subpattern
    \[ \hat{x}^{(i)} = [x_{(i-1)k+1} \ldots x_{ik} v_{ik+1}^{(i+1)}] \]
Error correction for matrix patterns

- Direct extension of the error correction procedure for the vector patterns
- Replace overlapping rows or overlapping columns of a matrix with the corresponding rows or columns of neighboring subpatterns
Image pattern processing with large scale neural associative memory
Black and white image patterns

Pattern size: 200-by-200 pixel
Black and white image pattern recall

- 25% of pixels complemented
- 400 networks

initial pattern

prototype pattern

output pattern (overlapping decomposition)

output pattern (disjoint decomposition)
Comparison of two types of memory
Black and white logo patterns---150-by-150
Logo recall example---30% pixels complemented

initial pattern

after error correction

after recall (25 networks)

output pattern
Gray scale image patterns---150-by-150, 6-bit
Gray scale image pattern recall---100 networks, salt-and-pepper noise (40%)

noisy initial image  output image

Salt-and-pepper noise---a 6-bit pixel randomly selected, replaced with 0s or 1s
Color image patterns---150-by-200, RGB 4 bit each
Color image pattern recall---300 networks, Gaussian noise, SD=2

noisy input image

after recall (before error correction)

Gaussian noise---a random number generated with Gaussian distribution added to each pixel
Color image pattern recall (continued)

after error correction

output image
Pattern sequence storage and retrieval with gBSB based neural associative memory

- Store sequences of patterns
- Recall the stored pattern sequence when a noisy pattern sequence is presented to the neural memory
- Composed of autoassociative and heteroassociative parts
  - Autoassociative part: noise elimination/reduction → constructed with gBSB neural network
  - Heteroassociative part: sequencing patterns
Pattern sequence storage and retrieval network model

- Dynamics of the proposed system:
  \[ x(k+1) = g \{ (1 - c(k)) \left( (I_n + \alpha W_a) x(k) + \alpha b \right) + c(k) W_h x(k) \}, \]

  \[ c(k) = \begin{cases} 
  1 & \text{for heteroassociative steps}, \\
  0 & \text{for autoassociative steps}. 
\end{cases} \]
Construct the autoassociative part using the algorithm of Lillo et al.

Construct the weight matrix of heteroassociative part:

\[ W_h = V_2 V_1^\dagger \]

where

\[ V_1 = [v_1^{(1)} \ v_2^{(1)} \ldots \ v_{L_1-1}^{(1)} \ v_{L_1}^{(1)} | \ldots | v_1^{(M)} \ v_2^{(M)} \ldots \ v_{LM-1}^{(M)} \ v_{LM}^{(M)} ] \]

and

\[ V_2 = [v_2^{(1)} \ v_3^{(1)} \ldots \ v_{L_1}^{(1)} \ v_1^{(1)} | \ldots | v_2^{(M)} \ v_3^{(M)} \ldots \ v_{LM}^{(M)} \ v_1^{(M)} ] \]

given \( M \) pattern sequences to store

\[ v_1^{(1)} \rightarrow v_2^{(1)} \rightarrow \ldots \rightarrow v_{L_1}^{(1)}, \quad \ldots \quad , \quad v_1^{(M)} \rightarrow v_2^{(M)} \rightarrow \ldots \rightarrow v_{LM}^{(M)} \]
Pattern Sequence Storage and Retrieval---Example
(24-by-20 bitmap images)

- Prototype pattern sequences

<table>
<thead>
<tr>
<th>Pattern Sequence 1</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern Sequence 2</td>
<td>I</td>
<td>J</td>
<td>K</td>
<td>L</td>
<td>M</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pattern Sequence 3</td>
<td>O</td>
<td>P</td>
<td>Q</td>
<td>R</td>
<td>S</td>
<td>T</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>Pattern Sequence 4</td>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Pattern Sequence Storage and Retrieval---continued
(each bit complemented with 20% probability)

- Snapshots of the pattern sequence recall
Large scale associative memory design using pattern sequence storage and retrieval

- Decompose large scale patterns into small size patterns and link them to obtain pattern sequences
- Store the sequences of the small size patterns in the neural associative memory that can store pattern sequences
When a noisy initial large scale pattern is presented to the neural associative memory, it is decomposed into a sequence of small patterns.

The neural memory recalls a corresponding prototype pattern sequence.

Recombine the small patterns to obtain a large scale pattern.
Example---160-by-200, 8 bit
(48 networks, Gaussian noise, SD=30)

k=0 (input)  k=9  k=10

k=19  k=20  k=25
Example (continued)

- k=26
- k=27
- k=69
- k=70
- k=71
- k=72 (output)
Summary

- Associative memories used to image pattern recall
- Large scale associative memory design using pattern decomposition method presented
- Interconnected neural associative memories proposed
- Novel associative memory design method using overlapping decomposition and error correction procedure presented
Future research

- Develop high performance neural associative memories that can process large scale patterns in cost-effective ways
- Apply neural associative memories to large scale patterns such as images or three dimensional objects
- Use gBSB net as basis for a neural information system to analyze and classify proteins
Thank You