Neuromorphic Architectures
Outline

• What are Neuromorphic Architectures?
  • Biological Comparison
• How Neuromorphic Architectures work?
  • Example 2-input Gate
  • Example: Two Layer Network with Inhibitor
• Why build specialized Neuromorphic Hardware?
  • Same strengths/weaknesses of neural networks
  • Low Power
  • Parallel, real time
• Example Hardware Architectures
  • IBM’s Architecture
  • MoNETA Architecture
• Conclusion
Neuromorphic Architectures

- Computer architectures that are similar to biological brains; computer architectures that implement artificial neural networks in hardware.
- Functional units are composed of neurons, axons, synapses, and dendrites.
- Synapses are connections between two neurons
  - Remembers previous state, updates to a new state, holds the weight of the connection
- Axons and dendrites connect to many neurons/synapses, like long range bus.
Biological vs Hardware

- ~$10^{10}$ synapses/cm²
- ~$10^6$ neurons/cm²
- ~$10^4$ neurons/cortical column
- ~$5 \times 10^8$ long range axons @ 1 Hz

CROSSBAR JUNCTION

CMOS SUBSTRATE

LAMINAR CIRCUIT

HIGH SPEED BUS

- ~$10^{10}$ intersections/cm² @ 100 nm pitch
- $5 \times 10^8$ transistors/cm² @ 500 transistors/neuron
- Layered cortical circuits with ~$10^6$ neurons/cm²
- Multi-Gbit/sec digital comms

Electronic Brain
How They Work

• Neural networks work based on a spike from a neuron.
• Neurons send a spike once a threshold voltage is meant.
• The voltage is added to or subtracted from neurons by what is passed through synapses.
• Signals are sent to a neuron through dendrites and sent out through the axon.
• No "program" to execute
  o Uses two phases of input: Learning and Operating
  o Learning is done by changing what the synapses lets through i.e. the weight of the synapse.
  o Operation results on spikes from neurons based on specific inputs.
Weight Calculation

• Simplest and most common Weight Change Algorithm: Random Weight Change
• Used in learning phase, where the correct output is known
• If error decreased, change by same as last time
• If error increased, change by a random small quantity
• Similar to a genetic algorithm
Simple Neuromorphic Gate

- Purely an example. Linear logic is not this design's strength
- Neural Network that can be trained as any 2 input logic gate
- Requires 3 neurons and 10 synapses
- With each input, define the correct input and let the network develop
Two Layer Architecture with Inhibitor Example

- "Pattern"=any subset of E1 firing
- Every E1 connected to every E2.
- Every E2 connected to I
- Initialized with random small weights, so the neurons act differently
- I causes "winner take all"
- Each E2 corresponds to one pattern
Inhibitor Synapse Weight Calculation

Feed Forward: E1 $\leftrightarrow$ E2
- Goal: detect neurons firing around the same time
- If two spikes are received nearby, strengthen that connection

Feed back: E2 $\leftrightarrow$ E1
- If a neuron in E2 fires after E1, strengthen that connection
  - E1 may have caused that spike
- If a neuron in E2 fires \textit{before} E1, weaken that connection
  - E1 could not have caused the spike
Advantages: Pattern Recognition

• Brains excellent at pattern recognition:
  o Sound: music, language, voices
  o Light: images, faces, objects
• Reduce huge amounts of data, to just the important parts
  o Incredibly important, as we gather magnitudes more data than we can process
• Inefficient at raw numerical computation, but possible
• These strengths could be helpful at finding patterns to increase weather prediction or increase capabilities of robotics along with many others.
• Currently used with co-processors, to boost a von Neumann machine's pattern analysis capabilities.
Disadvantage of Running Simulated Neuromorphic Architectures

• Every neuron operates independently and each spike requires the need to update each synapses.
• Computation and storage occur at the same place
  o distributed throughout the entire architecture
• Software simulations constrain a parallel process into serial operation
  o millions of parallel operations strain the von Neumann architecture
• Impossible for a simulation to run millions of processes parallel in real-time
• Currently, simulations of small portions of brains have been simulated on supercomputers. These simulations have been simulated at 83 times slower than real-time, however, power consumption is huge.
Simulation Disadvantage: Power

- Brain consumes 10-20W, while an equivalent supercomputer would consume MW
- Making use of the pattern recognition capabilities of these architectures would require low power consumption
- Lower frequency required
- Technically, power only needs to be consumed when a neuron fires i.e. a "spike"
  - relatively rare for internal nodes
- Computation=memory
  - no long, power intense wires
The Need for Hardware

• Advantages of pattern recognition could be very useful
• The inability to use this advantage on current architectures with reasonable power and time constraints is lacking
• Research shows these architectures
  o can be produced on current CMOS technology
  o consume little power
  o due to special architecture, preform in real-time
  o have scalability
IBM's Chip

- IBM along with DARPA funding, created 2 brain-like chips
- These chips are not as complex as the human, in fact they have less neurons and synapses than a snail
- Scaling is tough, as increasing # of neurons exponentially increases # of synapses
  - at 500 transistors/synapse, this gets large fast
- What sets these chips apart is that they were done on current CMOS technology
Example Pattern Recognition

- 10 networks each trained to identify one digit as its "pattern"
  - Each receives the input in parallel, and reaches output at the same time
- Every neuron fires proportional to how strong it thinks the input matches the internalized pattern
- The strongest output wins, so "3" is the result
Memristors

- Fourth circuit element that relates Flux and Charge
- TiO$_2$ has high resistance, TiO$_{2-x}$ has low resistance
- Moving the border (the oxidation) changes total resistance
- Effectively a flux controlled resistor
MoNETA: HP and U of Boston's Approach

• This architecture is composed of three types of devices: CPU-type cores, GPU-type cores and memristors.
• CPU-type cores act as neurons; there are few and they are more complex.
• GPU-types cores act as dendrites and axons which have specific type operations.
• Memristors connect the two and act as synapses.
Memristor Synapses

- Synapse made of 3 parts:
  - Weight storage
  - Weight update
  - Weight multiplication
- A single memristor can perform all three functions with 10-20 transistors, instead of 500
  - At least an order of magnitude smaller than CMOS implementation
- Can only do positive weights
  - Circuit is duplicated with inverted inputs
- Operates a higher frequency, because it is inherently a resistor
- Consistent memristor technology not available yet
Conclusion

- "Since our brains are made of memristors, the flood gate is now open for commercialization of computers that would compute like human brains, which is totally different from the von Neumann architecture underpinning all digital computers." Leon Chua, who first predicted memristors' existence
- Neuromorphic Architectures will be the next major step after von Neumann. These architecture will help realize how to create parallel locality-driven architectures.
- Used for what the brain is good at: compressing data into information
- Memristors will reduce power and area of these circuits by an order of magnitude or more
Questions?
References


