A Roadmap for Reverse-Architecting the Brain's Neocortex

J E Smith

FCRC 2019

“There is nothing that is done in the nervous system that we cannot emulate with electronics if we understand the principles of neural information processing.”

— Carver Mead, "Neuromorphic Electronic Systems" ***Proceedings of the IEEE,*** **1990**

Motivation

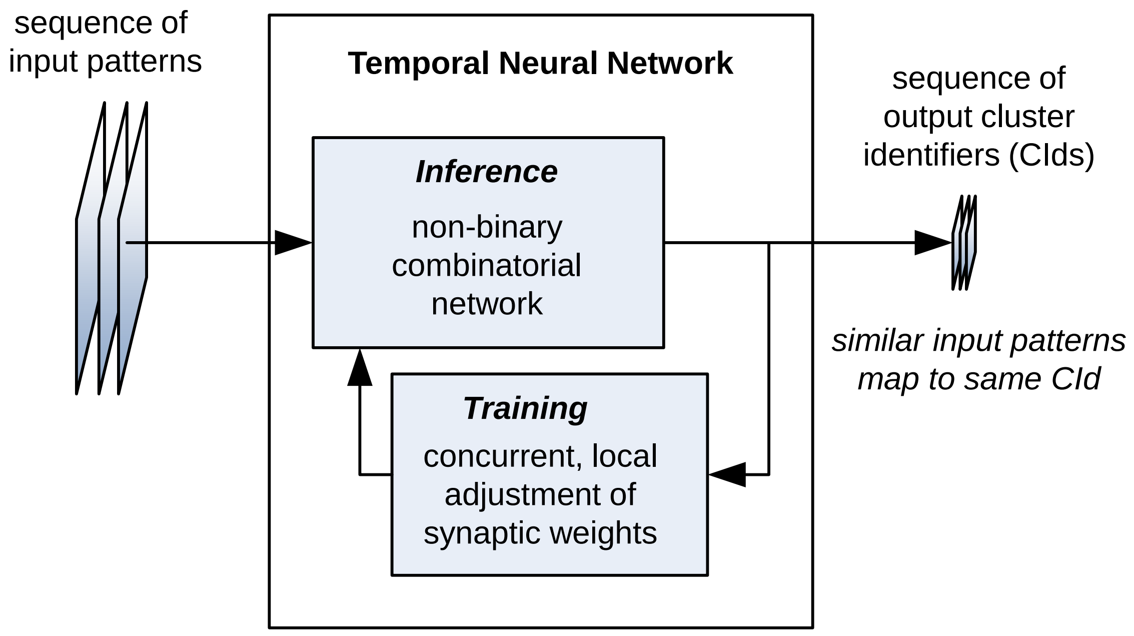
- The human brain is capable of:
  - Accurate sensory perception
  - High level reasoning and problem solving
  - Driving complex motor activity

- With some very impressive features:
  - Extremely efficient (*20 watts*)
  - Very flexible – supports a wide variety of cognitive functions
  - Learns dynamically, quickly, and concurrently with operation
- Far exceeds anything conventional machine learning has achieved
  - Will the trajectory of conventional machine learning *ever* achieve the same capabilities?
  - OR should we seek new approaches based on the way the brain actually works?

the-scientist.com

Milestone Temporal Neural Network

- Continual, Unsupervised Clustering
  - Learn and identify *similar* input patterns and map them to concise *cluster identifiers* (*CId*s)
  - Training and inference done concurrently and continually
- Emergent
  - *All* neural operations are local
  - Global behavior emerges
- Hardware implementation
  - Fast
  - Energy efficient
  - Implementable with digital CMOS
- This is a *processing core*
  - Not a complete system
  - Interfaces with external world will be required
  - For advanced apps this will be challenging



It has a mind of its own!

Outline

- The Biological Neocortex
- Computer Meta-Architecture
- Primitive Abstraction: Biological to Computational
- Column Level Abstraction (“RTL”)
- Mathematical Underpinnings
- Digital CMOS Implementation
- Closing Remarks

The Biological Neocortex

The Neocortex

- Neocortex
  - The “new shell” surrounding the older brain
  - Performs:

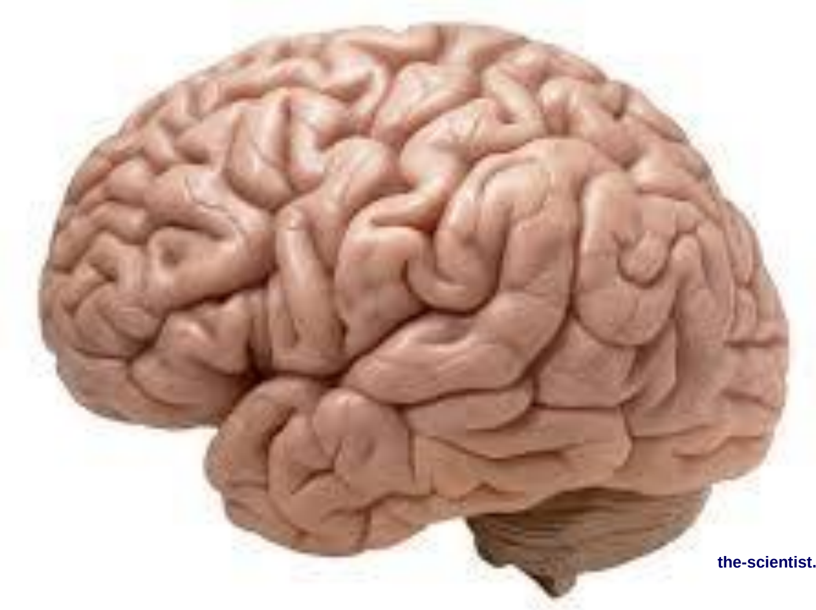
    sensory perception

    cognition

    intellectual reasoning

    generation of high level motor commands

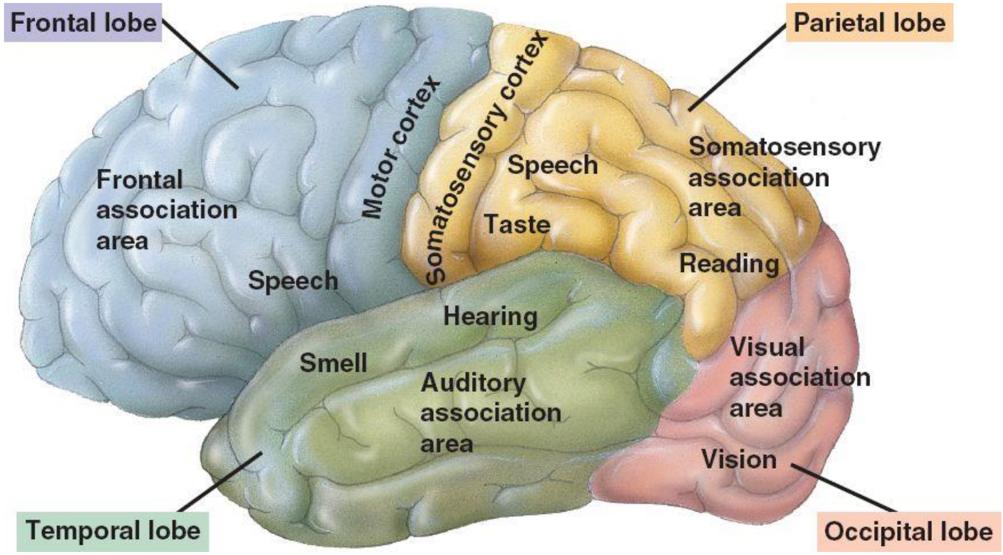
- Thin sheet of neurons
  - 2 to 3 mm thick
  - Area of about 2500 cm^2
  - Folds increase area
  - Approx. 100 billion neurons
  - 10K synapses each



the-scientist.com

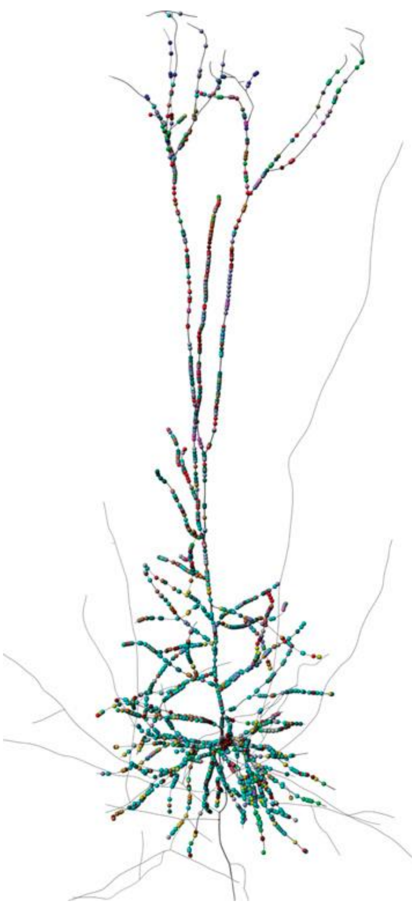
Physical Architecture of the Neocortex

- *Physical* architecture probably corresponds to *functional* architecture
- Physical Hierarchy (top down)
  - Lobes
  - Regions
  - Subregions
  - Macro-Columns
  - Micro-Columns
  - Neurons

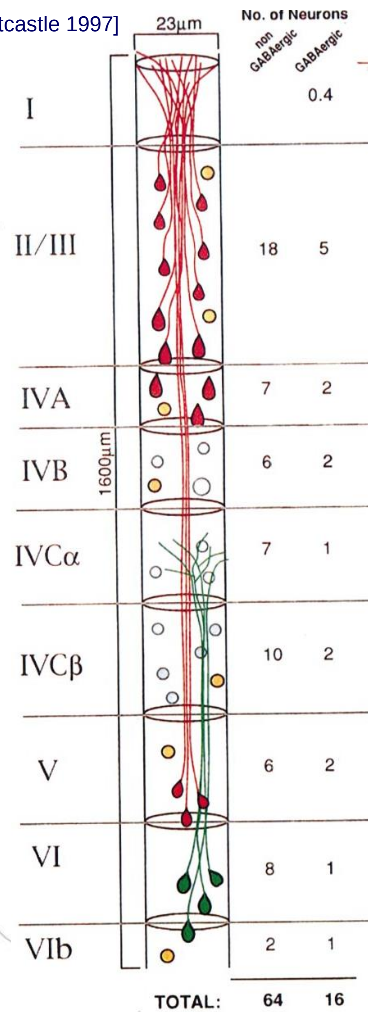


Physical Architecture Bottom-Up

[Hill et al. 2012]

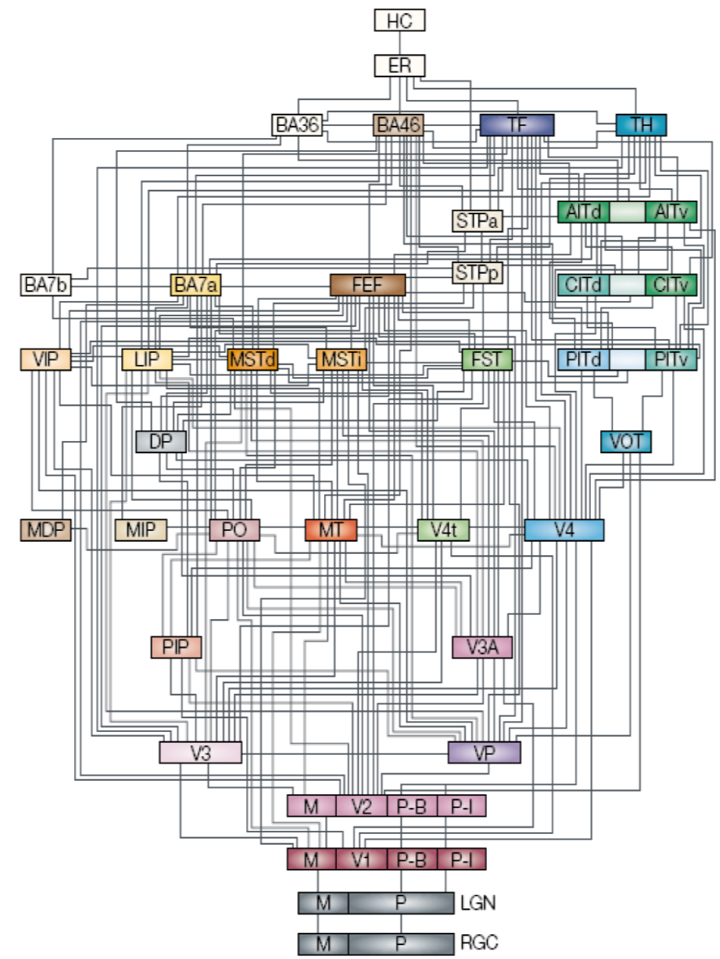


[Mountcastle 1997]



from Ramon y Cajal
(via wikipedia)

[Felleman and Van Essen 1991]



Neuron

Micro-Column
O(100) neurons

Macro-Column
O(100) micro-columns

Regions, Subregions
Many Macro-Columns

Biological Neurons

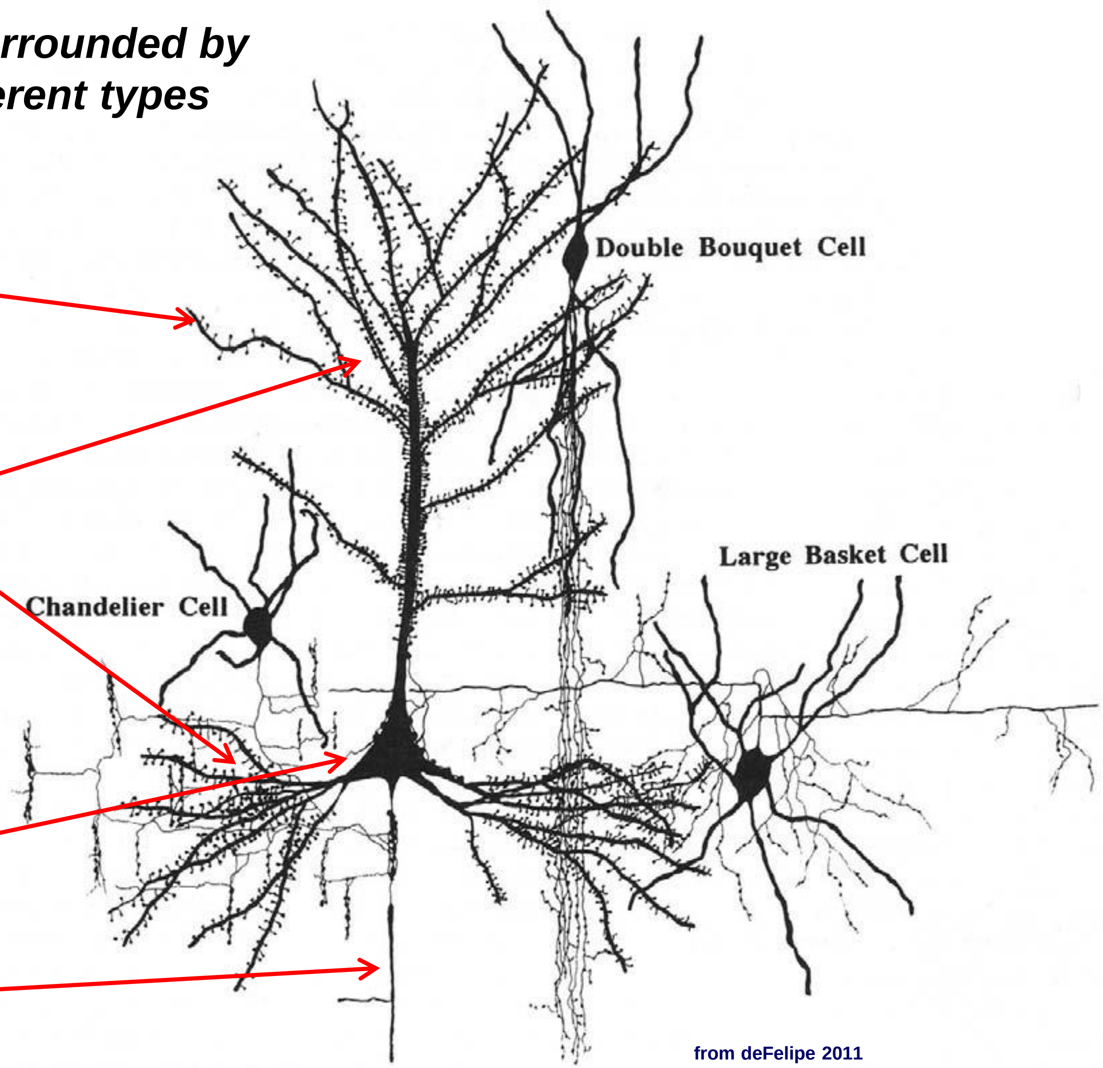
Excitatory pyramidal neuron surrounded by three inhibitory neurons of different types

tiny dots are ***Synapses*** (connection points)

Dendrites (Inputs)

Body

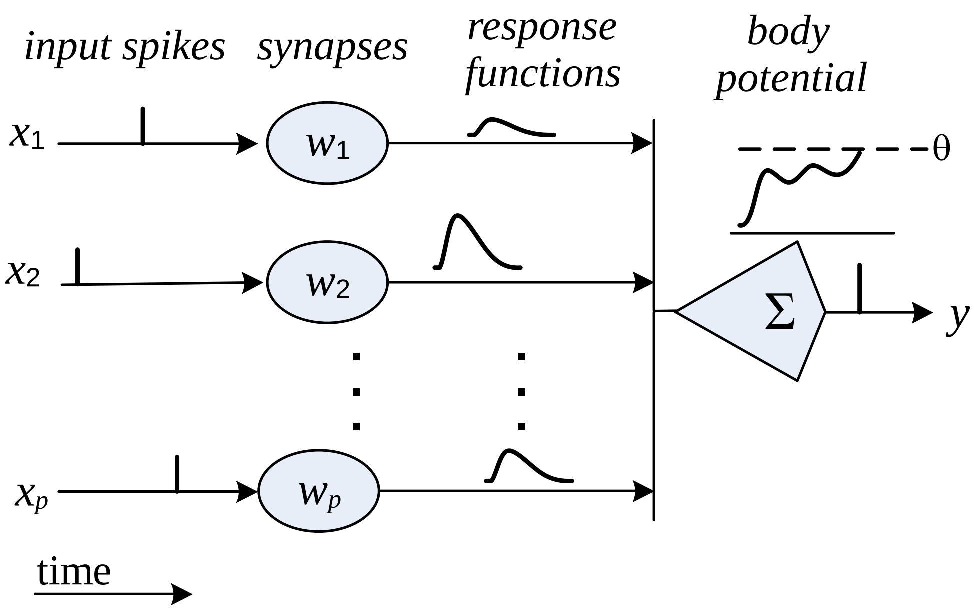
Axon (Output)

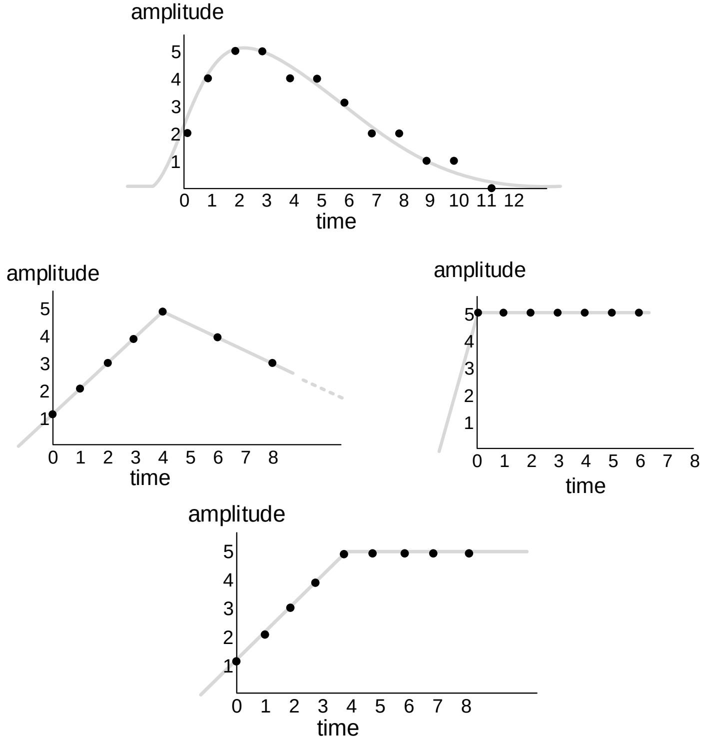


from deFelipe 2011

Excitatory Neuron Model

- Basic Spike Response Model (SRM0) -- Kistler, Gerstner, & van Hemmen (1997)





1) A volley of spikes is applied at inputs

2) At each input's synapse, the spike produces a weighted response function

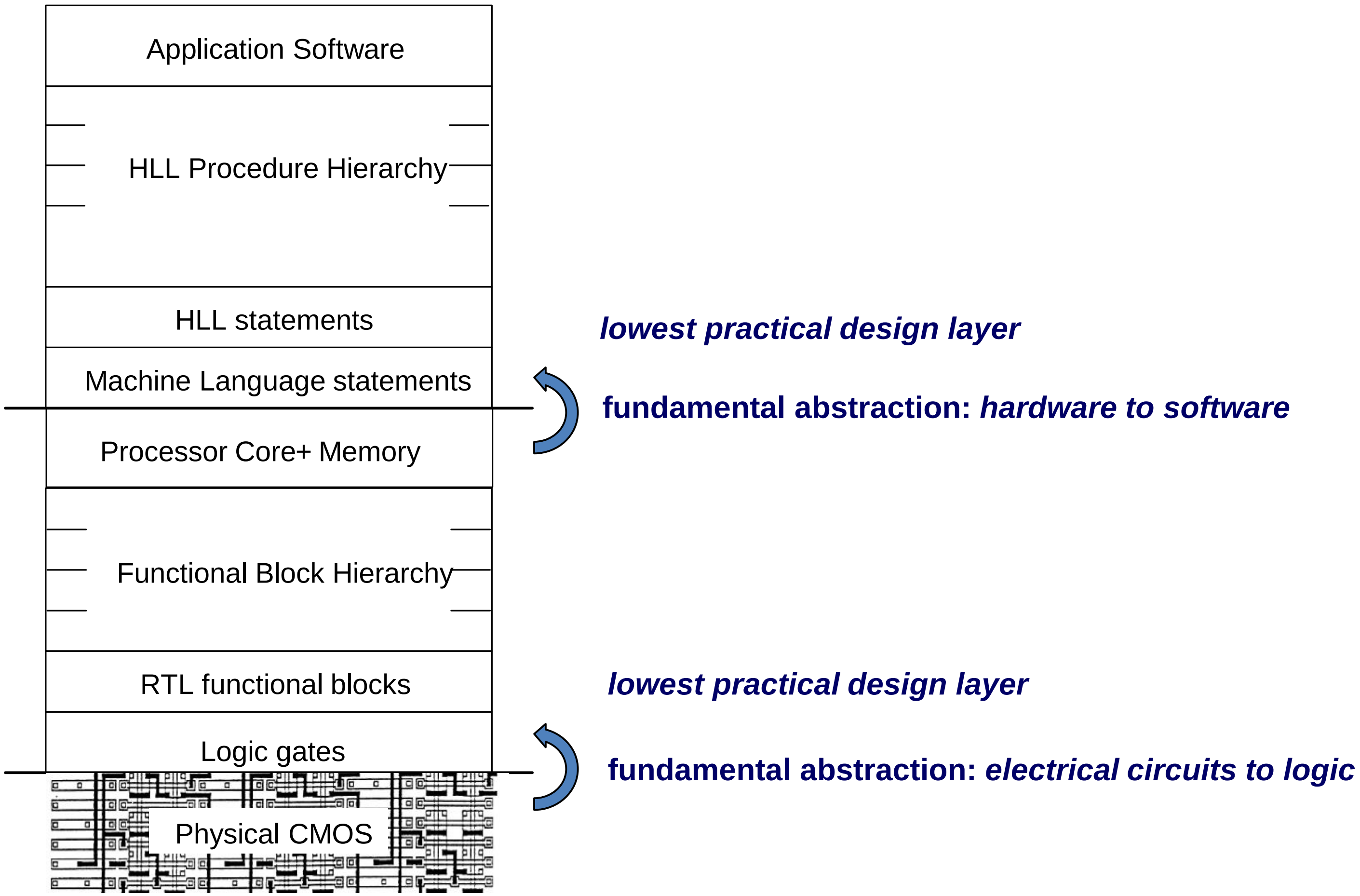
3) Responses are summed linearly at neuron body

4) An output spike is emitted if/when potential exceeds threshold value (θ)

Meta-Architecture

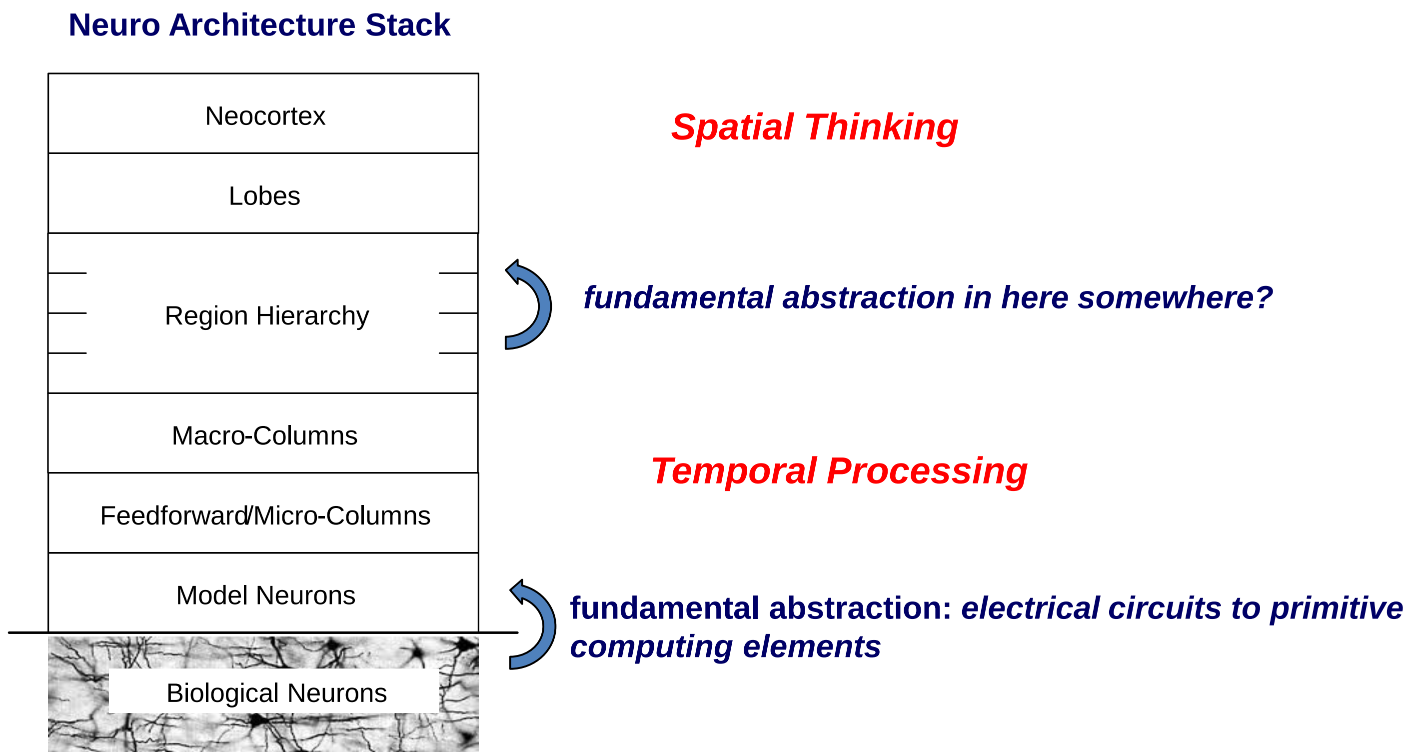
Architecture and Abstraction

- Engineering highly complex systems requires abstraction
  - Conventional computer architecture contains many levels of abstraction



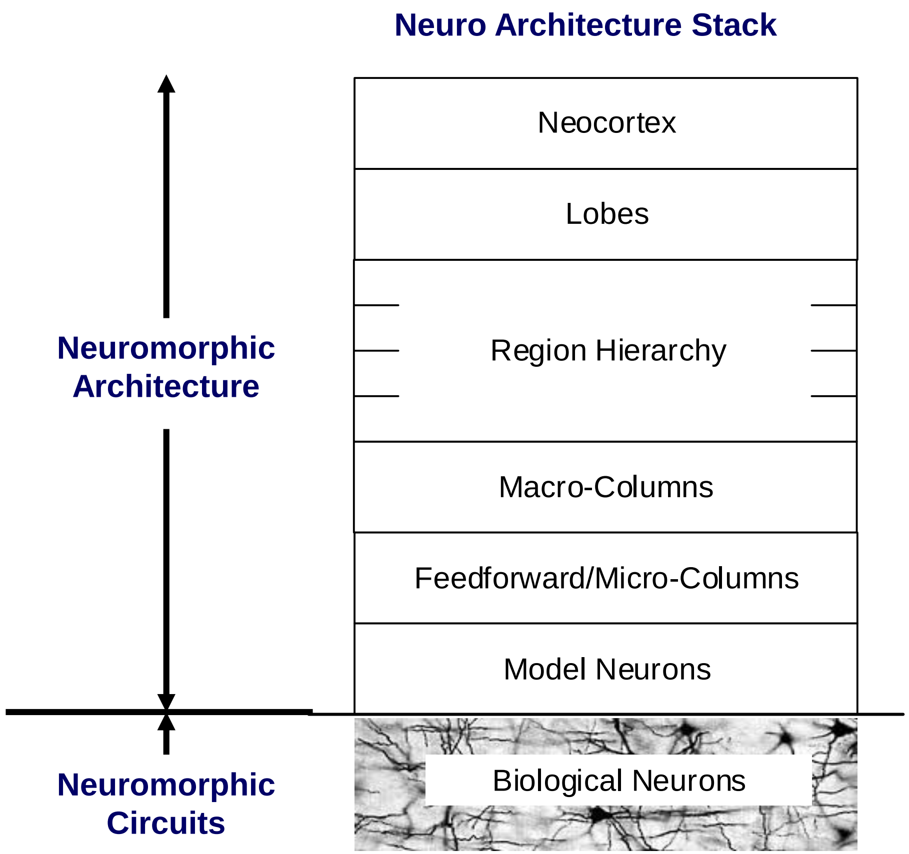
Neuro Architecture Stack

- Comprehending neocortical computing will require levels of abstraction
  - We (humans) can only comprehend assemblies of a certain limited complexity
    - So, we rely on abstraction
  - *Fortunately*, the physical hierarchy seems to match our ability to comprehend
    - Each functional block composed of 10 to 100 lower level blocks



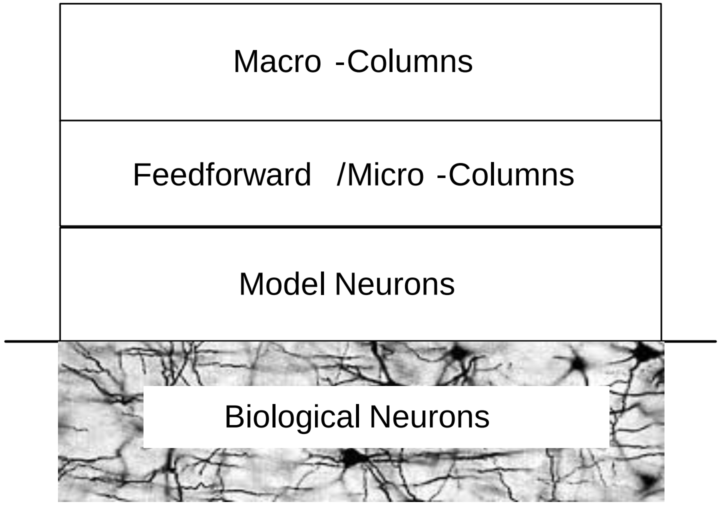
Long Term Roadmap

- Start at the bottom of the stack
  - With biological neurons
- Reverse-architect to the top
  - A *Neuromorphic Architecture* implements the computing paradigm(s) used in the neocortex
  - *Neuromorphic Circuits* are electrical circuits that function in ways similar to neurons and can be used to implement Neuromorphic Architectures.
  - *Neuromorphic Architectures* do not require *Neuromorphic Circuits*

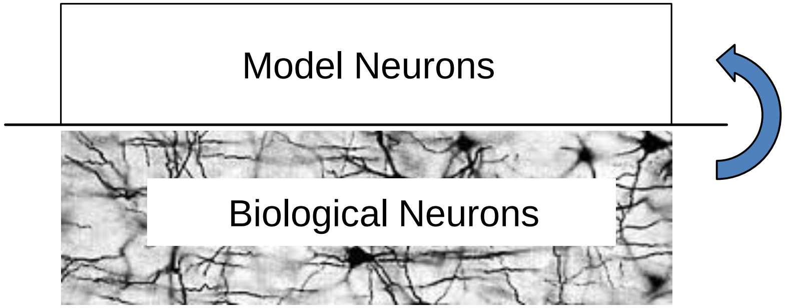


Near Term Roadmap

- First, focus on abstraction from biological neurons to computing elements
  - Consider *results* from *experimental* neuroscience
  - Consider *models* from *theoretical* neuroscience
  - Postulate a set of basic elements
- Next, develop *quasi-standard* building blocks (10-100 neurons)
  - Analogous to RTL blocks
  - Develop these blocks by constructing and experimenting with Temporal Neural Networks
- *First Major Milestone:* Deep TNNs
  - Described earlier
- Three layers of abstraction are simultaneously in play:
  - Model neurons
  - Column-level quasi-standard assemblies
  - Macro-Columns

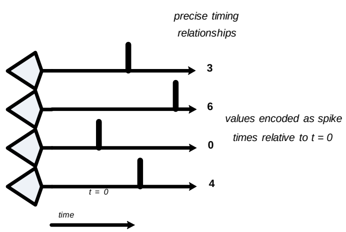


Primitive Abstraction: Biological to Computational

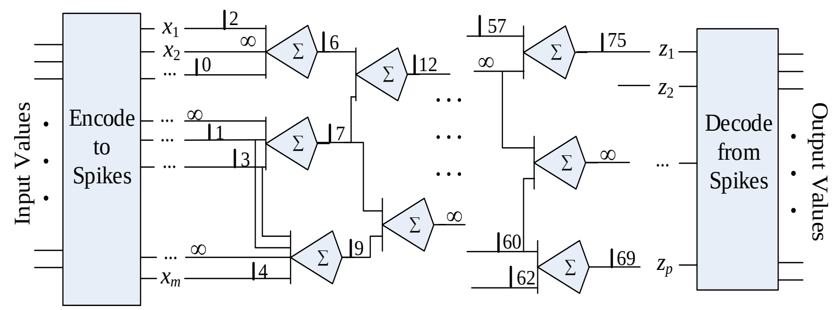


Basic Architectural Elements

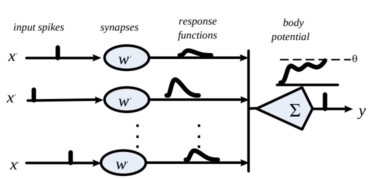
Temporal coding



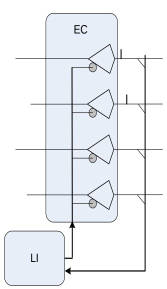
Temporal Neural Networks



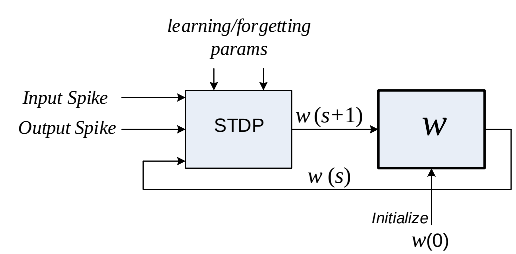
Excitatory Neurons



Bulk Inhibition

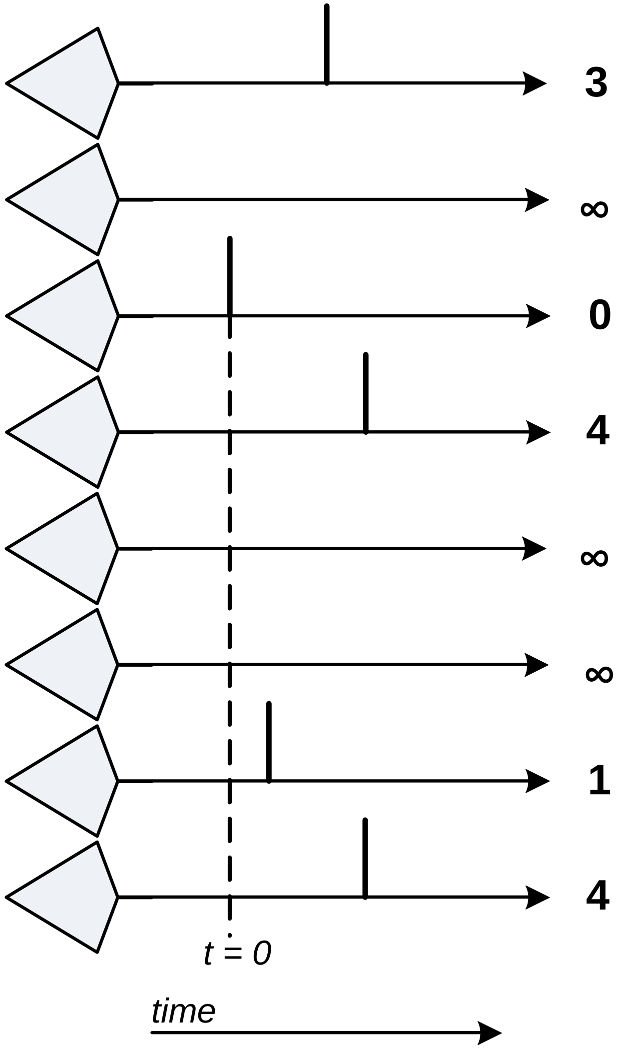


STDP



Temporal Coding

- Information is communicated via transient events
  - e.g., voltage spikes
  - Hereafter "spike" is shorthand for "transient temporal event"
- Values are encoded via spike timing relationships across parallel communication lines
  - Based on spike times relative to first (t = 0)
  - Low resolution: 1-in-8, say
  - Example is not a "toy" – values are realistic



Note: in practice, coding is sparser than in this example

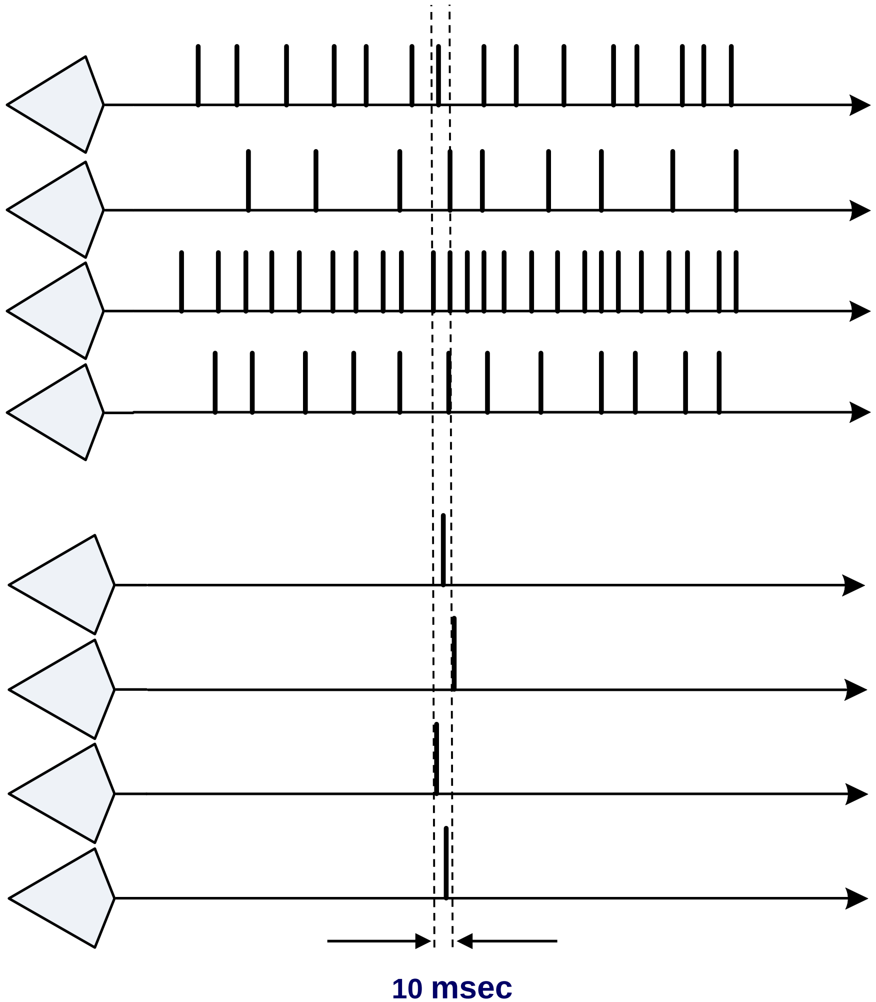
The Temporal Resource

The *flow of time* can be used effectively as a *communication* and *computation resource.*

- The *flow of time* has some ultimate engineering advantages
  - It requires no space
  - It consumes no energy
  - It is free – time flows whether we want it to or not
- Yet, we (humans) try to eliminate the effects of time when constructing computer systems
  - Synchronizing clocks & delay-independent asynchronous circuits
  - *This may be the best choice for conventional computing problems and technologies*
- How about natural evolution?
  - Tackles completely different set of computing problems
  - With a completely different technology

Compare with Rate Coding

- Plot spikes on same biological time scale
- Both methods convey similar information
- Temporal method is
  - An order of magnitude faster
  - An order of magnitude more efficient (#spikes)

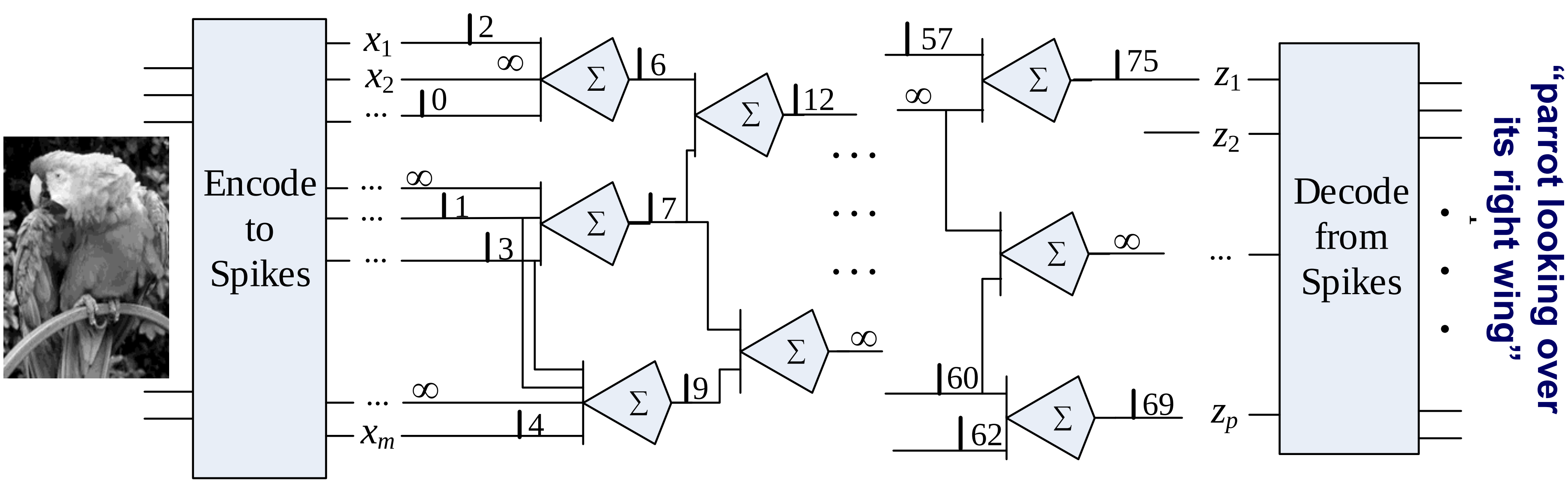


The temporal coding method has significant, broad experimental support

- The rate method does not.

Temporal Neural Network

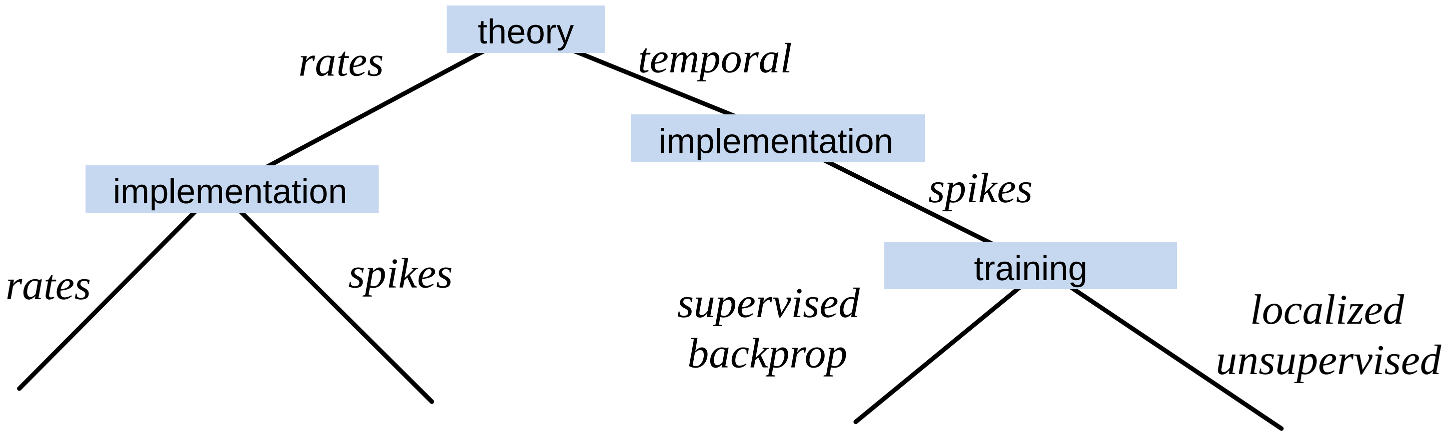
- A feedforward network of model neurons
  - Values communicated via temporal codes (*implemented as "spikes"*)
  - Feedforward flow (*without loss of computational generality*)
  - Computation: a wave of spikes passes from inputs to outputs
  - At most one spike per line per computation



Neural Network Taxonomy

- **Primary goal: *a computing paradigm* that learns in an unsupervised, continual, fast, and energy efficient way**
  - Separates this research from vast majority of “Spiking Neural Network” (SNN) research

Neural Networks



classic ANNs → deep CNNs
-classification
-supervised
-compute-intensive training

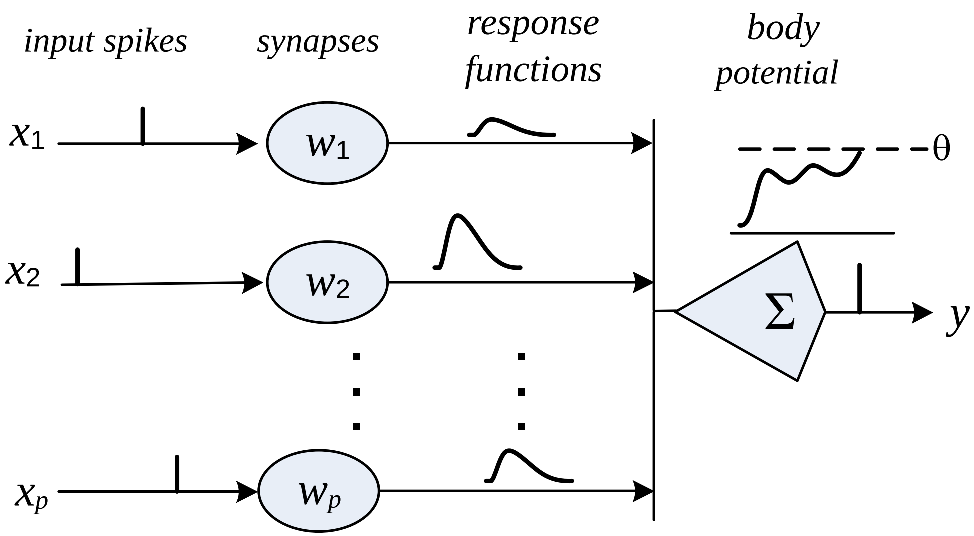
“SNNs”
-classification
-supervised
-compute-intensive training
-accuracy is a struggle
-energy efficiency in doubt

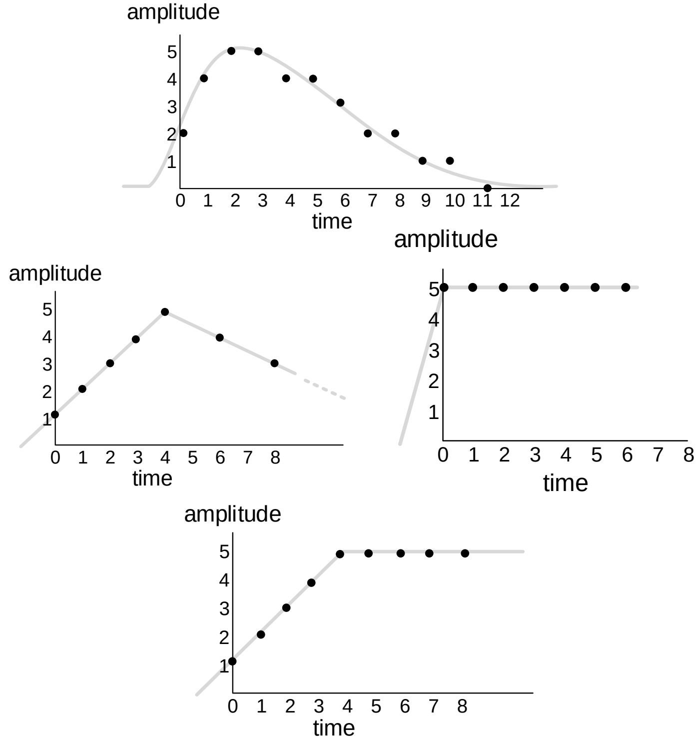
“SNNs”
-classification
-supervised
-compute-intensive training
-accuracy is a struggle
-probable *energy efficiency benefits*

“SNNs” (TNNs)
-clustering
-unsupervised
--simple, dynamic training
-accuracy-neutral(?)
-probable *energy efficiency benefits*

Excitatory Neuron Model (*repeat*)

- Basic Spike Response Model (SRM0) -- Kistler, Gerstner, & van Hemmen (1997)

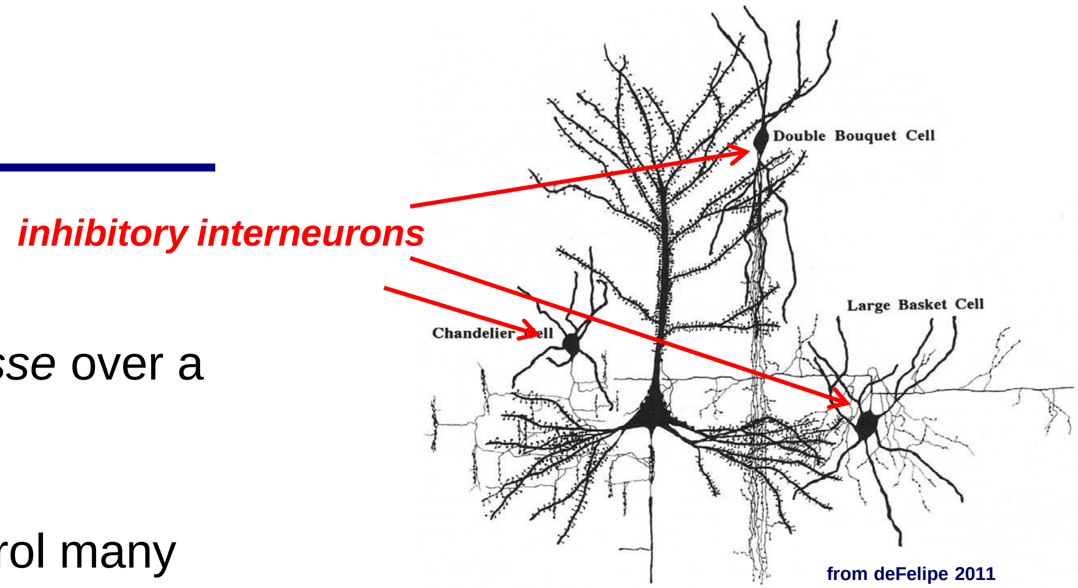


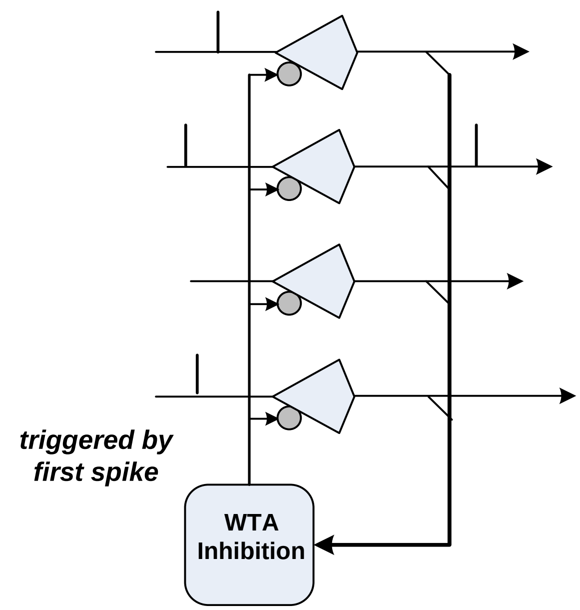


1) A volley of spikes is applied at inputs
2) At each input's synapse, the spike produces a weighted response function
3) Responses are summed linearly at neuron body
4) An output spike is emitted if/when potential exceeds threshold value (θ)

Bulk Inhibition

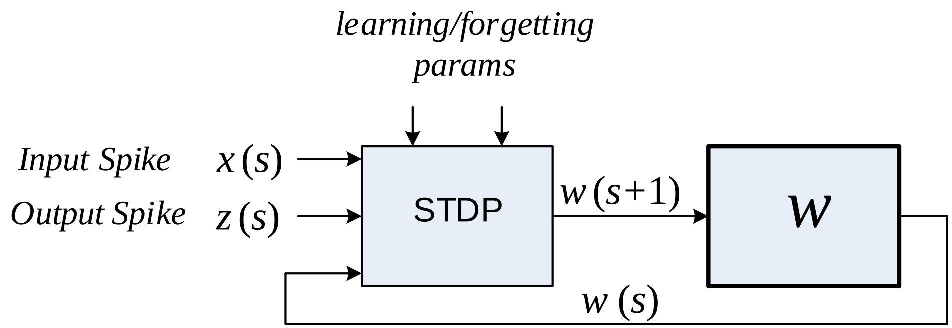
- Inhibitory neurons act *en masse* over a local volume of neurons
  - A "blanket" of inhibition
- A few inhibitory neurons control many excitatory neurons
  - Up to 30 synapses per target excitatory neuron (avg. = 15)
  - Some connections directly to excitatory body and axon
- Model as parameterized Winner-Take-All (WTA) inhibition
- Note: this mechanism is probably built into a soft synchronization method based on inhibitory oscillations

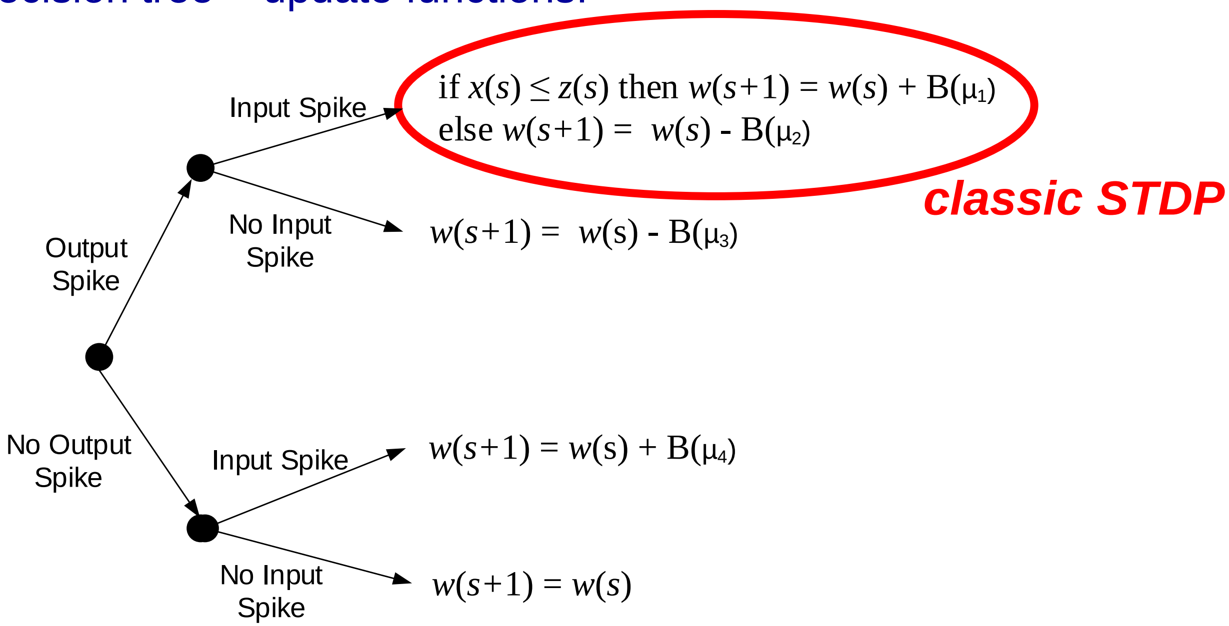




STDP

- Spike Timing Dependent Plasticity – where the magic is
  - Each synapse updates weight based on current weight and *local* spike time relationships
  - Implemented as a small finite state machine
  - Many methods under study
  - Decision tree + update functions:





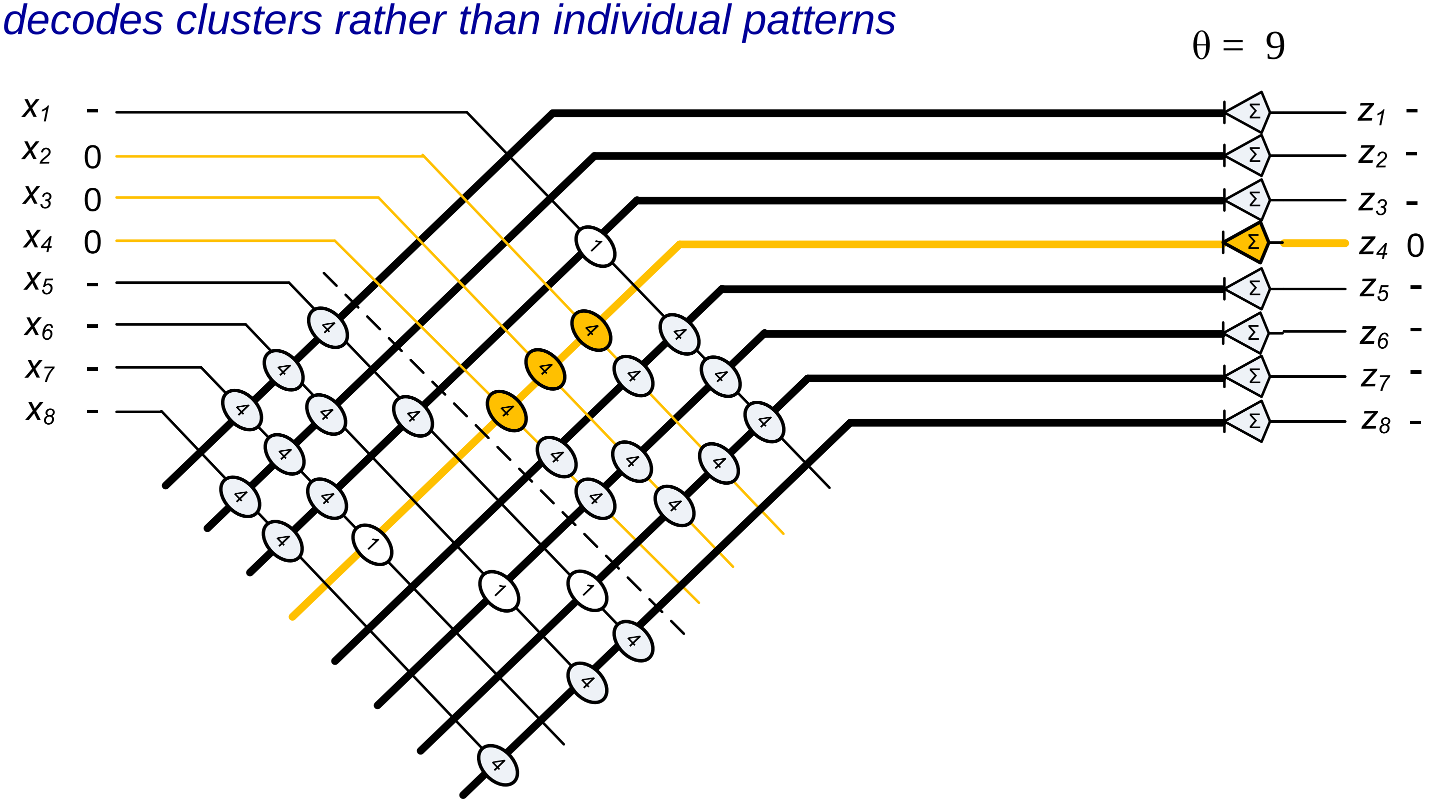
Example: Decode Matrix

- STDP establishes weights in a way that decodes the most frequent input patterns
  - Relies on bimodal synaptic weight distribution (0 or W_{max})
  - Timing of output spikes depends on response function

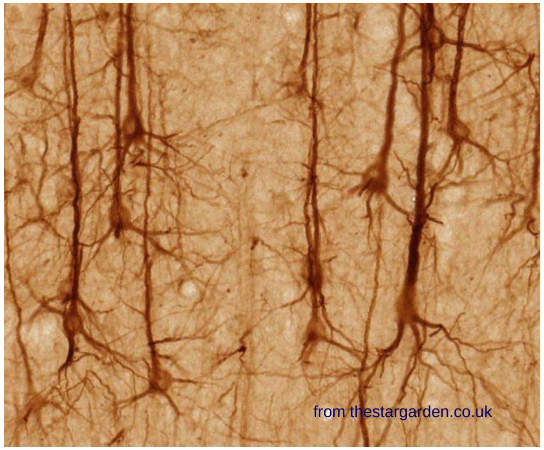
    Step no-leak in this example
  - *In general decodes clusters rather than individual patterns*

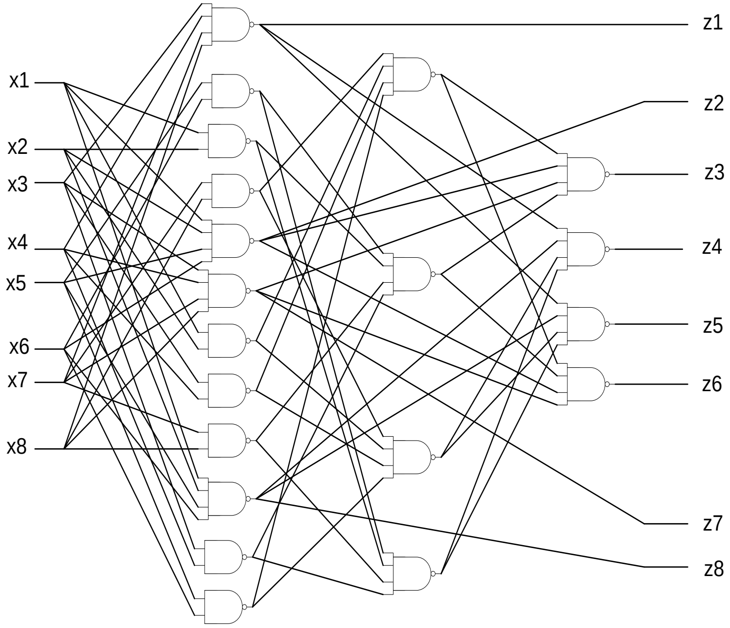
0* : *spike @ t = 0

– : = no spike

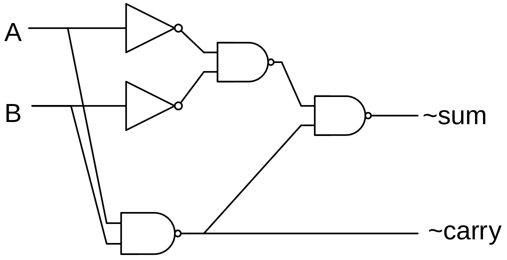


How can the computing model be simple?





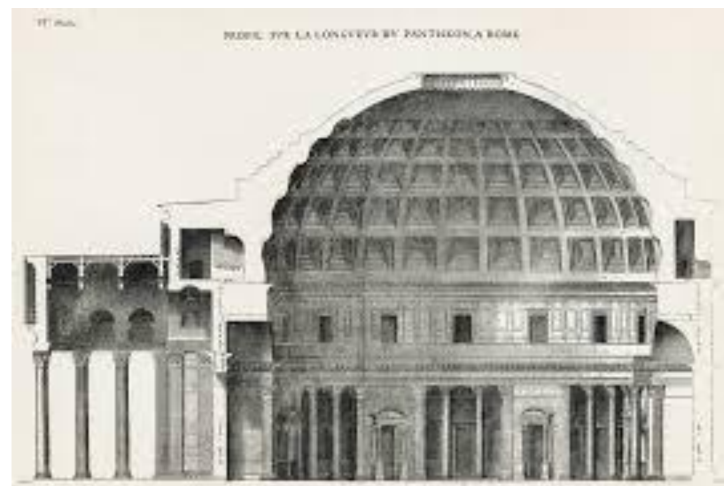
- *In the neocortex,* computation is inextricably combined with obfuscating infrastructure
- *In the computer architecture "lab",* we can consider the computing paradigm absent all the complications



A Pantheon of Neuroscience Architects

- Theoretical neuroscientists have been developing brain-based computing paradigms for over two decades
  - Lots of good ideas have been put forward
  - Computer architects don't start from scratch

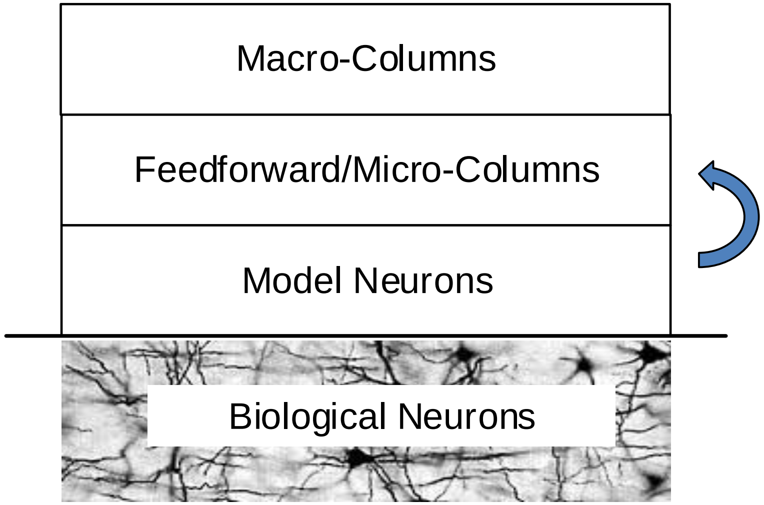
Simon Thorpe Damien Querlioz Rudy Guyonneau Rufin VanRullen Timothée Masquelier	*temporal coding,* *STDP,* *TNN architectures*
Wolfgang Maass	*TNN (SNN) theory*
Henry Markram	*STDP*
Wulfram Gerstner	*Neuron Models, STDP*
Sander Bohte	*TNN architecture*
Wolfgang Singer Pascal Fries	*Inhibitory oscillation;* *soft synchronization*



Column Level Abstraction: “RTL”

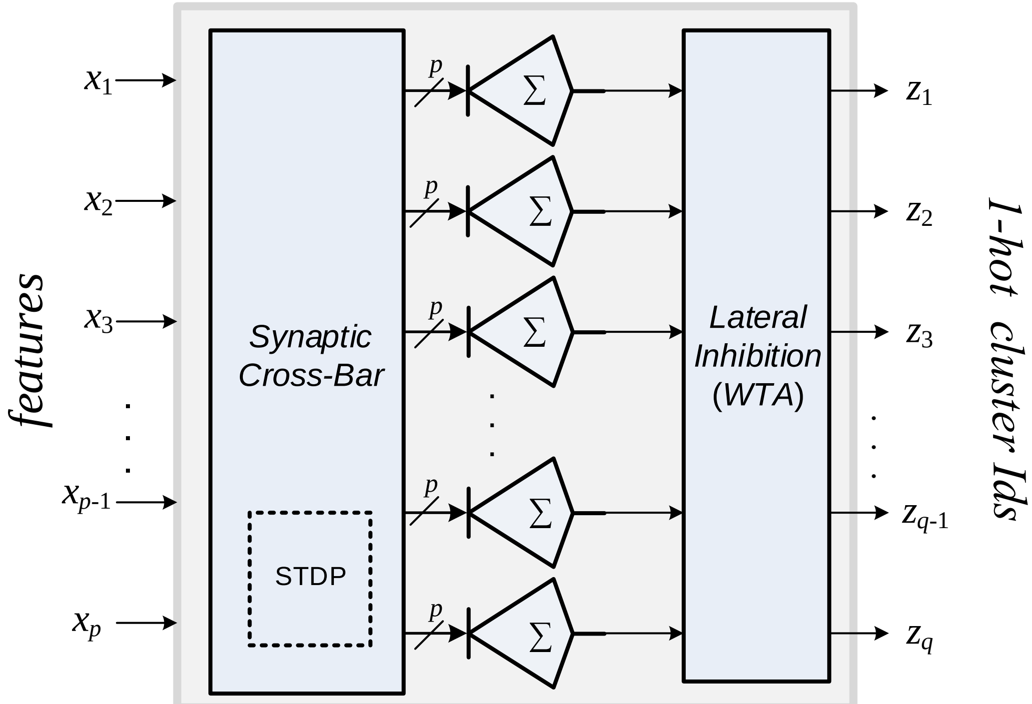
Column Level Abstraction

- Combine primitives into higher level computing assemblies
  - Analogous to Register Transfer Level (RTL) in digital logic
  - Design will probably be done at this level

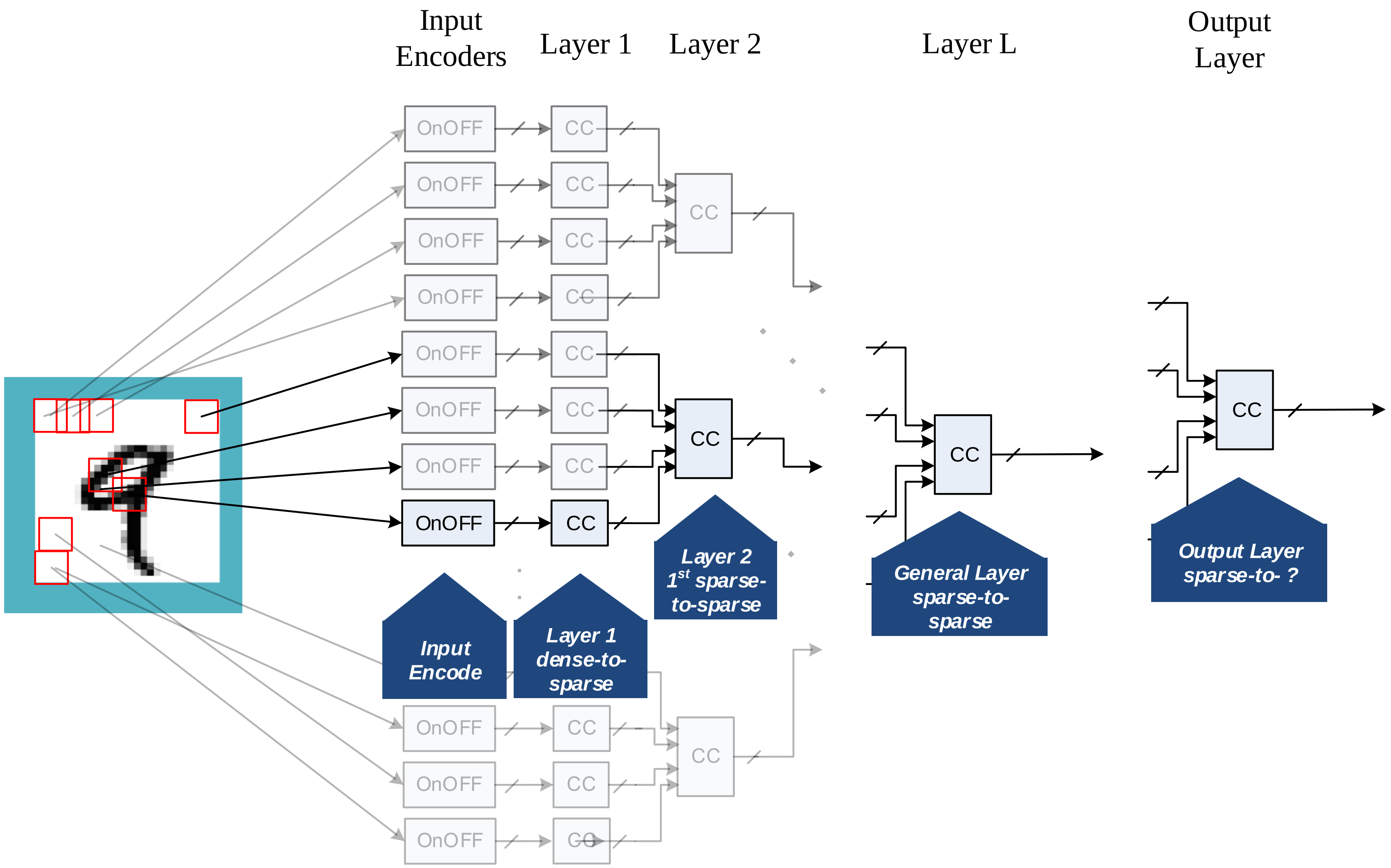


Computational Column (CC)

- Basic TNN building block
- Learns and maps inputs having similar features to the same Cluster Id
- Input lines may be interpreted as features
  - The presence of a spike indicates the presence of the feature
  - The timing of a spike indicates the relative strength of the feature
- A CId is a 1-hot temporal coding
  - The better the cluster “match”, the earlier the spike
  - CIds become features for the next network Layer

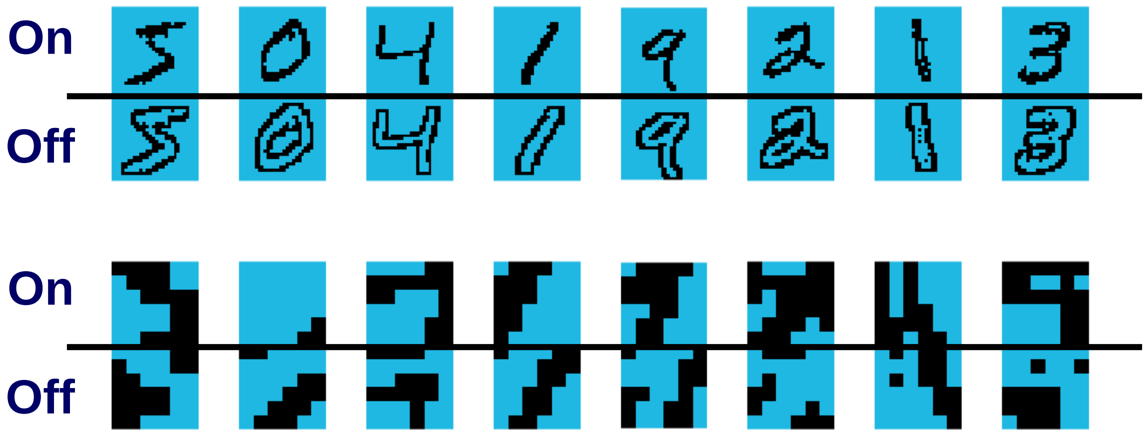


TNN Roadmap Waypoints



Waypoint 0: Input Encoding

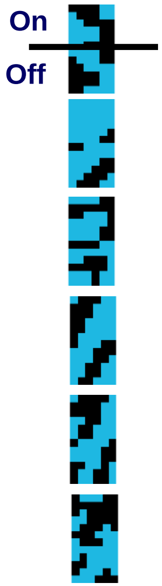
- Leverage biology
- Example: OnOff retinal ganglion cells
  - Perform edge detection
- Encode spikes according to contrast between center and surround
  - Most intense contrast yields earlier spikes
- *However,* binarize primary input to simplify initial experiments
  - Separates Layer 1 temporal *computation* from temporal *communication*

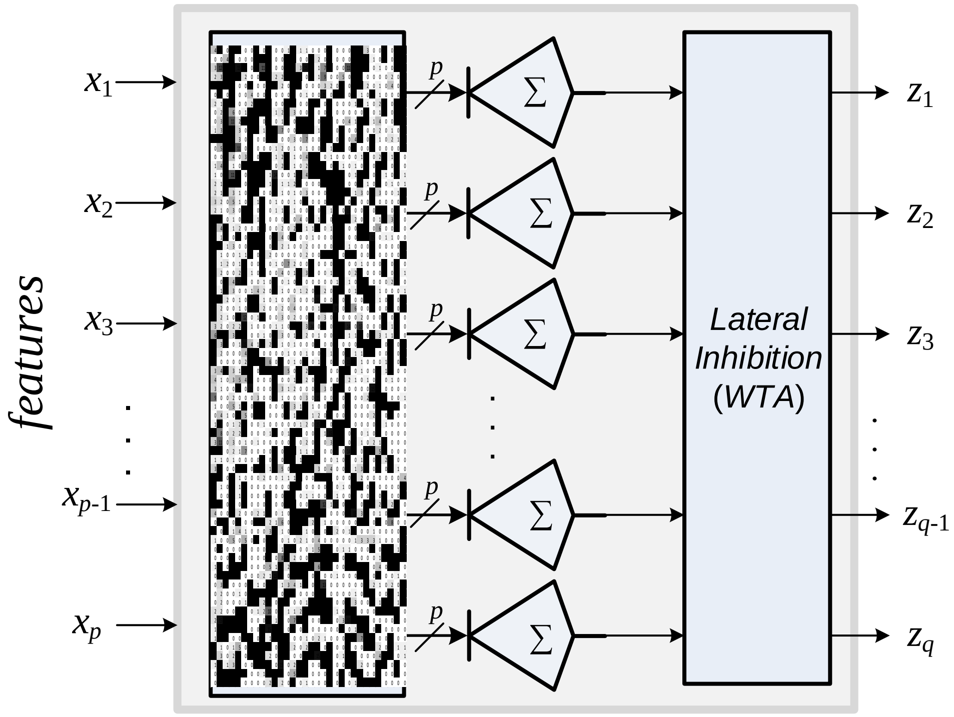


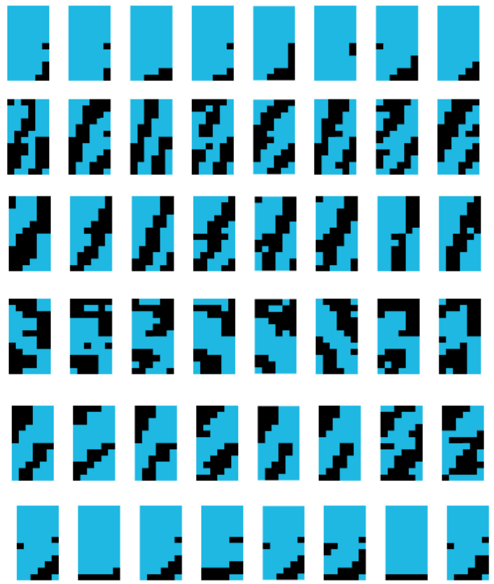
6x6 RF at [14,14]

Waypoint 1: Dense-to-Sparse CC

- Unsupervised clustering
  - Example 6x6 RFs from MNIST – OnOff encoded, *binarized*
  - State-of-the-art: **Kheradpisheh, et al. "STDP-based spiking deep neural networks for object recognition." *Neural Networks* 99 (2018): 56-67.**



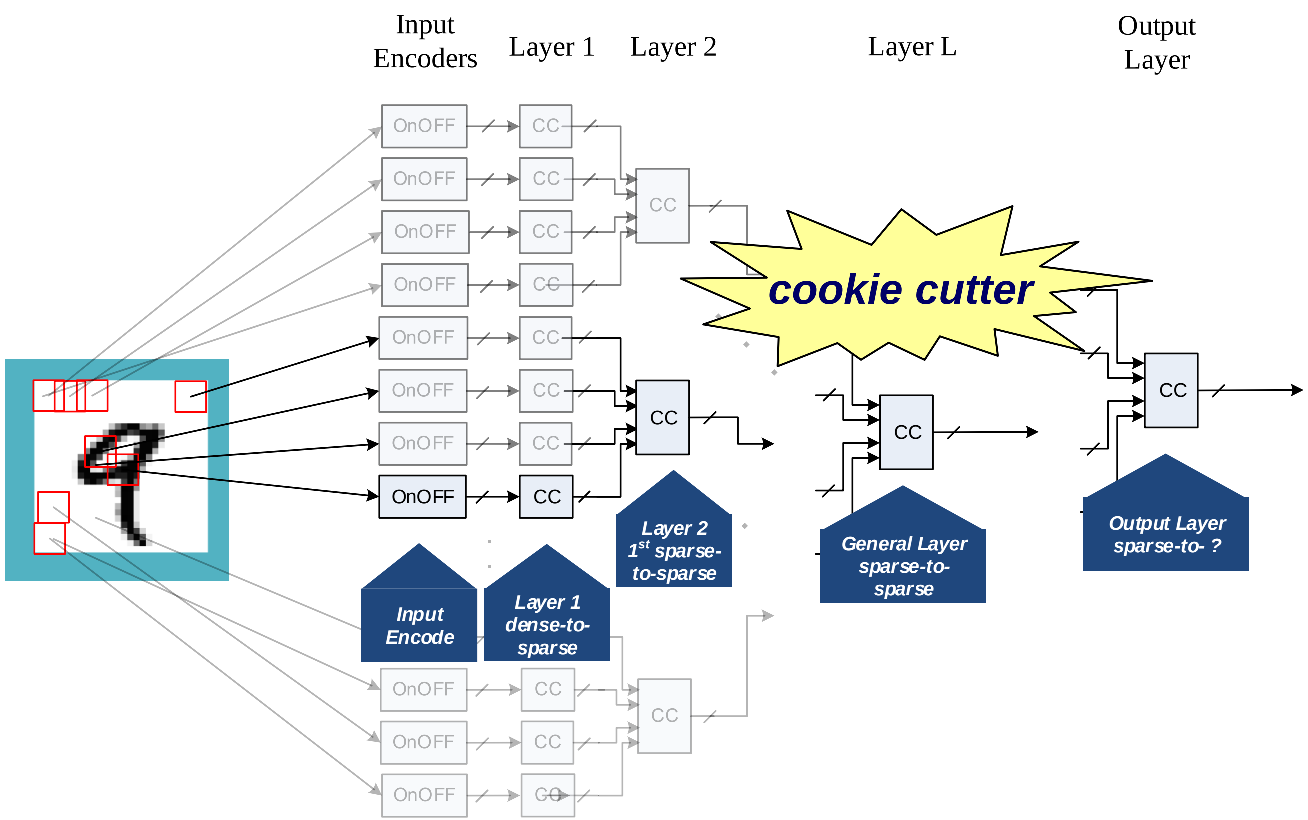




STDP Works.

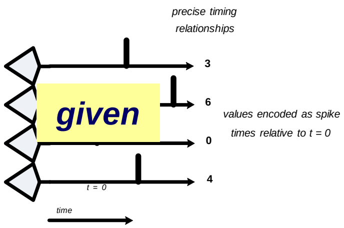
Waypoints 2 & 3: Sparse-to-Sparse CCs

- The goal is a “cookie cutter” CC
  - To allow construction of arbitrarily wide, arbitrarily deep TNNs
  - *No one has been successful to date – Wide-open research area*

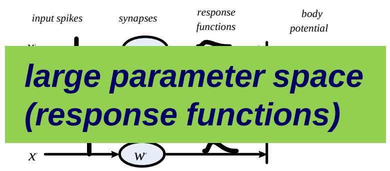


Research Space

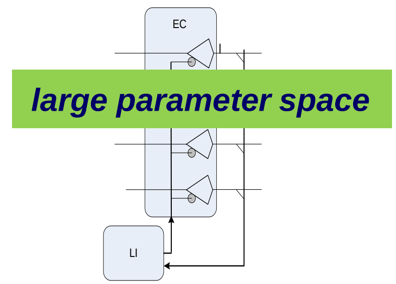
Temporal coding
efficient coding based on temporal relationships



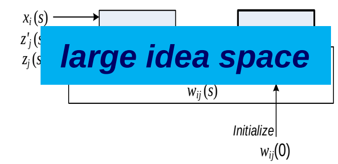
Excitatory Neurons
consistent with the rules of Newtonian time



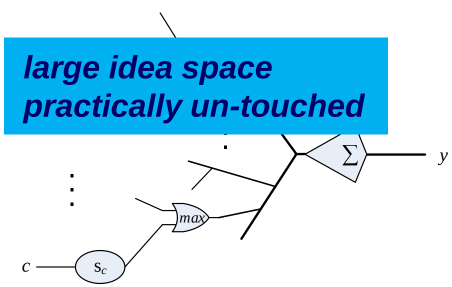
Inhibition Blocks
consistent with the rules of Newtonian time



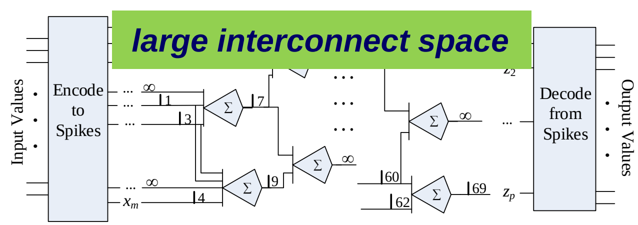
STDP
localized, unsupervised learning



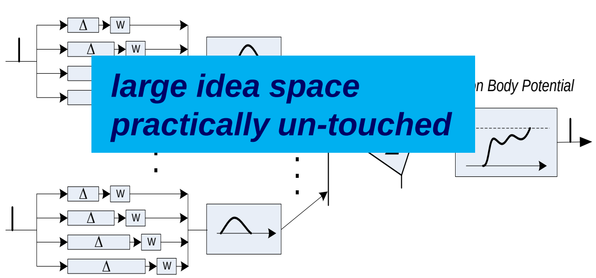
Dendritic Computation
largely unexplored



Temporal Neural Networks
Computation proceeds as a wave of spikes passes from inputs to outputs



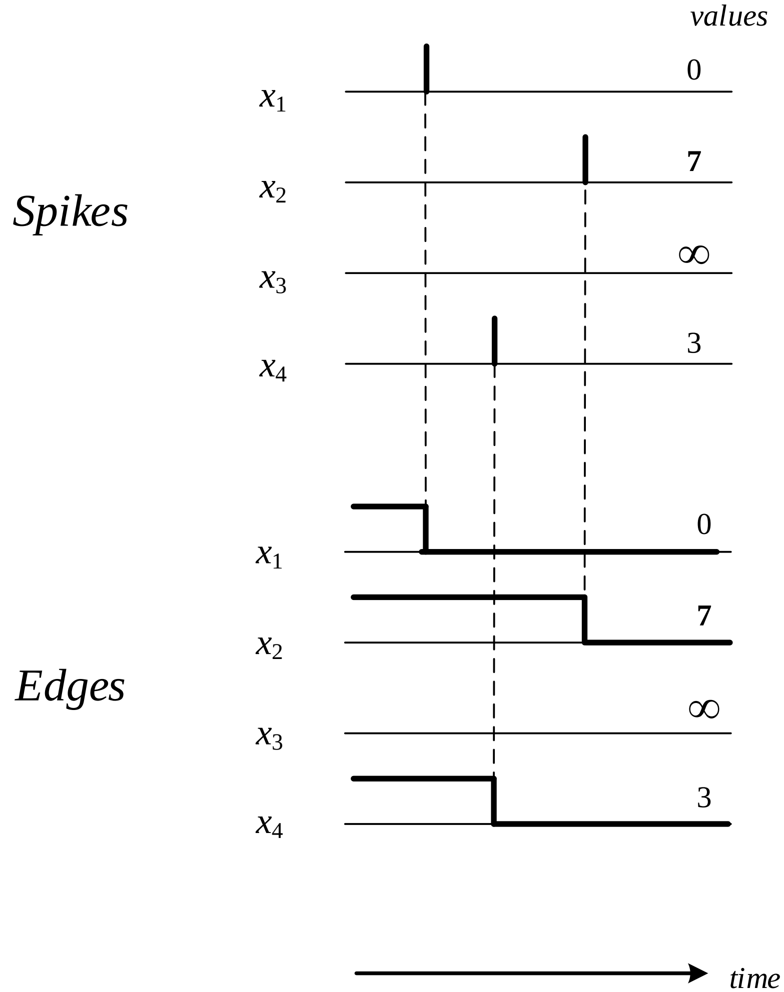
Compound Synapses
biologically correct; largely unexplored



Race Logic*

- *Spikes* are not the only way to encode values as the times of transient temporal events
- *Edges* work, too.
  - Signal via 1 → 0 transitions
- Efficiencies remain intact
- Edges + race logic yields direct off-the-shelf CMOS implementation
- An alternative to neuromorphic circuits

see 2018 ISCA paper



**Race logic: Madhavan, Sherwood, Strukov, UC-Santa Barbara*

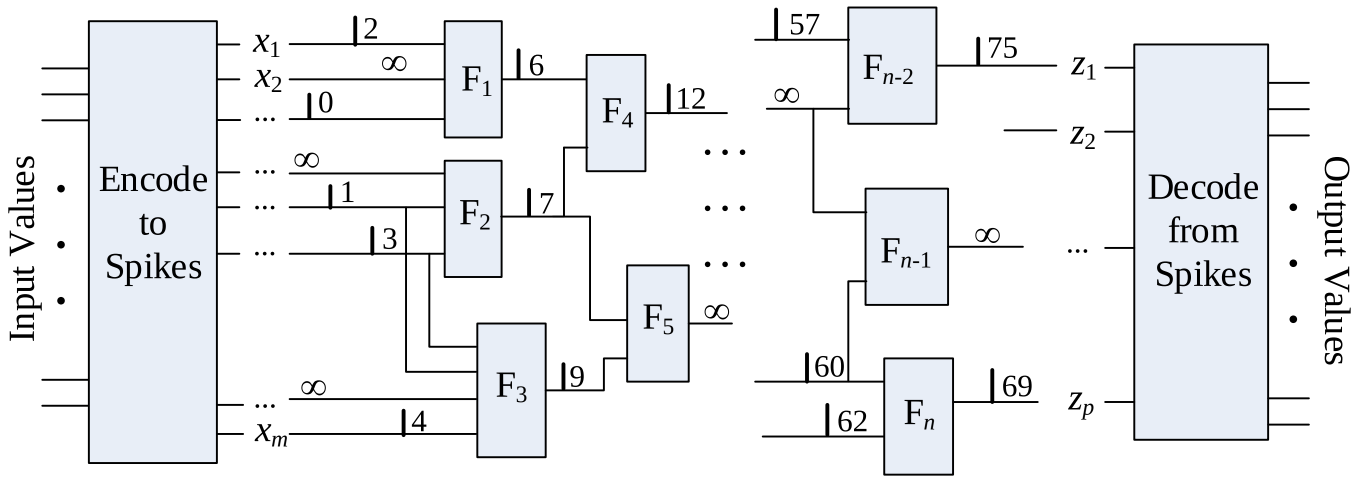
Mathematical Underpinnings

Contrasting Mathematical Approaches

- Neuroscience approach
  - Real arithmetic – differential equations
  - Supports unbounded computational resolution
  - Discretization done implicitly through conversion to floating point
- Computer Architecture approach
  - Simple mathematics (Boolean algebra)
  - Inherently discrete
- A Computer Architecture approach to modeling neural operation
  - The devices being modeled are naturally very low resolution (1-in-8)
  - Use discrete math and small integers to implement temporal functions

low resolution, unary computation

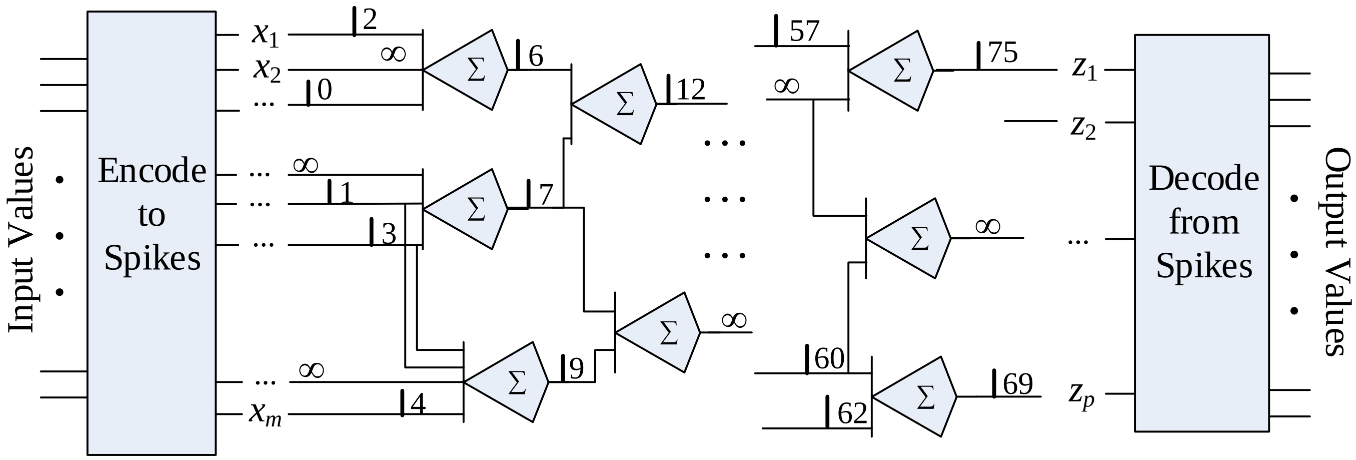
Space-Time Computing Network



A *Space-Time Computing Network* is a feedforward composition of functions, F_i, where:

1) Each F_i has a ***finite state implementation***
2) Each F_i is ***causal***
   The output spike time is independent of later input spike times
   No spontaneous output spikes
3) Each F_i is ***invariant***
   If all the input spikes are delayed by some constant amount then the output spike is delayed by the same constant amount

Space-Time Computing Network



A *Space-Time Computing Network* is a feedforward composition of functions, F_i, where:

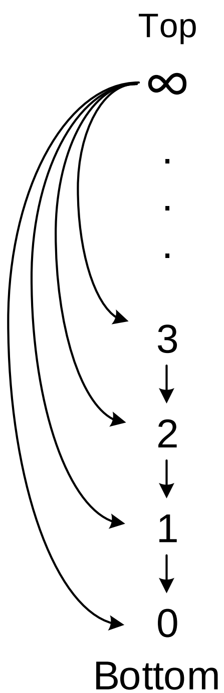
1) Each F_i has a ***finite state implementation***
2) Each F_i is ***causal***
   The output spike time is independent of later input spike times
   No spontaneous output spikes
3) Each F_i is ***invariant***
   If all the input spikes are delayed by some constant amount then the output spike is delayed by the same constant amount

TNNs are an important special case

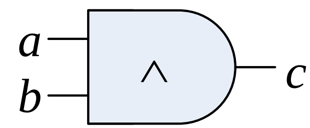
(Newtonian) Space-Time Algebra

Bounded Distributive Lattice

- 0, 1, 2,…, ∞
- Interpretation: points in time
- *not complemented*

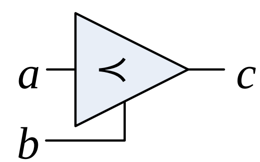


Primitive Operators



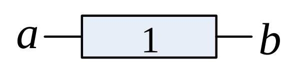
min: if $a < b$ then $c = a$
else $c = b$

"atomic excitation"



lt: if $a < b$ then $c = a$
else $c = \infty$

"atomic inhibition"

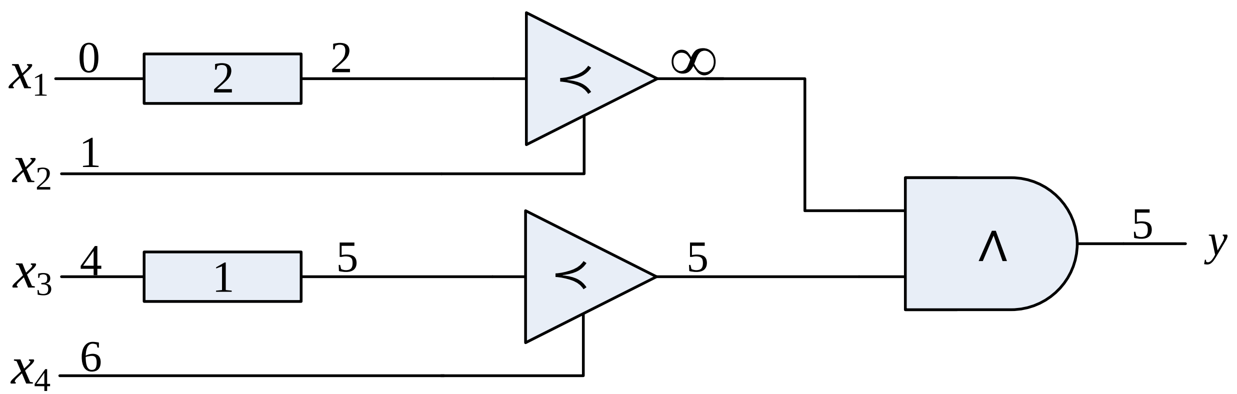


inc: $b = a + 1$

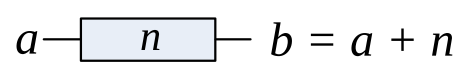
"atomic delay"

Space-Time Networks

- *Theorem*: Any feedforward composition of *s-t* functions is an *s-t* function
  - ⇒ Build networks by composing *s-t* primitives
  - Example:



note: shorthand for n increments in series:



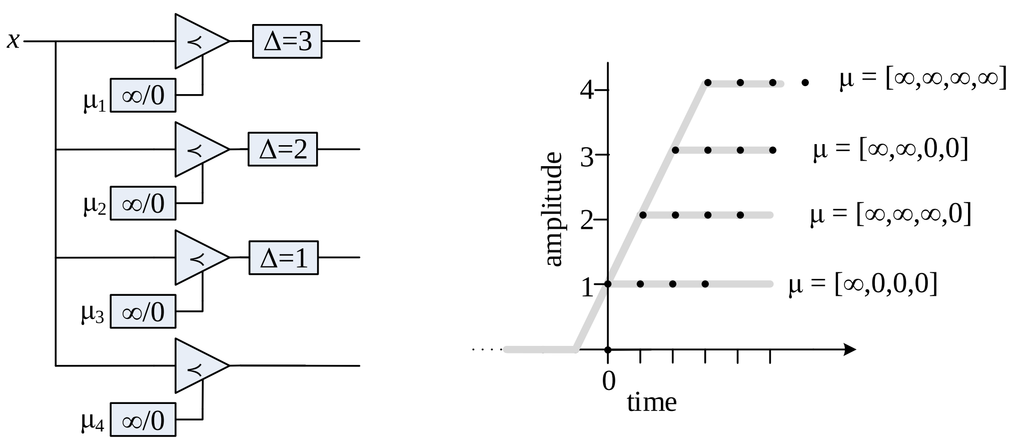
Elementary Functions

- Table of all two-input *s-t* functions
  - All implementable with the three primitives

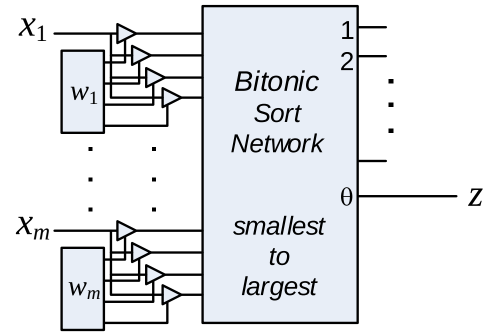
function	name	symbol
if $a < b$ then a ; else b	*min*	$\wedge$
if $a \leq b$ then a ; else ∞	*less or equal*	$\preccurlyeq$
if $a \neq b$ then a ; else ∞	*not equal*	$\neq$
if $a < b$ then a else if $b < a$ then b ; else ∞	*exclusive min*	x$\wedge$
if $a < b$ then a ; else ∞	*less than*	$\prec$
if $a \geq b$ then a ; else b	*max*	$\vee$
if $a > b$ then a else if $b > a$ then b ; else ∞	*exclusive max*	x$\vee$
if $a \geq b$ then a ; else ∞	*greater or equal*	$\succcurlyeq$
if $a = b$ then a ; else ∞	*equal*	$\equiv$
if $a > b$ then a ; else ∞	*greater than*	$\succ$

TNN Primitives Implemented as ST Functions

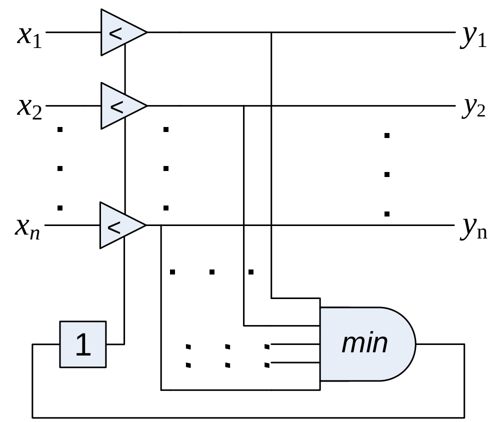
(*sort* is a space-time function)



Response function generator



SRM0 Neuron



WTA Inhibition

The Box: The way we (humans) think about computation

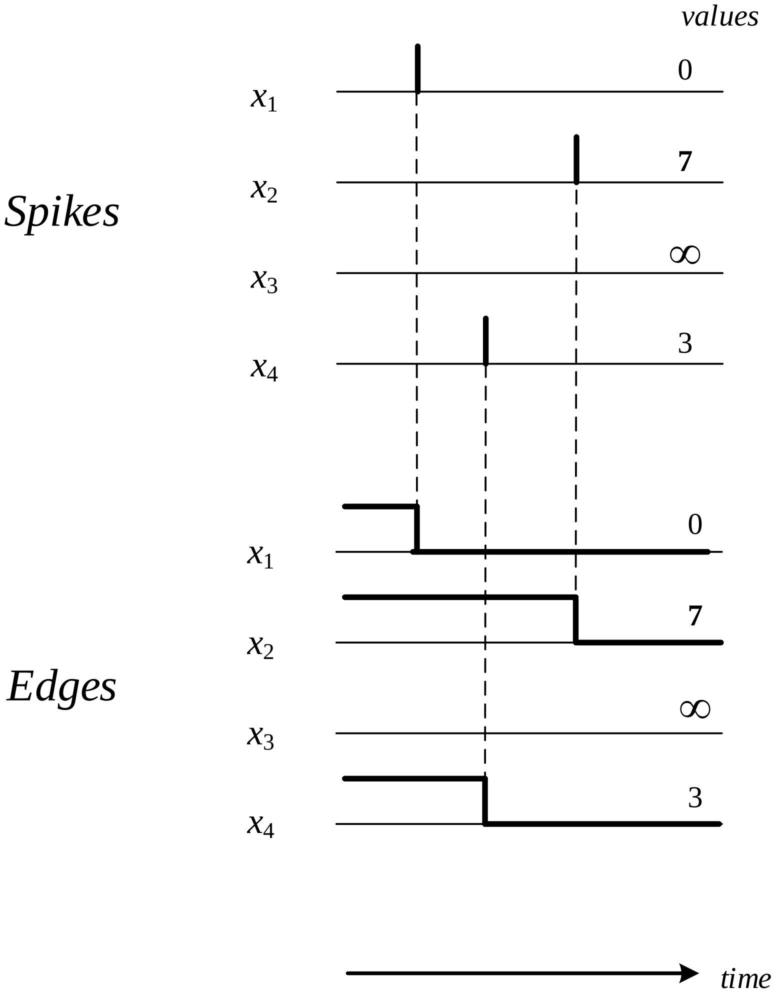
- We try to eliminate temporal effects when implementing functions
  - TNNs uses the uniform flow of time as a key resource
- We use *add* and *mult* as primitives for almost all mathematical models
  - Neither *add* nor *mult* (except add of a constant) is an *s-t* function
- We prefer high resolution (precision) data representations
  - *Unary computing* practical only for very low-res direct implementations
- We strive for complete functional completeness
  - *s-t* primitives complete *only* for *s-t* functions
  - There is no inversion, complementation, or negation

Digital CMOS Implementation

Race Logic*

- *Spikes* are not the only way to encode values as the times of transient temporal events
- *Edges* work, too.
    - Signal via 1 → 0 transitions
- Efficiencies remain intact
- Combined with race logic yields direct off-the-shelf CMOS implementation

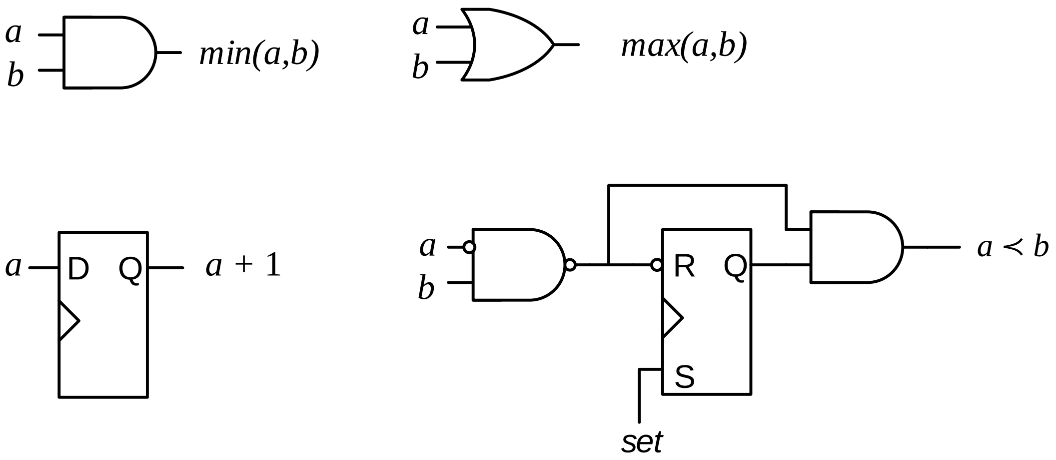
see 2018 ISCA paper



**Race logic: Madhavan, Sherwood, Strukov, UC-Santa Barbara*

Generalized Race Logic

- *S-T* primitives implemented directly with conventional digital circuits
  - Signal via 1 → 0 transitions

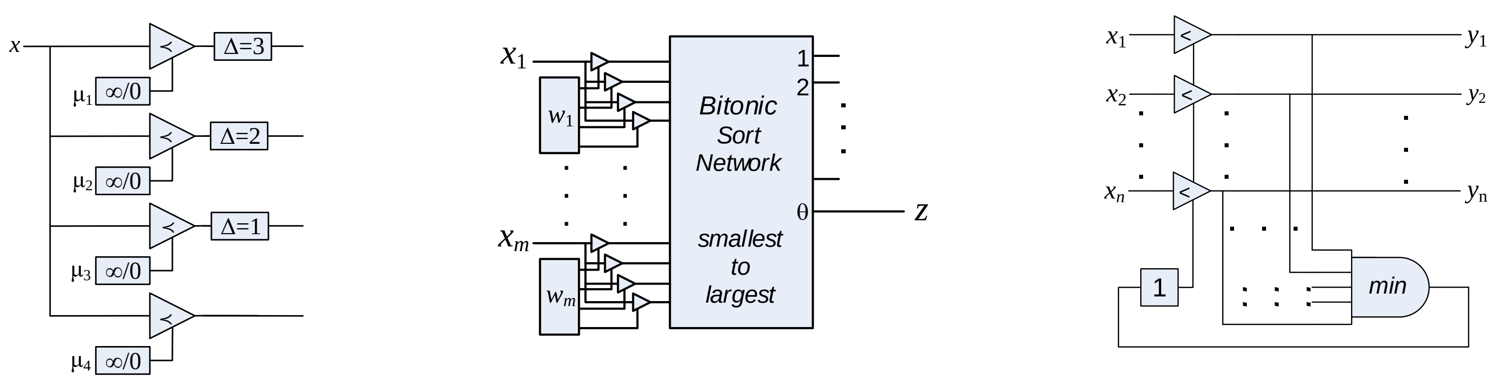


⇒ We can implement SRM0 neurons and WTA inhibition with off-the-shelf CMOS

⇒Very fast and efficient TNNs

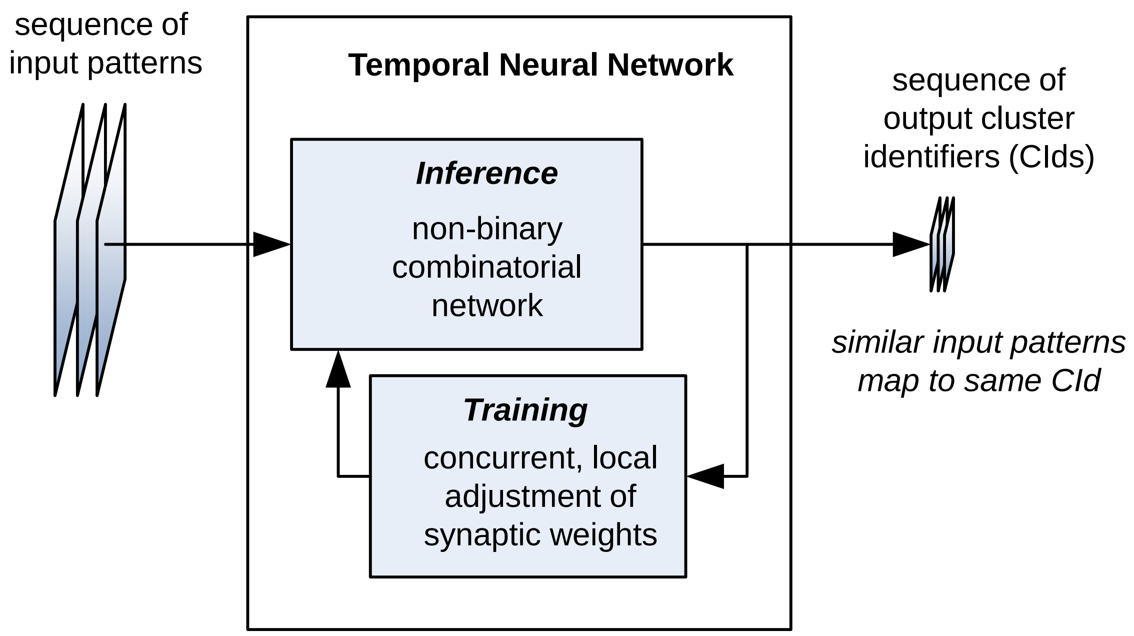
TNN Primitives Implemented with CMOS Gates

- Signal via edges w/ off-the-shelf CMOS
  - minimize static power
  - lots of wires
  - signaling and functional operation very sparse
- A *direct* implementation
  - An alternative to analog spiking neuromorphic circuits



Put It All Together: 1st Major Milestone

- TNN with unsupervised, continual learning via STDP
- Describable w/ a temporal algebra
  - Supports low resolution, discrete computation
- Hardware implementation
  - Implementable with digital CMOS
  - Fast
  - Energy efficient



Closing Remarks

The Barrier to Entry is Low

- The TNN literature is relatively small
  - TNN development is not very far along
  - So there isn't a lot of stuff to learn
- Low computational requirements
  - A high-end desktop computer running parallel threads is adequate
- It is possible to be up to speed in a few months (at most)
  - Writing a simulator is a good way to start

Are We at a Tipping Point?

- Experimental neuroscience spans more than 100 years
  - The published literature is vast and continues to grow at a fast rate
- What if all experimental neuroscience research were to cease tomorrow?
  - Is enough already known to allow reverse-architecting the neocortex?
- This would a *tipping point* for computer architecture research
  - *No more experimental data is needed*
  - We may already be there, or are fast approaching
- At the tipping point:
  - Sufficient first-order effects are known
  - It's only a matter of combining them in a coherent and effective way

Bibliography

J. E. Smith. "Space-Time Computing with Temporal Neural Networks." *Synthesis Lectures on Computer Architecture* 12, no. 2 (2017) -- *be sure to read 2019 preface*

J. E. Smith. "Space-time algebra: a model for neocortical computation." In *Proceedings of the 45th Annual International Symposium on Computer Architecture*, pp. 289-300. IEEE Press, 2018.

Temporal Coding

Hopfield, J. J. "Pattern recognition computation using action potential timing for stimulus representation." *NATURE* 376 (1995): 33.

Excitatory Neurons

Gerstner, Wulfram, and J. Leo Van Hemmen. "How to describe neuronal activity: spikes, rates, or assemblies?." In *Advances in neural information processing systems*, pp. 463-470. 1994.

STDP

Gerstner, Wulfram, Richard Kempter, J. Leo van Hemmen, and Hermann Wagner. "A neuronal learning rule for sub-millisecond temporal coding." Nature 383, no. 6595 (1996): 76-78.

Markram, Henry, Joachim Lübke, Michael Frotscher, and Bert Sakmann. "Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs." *Science* 275, no. 5297 (1997): 213-215.

Guyonneau, Rudy, Rufin Vanrullen, and Simon J. Thorpe. "Neurons tune to the earliest spikes through STDP." *Neural Computation* 17, no. 4 (2005): 859-879.

TNNs

Maass, Wolfgang, Networks of spiking neurons: the third generation of neural network models, *Neural networks* 10.9 (1997): 1659-1671.

Kheradpisheh, et al. "STDP-based spiking deep neural networks for object recognition." *Neural Networks* 99 (2018): 56-67.

Oscillatory Behavior (Network Synchronization)

Fries, Pascal, Danko Nikolić, and Wolf Singer. "The gamma cycle." *Trends in neurosciences* 30, no. 7 (2007): 309-316.

Acknowledgements

Raquel Smith

Mario Nemirovsky, Cristobal Camarero, Ravi Nair, Joel Emer, Abhishek Bhattacharjee

Mikko Lipasti, Mark Hill, Margaret Martonosi, Michael Morgan

John Shen, Harideep Nair, Amy Zhang

Tim Sherwood, George Tzimpragos, Advait Madhavan

Shlomo Weiss, Ido Guy

