Implementing high-order cognition in neuromorphic hardware

Terrence C. Stewart
Chris Eliasmith

Centre for Theoretical Neuroscience, University of Waterloo
• World's Largest Functional Brain Model
  - 2.5 million spiking neurons
  - 8 tasks, 1 eye, 1 arm, 20 brain areas

• Problems
  - Slow
    • (2.5 hours on high-end workstation for 1 second of simulation)
  - Power-hungry
    • (scaling up to 100 billion neurons would require a dedicated power station)

• Can neuromorphic hardware help?
Single Neuron

- Internal membrane voltage
- Spike output
- Post-synaptic current
Two Neurons

input → neurons

0.79
Four Neurons
Fifty Neurons
Communicating Between Pools
Communicating Between Pools

\[ J_i = x \cdot e_i \]

\[ a_i = G_i [J_i] \]
Communicating Between Pools

\[ J_i = x \cdot e_i \]

\[ a_i = G_i[J_i] \]

\[ \hat{x} = \sum_i a_i d_i \]
Communicating Between Pools

\[ J_i = x \cdot e_i \]

\[ a_i = G_i[J_i] \]

\[ \hat{x} = \sum_i a_i d_i \]

\[ J_j = \hat{x} \cdot e_j \]

\[ b_j = G_j[J_j] \]
Communicating Between Pools

\[ J_i = x \cdot e_i \]

\[ a_i = G_i[J_i] \]

\[ \hat{x} = \sum_i a_i d_i \]

\[ J_j = \hat{x} \cdot e_j \]

\[ b_j = G_j[J_j] \]
Communicating Between Pools

\[ J_i = x \cdot e_i \]
\[ \omega_{ij} = d_i \cdot e_j \]
\[ a_i = G_i[J_i] \]
\[ J_j = \sum_i \omega_{ij} a_i \]
\[ b_j = G_j[J_j] \]
\[ \hat{x} = \sum_i a_i d_i \]
\[ J_j = \hat{x} \cdot e_j \]
Communication Channel
Computation
Memory
Neural Engineering Framework

- Groups of neurons represent data
  - D-dimensional vectors (D<N)
  - Increasing N decreases error
- Connections compute functions
  - \( y = f(x) \)
  - More non-linear, less accurate
- Recurrent connections compute dynamics
  - \( \dot{x} = f(x) + g(u) \)
  - Need post-synaptic filter of \( h(t) = Ae^{-t/\tau} \)
Cognition

• Concepts are vectors
  – ~700 dimensional
• Memory is an integrator
• Concept combinations are vectors
  – ~700 dimensional
  – RED*CIRCLE + BLUE*SQUARE
  – SUBJECT*TOM + VERB*KNOWS + OBJECT*(SUBJ*DOGS + VERB*CHASE + OBJ*CATS)
  – * is circular convolution
Neuromorphic Hardware

- **Groups of Neurons**
  - ~30 to ~60,000 neurons in a group
  - Any nonlinear neuron model
    - We use standard LIF
    - Customize neuron model based on what's most efficient in the implementation
  - Need to be heterogeneous
    - Varying gains, bias, etc
    - In analog hardware, transistor mismatch is a good thing!
Neuromorphic Hardware

- Connections between neurons
  - Need recurrent connections for memory, dynamics
  - By default, all-to-all
    - But very low-rank weight matrices
    - Weight matrix is factorable: $\mathbf{N} \times \mathbf{N} \rightarrow \mathbf{N} \times \mathbf{D}, \mathbf{D} \times \mathbf{N}; \mathbf{D} \ll \mathbf{N}$
  - Can make weight matrix sparse
    - Lowers accuracy for nonlinear functions
  - Can use probability rather than weights
  - For cognition, need $\sim 60,000$ recurrent for memory
    - Need $60,000 \rightarrow 700 \times 200 \rightarrow 60,000$ for conceptual combination, extraction
Current Progress

• Nengo (http://nengo.ca)
  - Free, Open Source software for building models (including Spaun)
  - Tutorials, Demos, GUI, Scripting

• SpiNNaker
  - Export models from Nengo to SpiNNaker

• Neurogrid
  - Extending Nengo to deal with different types of connection constraints (e.g. input diffusers)
  - Starting 5-year project to develop new hardware
More Information

- Nengo (the software)
  - http://nengo.ca

- Spaun (the functional brain model)

- NEF (the underlying theory)
  - Eliasmith & Anderson (2003). Neural Engineering

- SPA (the cognitive architecture)
  - Eliasmith (2013). How to build a brain