How do biological neural networks encode, learn, memorize, recall, and generalize as a “learning machine”?

James Ting-Ho Lo  
Department of Mathematics and Statistics  
University of Maryland Baltimore County  
Baltimore, MD 21250  
jameslo@umbc.edu
“We require exquisite numerical precision over many logical steps to achieve what brains accomplish in very few short steps.”

The Computer and the Brain, 1958, p. 63.
A mathematical approach

biological postulates
  ↓
mathematical inference

learning machine
  ↓
biological interpretation

computational model of neural network
Results

1. An ANN or learning machine
   THPAM (Temporal Hierarchical Probabilistic Associative Memory)
   *Cognitive Neurodynamics*, 2010

2. A biologically plausible model
   LOM (Low-Order Model of Biological NNs)
   *Neural Computation*, 2011

3. An ANN or learning machine
   CIPAM (Clustering Interpreting Probabilistic Associative Memory)
   *Neurocomputing*, 2012
Low-Order Model (LOM)

Single model providing logically coherent answers to holy-grail questions:

• What information does a spike train carry?
• What computation do dendritic trees do?
• How is supervised learning performed?
• How is unsupervised learning performed?
• How is information stored in synapses?
• What computation do spiking and nonspiking somas do?
• How are corrupted, distorted and occluded patterns recognized?
• How are neurons organized into a neural network?
Single model providing logically coherent answers to holy-grail questions:

- What information does a spike train carry?
- What computation do dendritic trees do?
- How is supervised learning performed?
- How is unsupervised learning performed?
- How is information stored in synapses?
- What computation do spiking and nonspiking somas do?
- How are corrupted, distorted and occluded patterns recognized?
- How are neurons organized into a neural network?
What information **should be** communicated between spiking neurons?
Recognition of 26 Upper Case Letters:

\[ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z \]

What letter is this?   \[ ▲ \]
What is the best representation?

A       W

Relative frequencies of the 26 letters:

A  8.17%  (binary code 01000001)
W  2.36%  (binary code 01010111)

8 somas needed to generate 8 bits.
Each bit assigned by the probability:
8.17/ (8.17+2.36) or 2.36/ (8.17+2.36)

Conjecture:
Neurons operate in groups generating labels.
Single model answering holy-grail questions:

- What information does a spike train carry?
- **What computation do dendritic trees do?**
- How is supervised learning performed?
- How is unsupervised learning performed?
- How is information stored in synapses?
- What computation do spiking and nonspiking somas do?
- How corrupted, distorted and occluded patterns recognized?
- How are neurons organized into a neural network?
Dendrites

- Use more than 60% of the energy consumed by the brain
- Occupy more than 99% of the surface of some neurons
- Are the largest component of neural tissue in volume

Yet, dendrites are missing in all ANNs.
A model of a dendrite tree

Outputs from 3 neurons:
\[ v_\tau = [v_{\tau_1} \ v_{\tau_2} \ v_{\tau_3}] \text{ binary} \]

Take all the $2^3$ combinations of the 3 entries:
\{ \}, \{v_1\}, \{v_2\}, \{v_2, v_1\}, \{v_3\}, \{v_3, v_1\}, \{v_3, v_2\}, \{v_3, v_2, v_1\}

Apply a function $\phi$ to each combinations:
0, $\phi(v_1), \phi(v_2), \phi(v_2, v_1), \phi(v_3), \phi(v_3, v_1), \phi(v_3, v_2), \phi(v_3, v_2, v_1)$

These are the outputs of the dendritic tree
A model of a dendritic tree

Input from axons:
\[ v_\tau = [v_{\tau 1}, v_{\tau 2}, v_{\tau 3}]' \] binary (essentially)

Neurons

Output by an axon:
\[ u_\tau \]

A dendritic branch is a parity function

A dendritic encoder
HYPOTHESIS

A dendritic branch is a parity function \( \phi : \)
\[
\phi(v_1, v_2, \cdots, v_n) =
\begin{cases}
1 & \text{if } v_1 + v_2 + \cdots + v_n \text{ is odd} \\
0 & \text{if } v_1 + v_2 + \cdots + v_n \text{ is even}
\end{cases}
\]

Experiment required to test the hypothesis:
Measure the inputs and output of dendritic branches.
\[ v_r = \begin{bmatrix} v_{r1} & v_{r2} & v_{r3} \end{bmatrix} \]
\[ \tilde{v}_r = \begin{bmatrix} 0 & v_{r1} & v_{r2} & \phi(v_{r2}, v_{r1}) & v_{r3} & \phi(v_{r3}, v_{r1}) & \phi(v_{r3}, v_{r2}) & \phi(v_{r1}, v_{r2}, v_{r3}) \end{bmatrix} \]

\[ \tilde{v}_r = ([1 \ 0 \ 1]') = [0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0]' \]
\[ \tilde{v}_t = ([0 \ 1 \ 1]') = [0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0]' \]

A dendritic encoder
ORTHOGONALITY OF DENDRITIC CODES

\[ \tilde{v}_r = ([1 \ 0 \ 1]) = [0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0] \]
\[ \tilde{v}_t = ([0 \ 1 \ 1]) = [0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0] \]
\[ \mathbf{1} = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1] \]

\[ \left( \tilde{v}_r - \frac{1}{2} \mathbf{1} \right)' = \left[ -\frac{1}{2} \ \frac{1}{2} \ -\frac{1}{2} \ \frac{1}{2} \ \frac{1}{2} \ -\frac{1}{2} \ \frac{1}{2} \ -\frac{1}{2} \right] \]
\[ = \frac{1}{2} \left[ -1 \ 1 \ -1 \ 1 \ 1 \ -1 \ 1 \ -1 \right] \]

\[ \left( \tilde{v}_t - \frac{1}{2} \mathbf{1} \right)' = \left[ -\frac{1}{2} \ -\frac{1}{2} \ \frac{1}{2} \ \frac{1}{2} \ \frac{1}{2} \ \frac{1}{2} \ -\frac{1}{2} \ -\frac{1}{2} \right] \]
\[ = \frac{1}{2} \left[ -1 \ -1 \ 1 \ 1 \ 1 \ -1 \ -1 \right] \]

\[ \frac{1}{2} \left( \tilde{v}_r - \frac{1}{2} \mathbf{1} \right)' \left( \tilde{v}_t - \frac{1}{2} \mathbf{1} \right) = 0 \]
\[ \frac{1}{2} \left( \tilde{v}_t - \frac{1}{2} \mathbf{1} \right)' \left( \tilde{v}_t - \frac{1}{2} \mathbf{1} \right) = \frac{1}{2} \left( \frac{1}{2} \right) \left( \frac{1}{2} \right) (8) = 1 \]
General Dendritic Code

Inputs to a neuron

- Avoiding the curse of dimensionality
- Enhancing generalization

Each color-coded low-resolution input field is uniformly distributed
Single model answering holy-grail questions:

- What information does a spike train carry?
- What computation do dendritic trees do?
- **How is supervised learning performed?**
- How is unsupervised learning performed?
- How is information stored in synapses?
- What computation do spiking and nonspiking somas do?
- How corrupted, distorted and occluded patterns recognized?
- How are neurons organized into a neural network?
Supervised Covariance Learning

Hebbian-type rule:

\[ D_{ij} \leftarrow D_{ij} + (w_{ti} - \langle w_{ti} \rangle)(\bar{v}_{ij} - \langle \bar{v}_{ij} \rangle) \]
Single model answering holy-grail questions:

• What information does a spike train carry?
• What computation do dendritic trees do?
• How is supervised learning performed?
• **How is unsupervised learning performed?**
• How is information stored in synapses?
• What computation do spiking and nonspiking somas do?
• How corrupted, distorted and occluded patterns recognized?
• How are neurons organized into a neural network?
Unsupervised Covariance Learning

Hebbian-type rule:  
\[ D_{ij} \leftarrow D_{ij} + (u_{ti} - \langle u_{ti} \rangle)(\tilde{v}_{ij} - \langle \tilde{v}_{ij} \rangle) \]

T. J. Sejnowski 1977:  
\[ D_{ij} \leftarrow D_{ij} + (u_{ti} - \langle u_{ti} \rangle)(v_{ij} - \langle v_{ij} \rangle) \]
Single model answering holy-grail questions:

- What information does a spike train carry?
- What computation do dendritic trees do?
- How is supervised learning performed?
- How is unsupervised learning learning performed?
- **How is information stored in synapses?**
- What computation do spiking and nonspiking somas do?
- How corrupted, distorted and occluded patterns recognized?
- How are neurons organized into a neural network?
Code Covariance Matrix $D$

*learned by* supervised learning

$$D_{ij} \leftarrow D_{ij} + (w_{ti} - <w_{ti}>)(\bar{v}_{tj} - <\bar{v}_{tj}>)$$

$$D_{ij} = \sum_{t=1}^{T} (w_{ti} - <w_{ti}>)(\bar{v}_{tj} - <\bar{v}_{tj}>)$$
Code Covariance Matrix $D$

*learned by*

unsupervised learning

\[
D_{ij} \leftarrow D_{ij} + (u_{ti} - \langle u_{ti} \rangle)(\bar{v}_{tj} - \langle \bar{v}_{tj} \rangle)
\]

\[
D_{ij} = \sum_{t=1}^{T} (u_{ti} - \langle u_{ti} \rangle)(\bar{v}_{tj} - \langle \bar{v}_{tj} \rangle)
\]
Decovariance Retrieval

\[ d_{\tau ij} = D_{ij} \left( \tilde{v}_{\tau j} - < \tilde{v}_{\tau j} > \right) \]

Synapse with \( D_{ij} \)

\[ \tilde{v}_{\tau j} \quad \rightarrow \quad d_{\tau ij} \]
Single model answering holy-grail questions:

• What information does a spike train carry?
• What computation do dendritic trees do?
• How is supervised learning performed?
• How is unsupervised learning performed?
• How is information stored in synapses?
• **What computation do spiking and nonspiking somas do?**
• How corrupted, distorted and occluded patterns recognized?
• How are neurons organized into a neural network?
Spiking Soma $k$

If $c_\tau = 0$, the set $p_{tk} = 1/2$, 
else 
$$p_{tk} = \frac{1}{2} \left( \frac{\sum_j d_{tkj}}{c_\tau} \right) + \frac{1}{2}. $$

Generate a pseudo-random number $v\{p_{tk}\}$:

$$P(v\{p_{tk}\} = 1) = p_{tk}$$
$$P(v\{p_{tk}\} = 0) = 1 - p_{tk}$$

$- c_\tau$
Single model answering holy-grail questions:

- What information does a spike train carry?
- What computation do dendritic trees do?
- How is supervised learning performed?
- How is unsupervised learning performed?
- How is information stored in synapses?
- What computation do spiking and nonspiking neurons do?
- **How are corrupted, distorted and occluded patterns recognized?**
- How are neurons organized into a neural network?
Maximal Generalization

RF (Receptive Field)
of a PU

RF subfield
w/o error

RF subfield
w/ error

GENERALIZATION:
Using information in an RF subfield
w/o error to estimate the label of the RF

MAXIMAL GENERALIZATION:
Generalization from the largest RF
subfield w/o error

Maximal region <=> Best subjective probability available
\( \phi(\vec{v}_{r1}, \vec{v}_{r2}) \)

\( \phi(\vec{v}_{r1}, \vec{v}_{r2}, \vec{v}_{r3}) \)

\( \phi(\vec{v}_{r2}, \vec{v}_{r3}) \)

\( (\text{diag} \hat{I}(1^-))\vec{v}_r \)

\( (\text{diag} \hat{I}(2^-))\vec{v}_r \)

\( (\text{diag} \hat{I}(3^-))\vec{v}_r \)
\[ \mathbf{v} = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}' \]
\[ \mathbf{\hat{v}} = \begin{bmatrix} 1 & v_1 & v_2 v_1 & v_3 v_1 & v_3 v_2 & v_3 v_2 v_1 \end{bmatrix}' \]
\[ \mathbf{\hat{i}}(1^-) = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}' \]
\[ \mathbf{\hat{i}}(2^-) = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}' \]
\[ \mathbf{\hat{i}}(3^-) = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}' \]
\[ \sum_{i=1}^{3} \mathbf{\hat{i}}(i^-) = \begin{bmatrix} 3 & 2 & 2 & 1 & 2 & 1 & 1 & 0 \end{bmatrix}' \]
\[ \text{diag } \mathbf{\hat{i}}(1^-) \mathbf{\hat{v}} = \begin{bmatrix} 1 & 0 & v_2 & 0 & v_3 & 0 & v_3 v_2 & 0 \end{bmatrix}' \]
\[ \text{diag } \mathbf{\hat{i}}(2^-) \mathbf{\hat{v}} = \begin{bmatrix} 1 & v_1 & 0 & 0 & v_3 & v_3 v_1 & 0 & 0 \end{bmatrix}' \]
\[ \text{diag } \mathbf{\hat{i}}(3^-) \mathbf{\hat{v}} = \begin{bmatrix} 1 & v_1 & v_2 & v_2 v_1 & 0 & 0 & 0 & 0 \end{bmatrix}' \]
\[ \text{diag } \mathbf{\hat{i}}(i_1^-, \cdots, i_j^-) \mathbf{\hat{v}} \] sets the \( i_1 \text{th}, \cdots, i_j \text{th} \]
\[ \text{components of } \mathbf{v} \text{ in } \mathbf{\hat{v}} \text{ set equal to 0} \]
Example Masking Matrix $M$

$$M = I + 2^{-5} \sum_{i=1}^{3} \text{diag} \hat{I}(i^-)$$

$$M = I + 2^{-5} \begin{bmatrix}
3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

a diagonal matrix
Masking Matrix $M$

$$M = I + \sum_{j=1}^{J} 2^{-5j} \sum_{i_{j}=j}^{m} \sum_{i_{2}=2}^{i_{j}-1} \sum_{i_{1}=1}^{i_{2}-1} \text{diag} \hat{I}(i_{1}^{-}, \ldots, i_{j}^{-})$$

$$M = \begin{bmatrix} M_{11} & 0 & \cdots & 0 \\ 0 & M_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & M_{qq} \end{bmatrix}$$

a diagonal matrix

Generalization from the largest receptive subfield that matches a stored subfield
Single model answering holy-grail questions:

- What information does a spike train carry?
- What computation do dendritic trees do?
- How is supervised learning performed?
- How is unsupervised learning performed?
- How is information stored in synapses?
- What computation do spiking and nonspiking somas do?
- How corrupted, distorted and occluded patterns recognized?
- **How are neurons organized into a neural network?**
A pattern recognizer
Supervised Processing Unit (SPU)

1. Dendritic Encoders
2. Learning
3. Retrieval from Synapses
4. Computing probabilities
5. Max. generalization by \( M \)
6. Generation of spikes

\[
\begin{align*}
\tau & \quad \text{Delay} \\
& \quad \text{Learning Mechanism} \\
& \quad \text{Covariance Learning} \\
& \quad \text{Synapses & Masking Matrix} \quad M \\
& \quad \text{Probability Estimator} \\
& \quad R \text{ spiking somas} \\
& \quad 1 \text{ nonspiking soma} \\
& \quad \text{Spike Generator} \\
& \quad p_{\tau} \\
& \quad R_{\text{spiking somas}} \\
& \quad 1_{\text{nonspiking soma}} \\
\end{align*}
\]
Unsupervised Processing Unit (UPU)

2. Learning

Covariance Learning

Learning Mechanism

Delay

(Delay)

$\tau$

1/2

Synapses & Masking Matrix $M$

$C$

$D$

$C_j M_{jj} \left( \bar{v}_j - \langle \bar{v}_j \rangle \right)$

$d_{skj} = D_{kj} M_{jj} \left( \bar{v}_j - \langle \bar{v}_j \rangle \right)$

$[c_{\tau}]$

$[d_{skj}]$

3. Retrieval from Synapses

5. Max. generalization by $M$

6. Generation of spikes

$R$ spiking somas
1 nonspiking soma

Spike Generator

Probability Estimator

For $k = 1, ..., R$, if $\sum_j c_{\tau j} = 0$, then set $y_{\tau k} = 0$, else

$y_{\tau k} = \sum_j d_{skj} / \sum_j c_{\tau j}$

$p_{\tau k} = (y_{\tau k} + 1)/2$

1. Dendritic Encoders

3. Retrieval from Synapses

4. Computing probabilities

5. Max. generalization by $M$
A Network of Unsupervised Processing Units (UPUs)
Offshoot Supervised Processing Units (SPUs)

Clusterer

Interpreter

\[ \{ p_{\tau}(1^3) \} \]

\[ \{ p_{\tau}(1^2) \} \]

\[ \{ p_{\tau}(2^2) \} \]

\[ r_{\tau}(1^3) \]

\[ r_{\tau}(1^2) \]

\[ r_{\tau}(2^2) \]

\[ r_{\tau}(3^2) \]
Current and Future Work

• **Applications**

  - Spatial Pattern Recognition
    Handwriting, face, target, fingerprint, DNA, smell, taste, explosive/weapon (in baggage, containers)

  - Temporal Pattern Understanding/Classification
    Touch, speech, text, video (computer vision), financial data

• **Theory**

  - Extension to visual, auditory, gustatory, olfactory, somatosensory, and somatomotor systems

  - Motion detection, attention selection, prediction
Learning Machine for BIG Data

DLMs including CNN are present workhorses.

They are inadequate.
Capabilities needed for learning big data

1. Handcrafting labels impossible for big data
   Learning without supervision

2. Big data too big for iterative optimization
   Learning with photographic memory

3. Big data streaming in
   Online learning

4. Big data not all conditioned for processing
   Maximal generalization (treating noise, distortion, occlusion, translation, scaling, etc.)

5. Big data containing info about hierarchical worlds
   Learning the hierarchical worlds
   (Recall the success of CNN.)

6. Big data containing temporal data (e.g., videos, texts)
   Learning time series

Wish list!!
Capabilities of LOM

1. Learning data w/o handcrafted labels
2. Learning big data with photographic memory
3. Learning streaming data online
4. Maximal generalization (treating noise, distortion, occlusion, translation, scaling, etc.)
5. Learning the hierarchical worlds
   (Recall the success of CNN.)
6. Learning temporal big data (e.g., videos, texts)

Wish list!! Fulfilled!!
LOM for finding all the gene mutations that cause each type of disease and their empirical probabilities

jameslo@umbc.edu
Thank you!

Questions, comments, suggestions?

If you are interested, please talk to or email me.

jameslo@umbc.edu