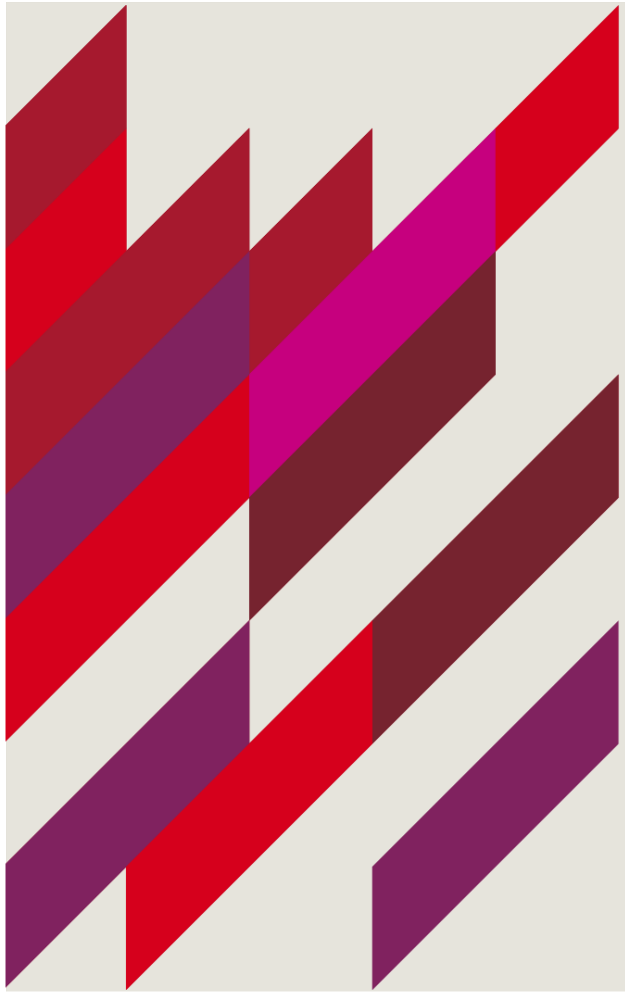
Stochastic Computing Systems

Tara Julia Hamilton

11 October 2018

Stochastic Computing Systems

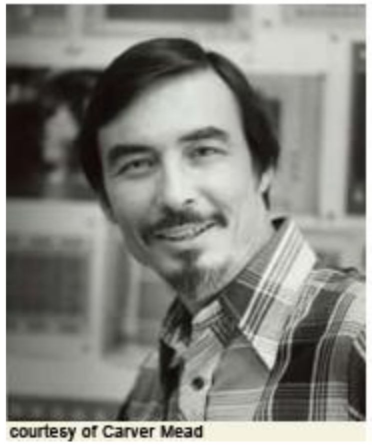


1. Neuromorphic Engineering
   a) Early Work
   b) My Work
   c) The Future…
2. Stochastic Electronics
   a) Some interesting (I hope!) algorithms
3. Stochastic Computation
   a) It's old but it's cool

Neuromorphic Engineering

THE EARLY YEARS

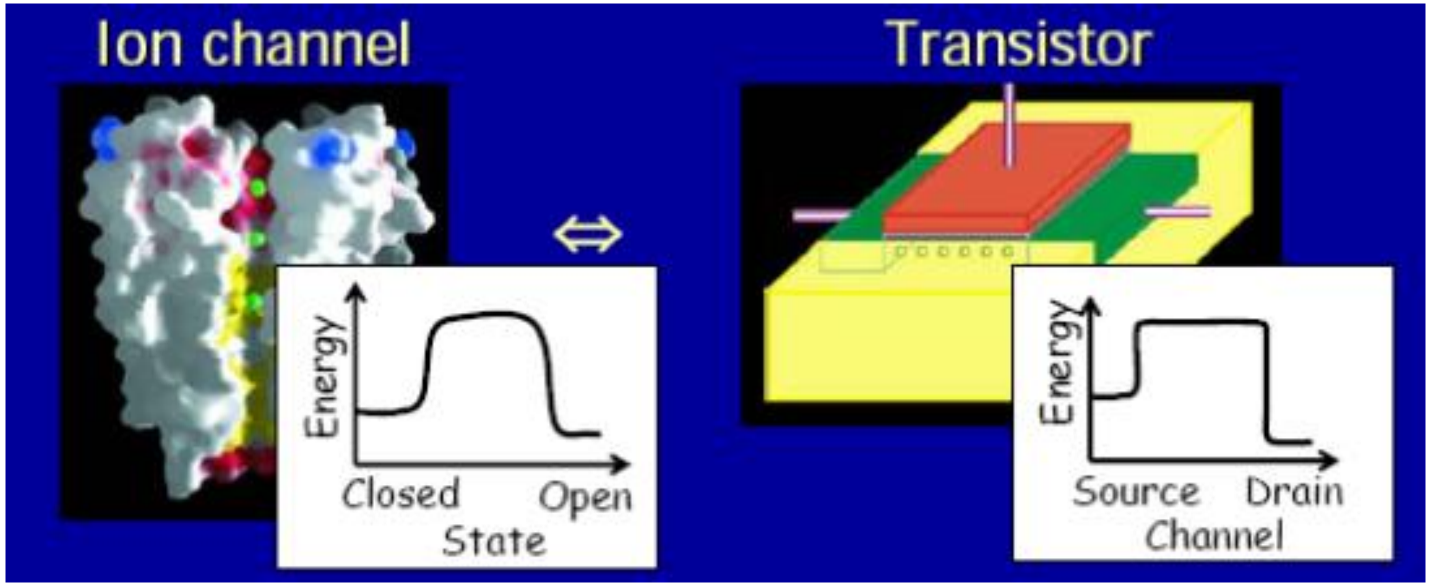
- Neuromorphic Engineering is a little over 30 years old.
- It was conceived through a collaboration between **Carver Mead** - a leading researcher in VLSI, **Max Delbruck** - a Nobel prize winning biochemist, **Richard Feynmann** - a Nobel prize winning physicist, and **John Hopfield** - a leading researcher in computational intelligence.



Neuromorphic Engineering

THE EARLY DAYS

And the basic premise...



Neuromorphic Engineering

MOTIVATIONS

Two goals of Neuromorphic Engineering:

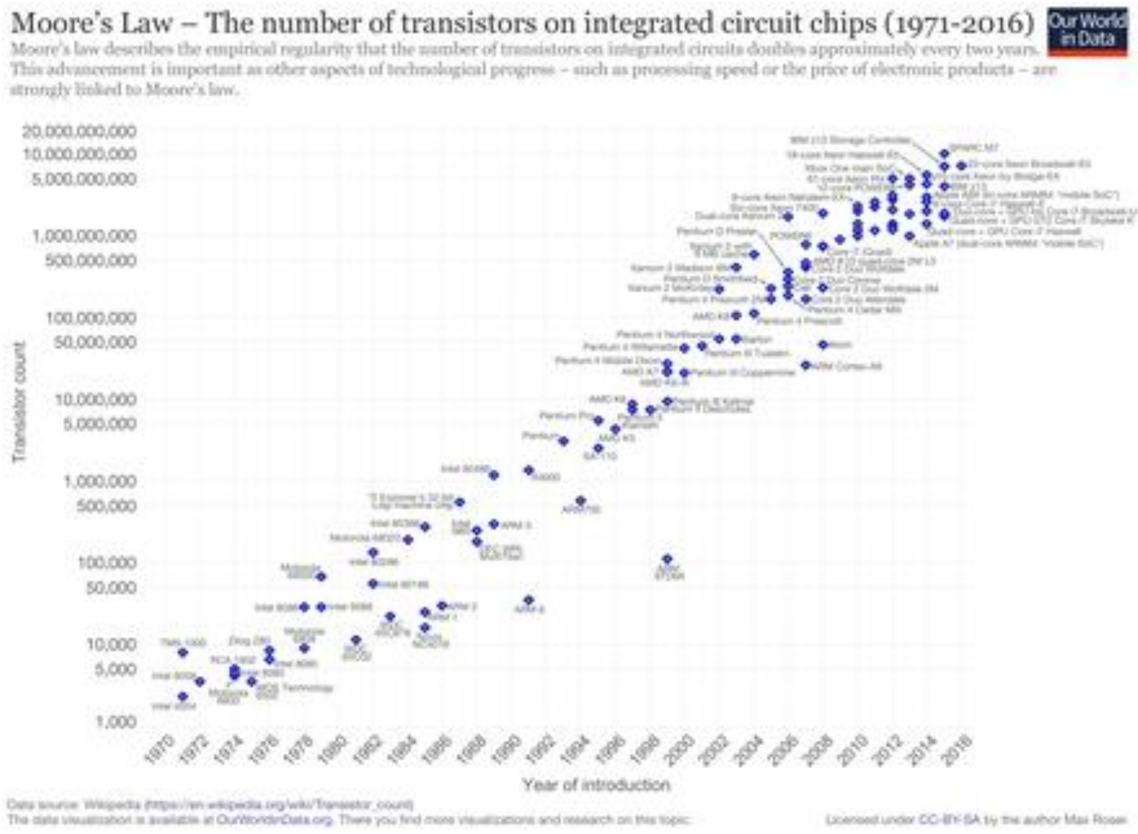
- **Science**: Understand the computational properties of biological neural systems using models implemented in Integrated Circuits;
- **Engineering**: Exploit the known properties of biological systems to design and implement efficient devices for engineering applications.

Neuromorphic Engineering

MOTIVATIONS

Moore's Law

- Every time you think it's going to end...



Neuromorphic Engineering

MOTIVATIONS

What will fill the gap when Moore's Law eventually ends?

- Memristors?
- Spintronics?
- Quantum computing?
- Can we do something different with CMOS?
- Something else?

Neuromorphic Engineering

RESEARCH

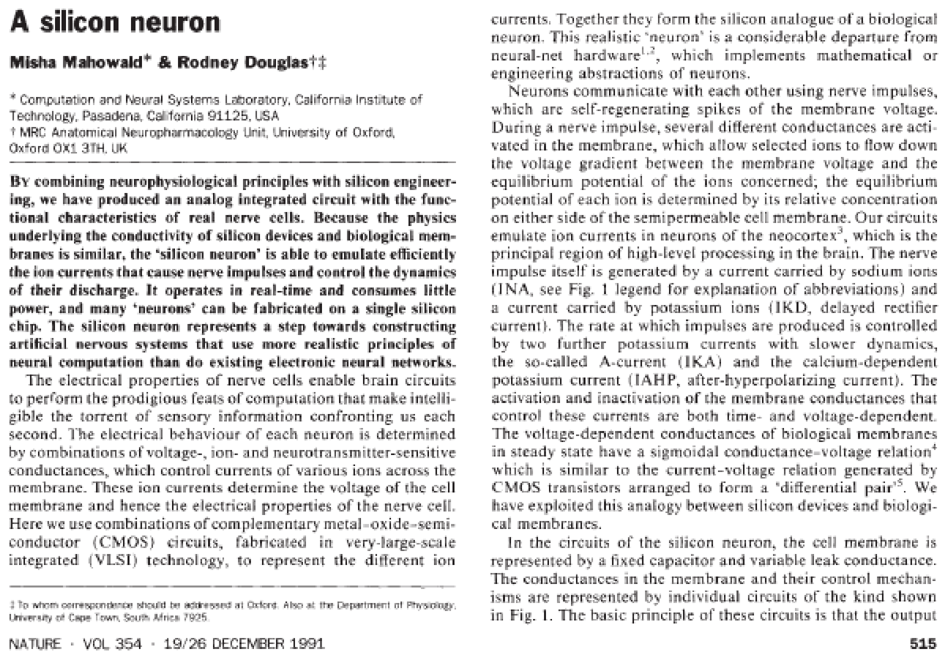
Early Neuromorphic Engineering (c. 1990 - 2008):

- Sensors
  - Vision - event-based retina
  - Hearing - silicon cochlea
- Integrated Circuit Design - low-power/current-mode
- Neuron modelling in Silicon
- Address-Event Representation (AER)
- Spiking computation

Neuromorphic Engineering

RESEARCH

Silicon Neurons



A silicon neuron

Misha Mahowald* & Rodney Douglas†‡

* Computation and Neural Systems Laboratory, California Institute of Technology, Pasadena, California 91125, USA
† MRC Anatomical Neuropharmacology Unit, University of Oxford, Oxford OX1 3TH, UK

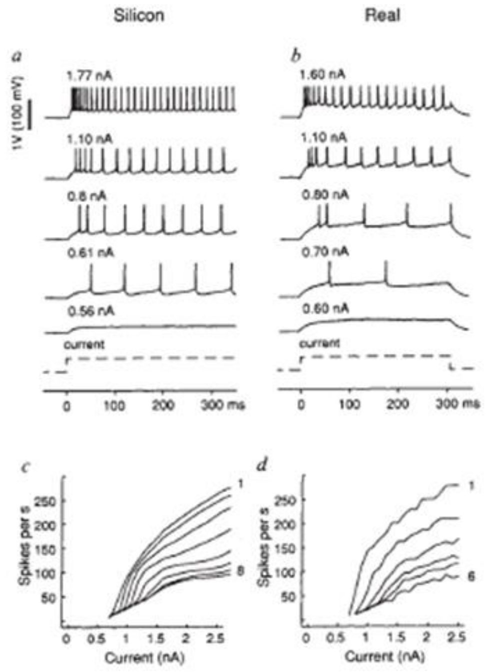
BY combining neurophysiological principles with silicon engineering, we have produced an analog integrated circuit with the functional characteristics of real nerve cells. Because the physics underlying the conductivity of silicon devices and biological membranes is similar, the 'silicon neuron' is able to emulate efficiently the ion currents that cause nerve impulses and control the dynamics of their discharge. It operates in real-time and consumes little power, and many 'neurons' can be fabricated on a single silicon chip. The silicon neuron represents a step towards constructing artificial nervous systems that use more realistic principles of neural computation than do existing electronic neural networks.

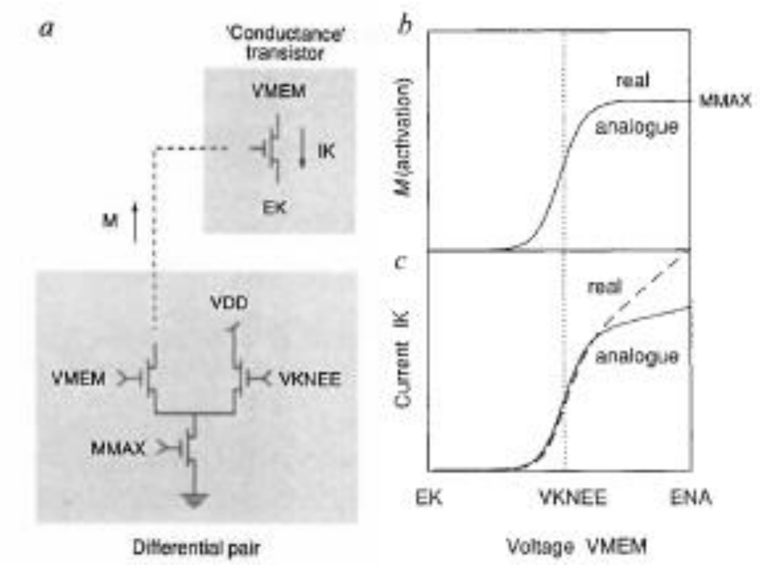
The electrical properties of nerve cells enable brain circuits to perform the prodigious feats of computation that make intelligible the torrent of sensory information confronting us each second. The electrical behaviour of each neuron is determined by combinations of voltage-, ion- and neurotransmitter-sensitive conductances, which control currents of various ions across the membrane. These ion currents determine the voltage of the cell membrane and hence the electrical properties of the nerve cell. Here we use combinations of complementary metal–oxide–semiconductor (CMOS) circuits, fabricated in very-large-scale integrated (VLSI) technology, to represent the different ion currents. Together they form the silicon analogue of a biological neuron. This realistic 'neuron' is a considerable departure from neural-net hardware[1,2], which implements mathematical or engineering abstractions of neurons.

Neurons communicate with each other using nerve impulses, which are self-regenerating spikes of the membrane voltage. During a nerve impulse, several different conductances are activated in the membrane, which allow selected ions to flow down the voltage gradient between the membrane voltage and the equilibrium potential of the ions concerned; the equilibrium potential of each ion is determined by its relative concentration on either side of the semipermeable cell membrane. Our circuits emulate ion currents in neurons of the neocortex[3], which is the principal region of high-level processing in the brain. The nerve impulse itself is generated by a current carried by sodium ions (INA, see Fig. 1 legend for explanation of abbreviations) and a current carried by potassium ions (IKD, delayed rectifier current). The rate at which impulses are produced is controlled by two further potassium currents with slower dynamics, the so-called A-current (IKA) and the calcium-dependent potassium current (IAHP, after-hyperpolarizing current). The activation and inactivation of the membrane conductances that control these currents are both time- and voltage-dependent. The voltage-dependent conductances of biological membranes in steady state have a sigmoidal conductance–voltage relation[4] which is similar to the current–voltage relation generated by CMOS transistors arranged to form a 'differential pair'[5]. We have exploited this analogy between silicon devices and biological membranes.

In the circuits of the silicon neuron, the cell membrane is represented by a fixed capacitor and variable leak conductance. The conductances in the membrane and their control mechanisms are represented by individual circuits of the kind shown in Fig. 1. The basic principle of these circuits is that the output

‡ To whom correspondence should be addressed at Oxford. Also at the Department of Physiology, University of Cape Town, South Africa 7925.

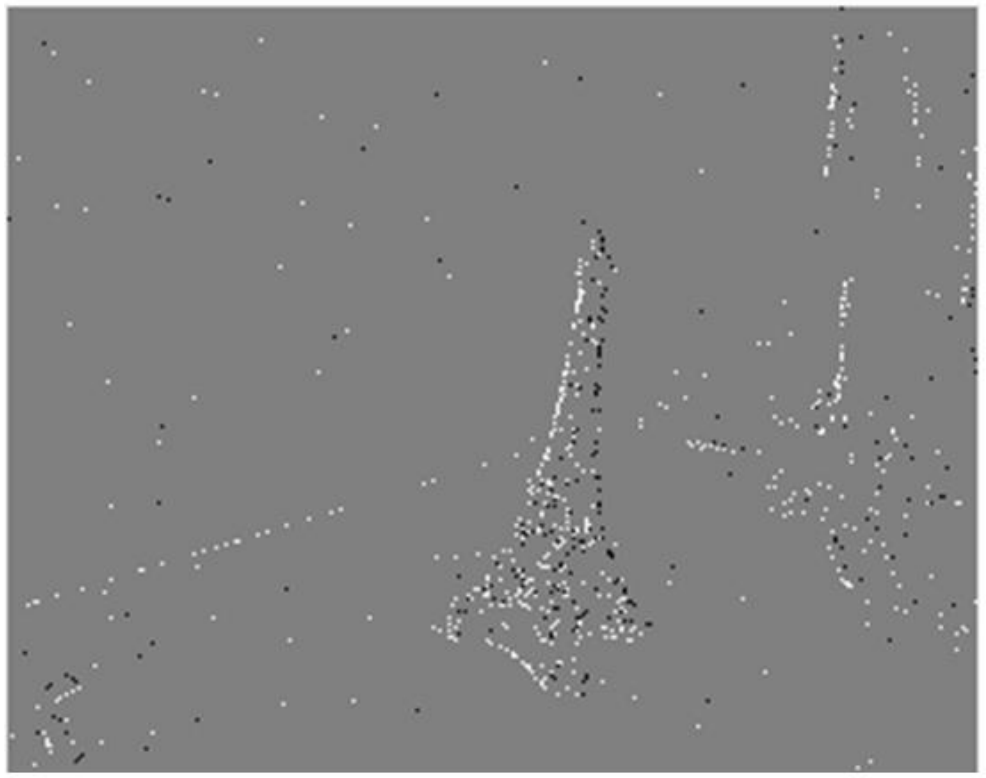




Neuromorphic Engineering

RESEARCH

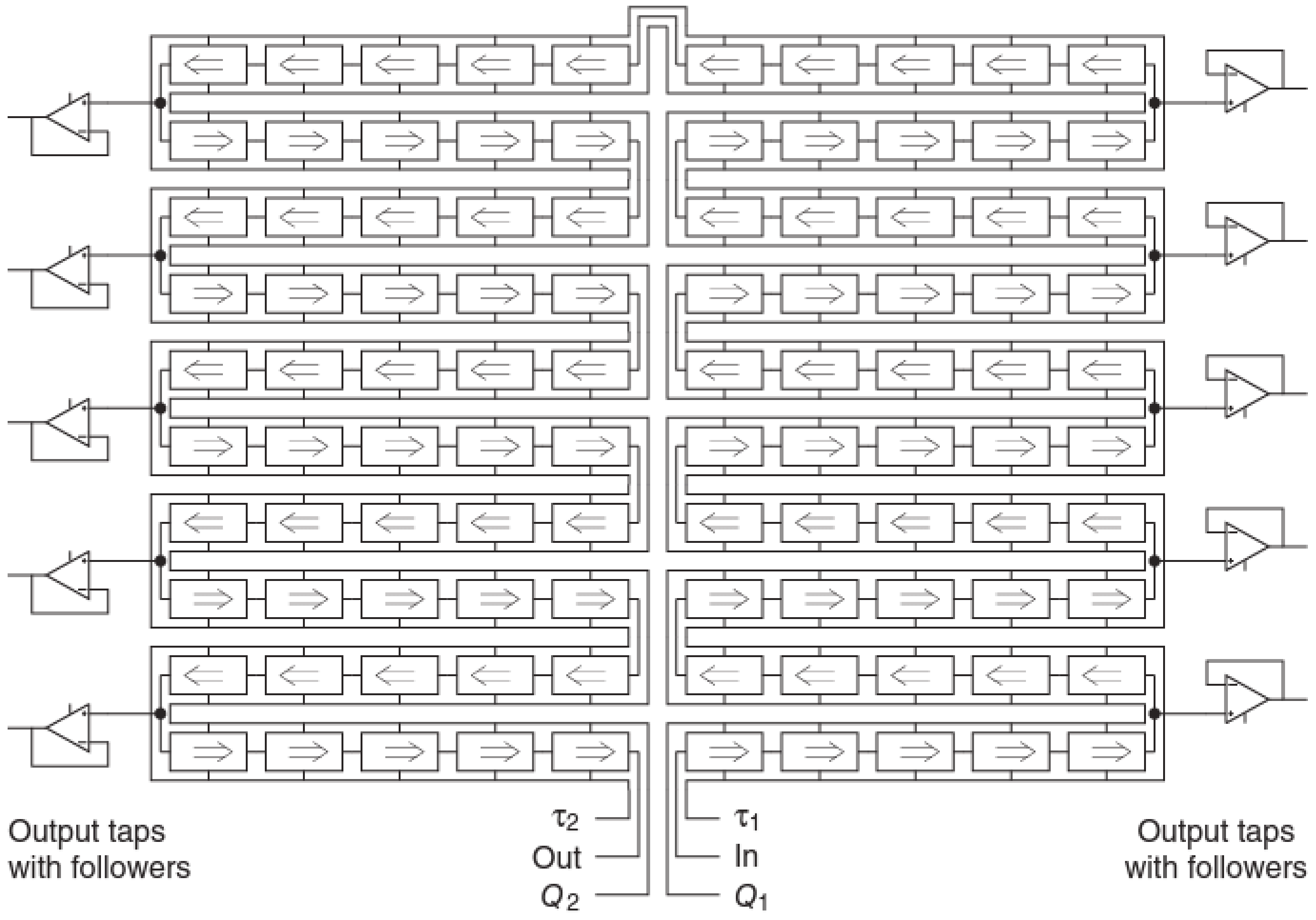
Silicon Retina/Event-based Cameras (https://www.youtube.com/watch?v=feBozLYZhB8**)**



Neuromorphic Engineering

RESEARCH

Silicon Cochlea



Neuromorphic Engineering

RESEARCH

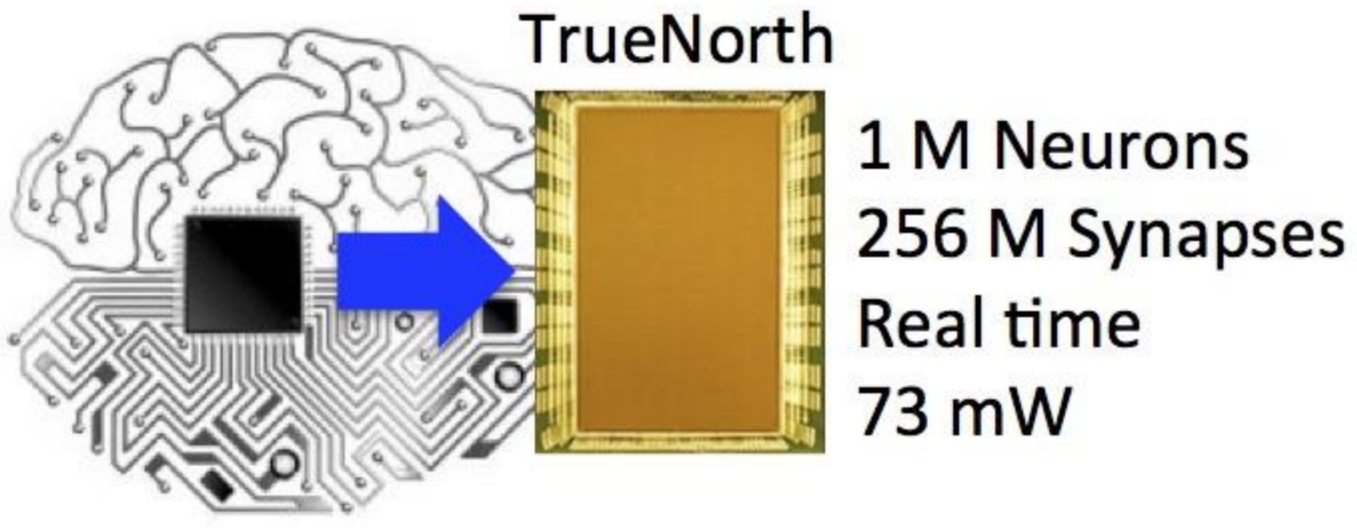
Commercial Products

Neuromorphic Engineering

RESEARCH

More recently:

- Computational Algorithms
- Neuromorphic Processors
- Spiking Neural Networks (SNNs)
- Machine Learning/Artificial Intelligence



Neuromorphic Engineering

RESEARCH

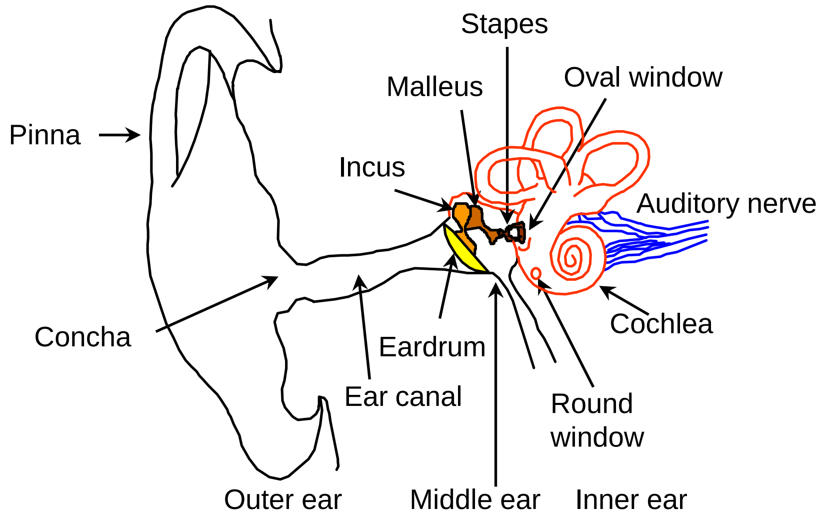
My Early Work

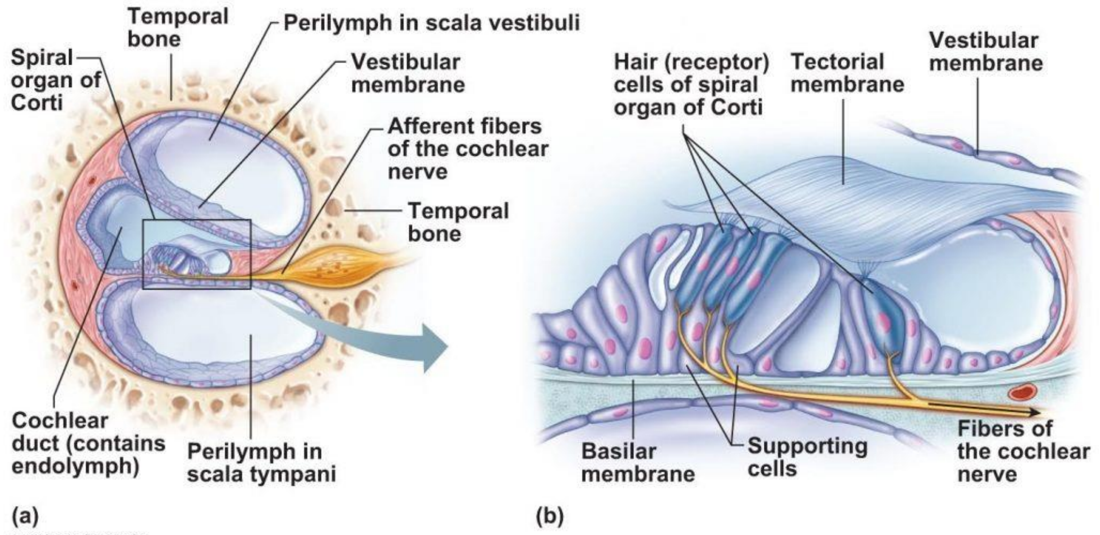
- Silicon Cochlea
- Silicon Neurons

Neuromorphic Engineering

RESEARCH

Silicon Cochlea



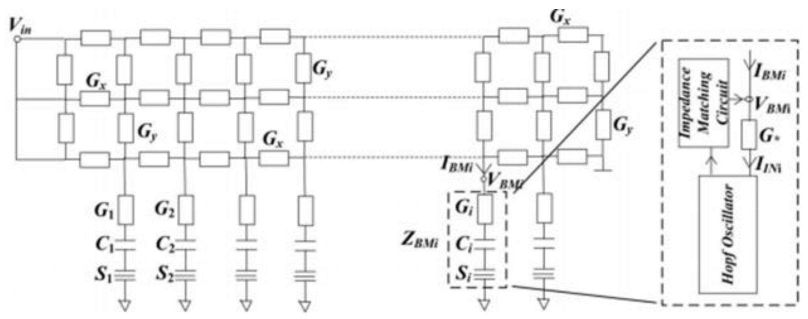

Temporal bone
Perilymph in scala vestibuli
Spiral organ of Corti
Vestibular membrane
Afferent fibers of the cochlear nerve
Temporal bone
Cochlear duct (contains endolymph)
Perilymph in scala tympani
(a)
Hair (receptor) cells of spiral organ of Corti
Tectorial membrane
Vestibular membrane
Basilar membrane
Supporting cells
Fibers of the cochlear nerve
(b)

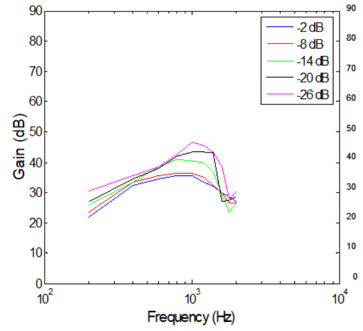
Neuromorphic Engineering

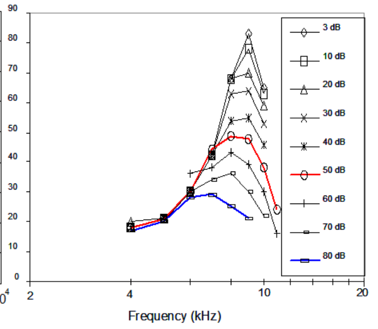
RESEARCH

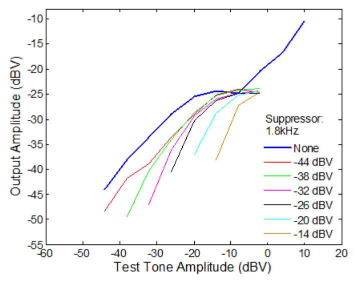
Silicon Cochlea

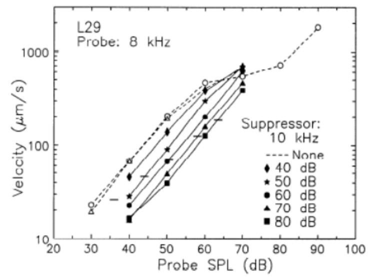
- 2D coupled oscillator model
- implemented in 0.5um and 1.2um technology (Am I giving away my age here...?)

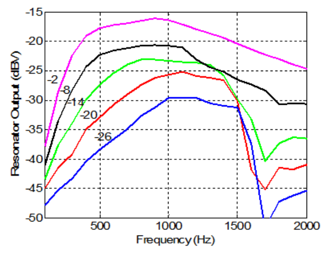


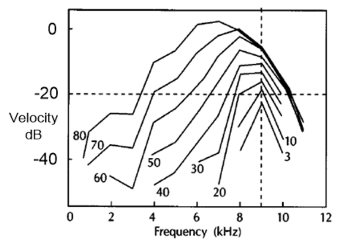


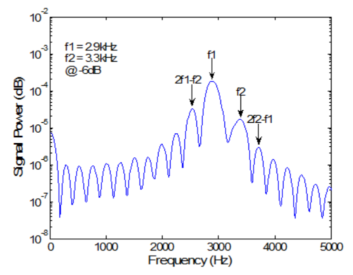


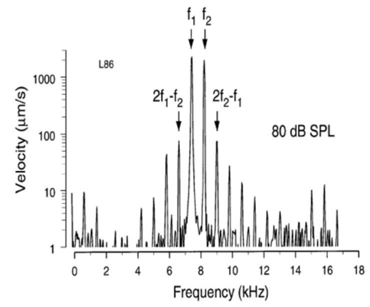












Neuromorphic Engineering

RESEARCH

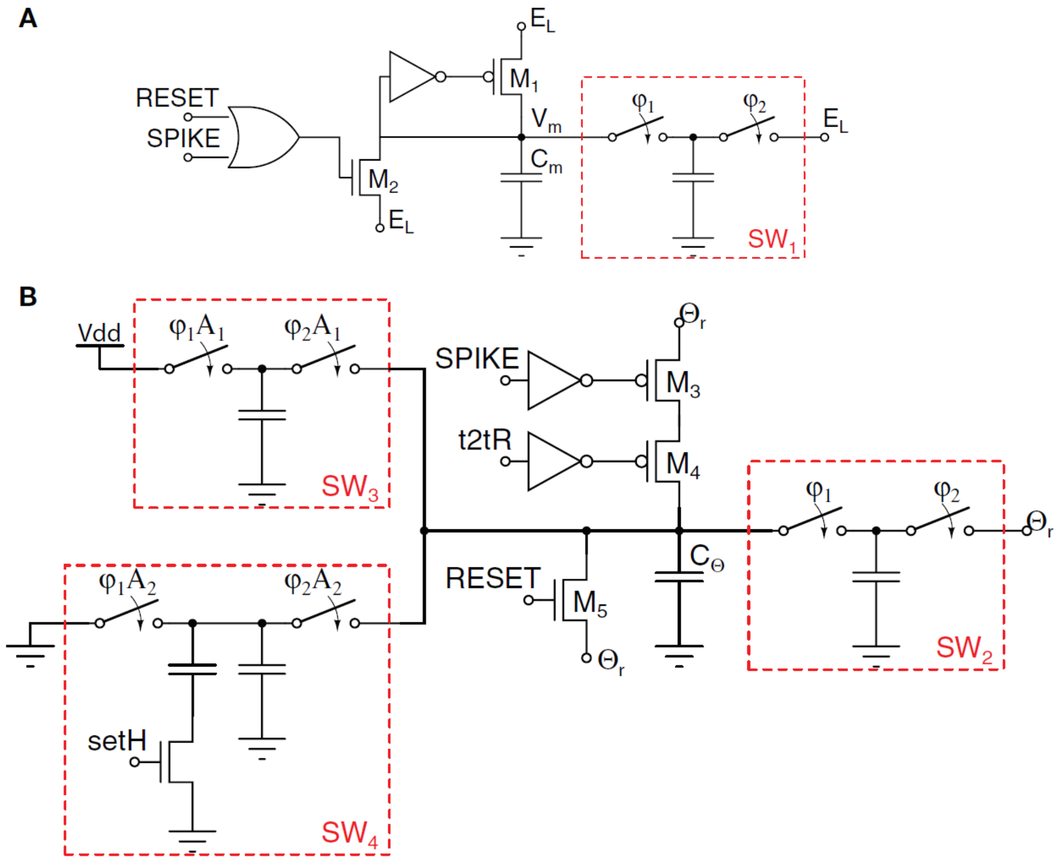
Looks good but...

- highly unstable
- upwards of 30 parameters to tune!
- current-mode, subthreshold - very noisy and mismatch between sections of the cochlea made it very hard!

Neuromorphic Engineering

RESEARCH

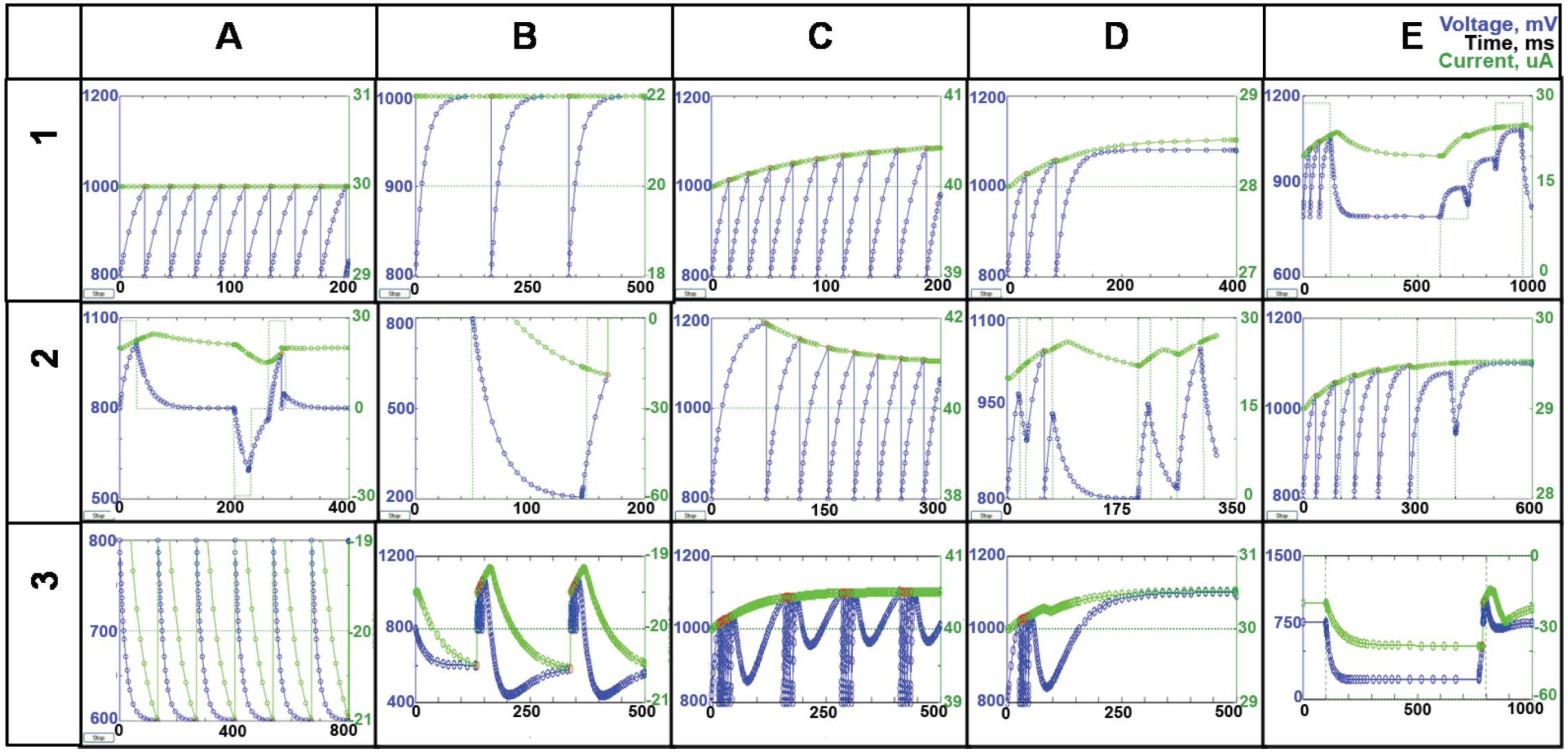
- The Generalized Integrate and Fire Neuron (Mihalas and Niebur 2008)



Neuromorphic Engineering

RESEARCH

Silicon neurons posed similar problems with added problem...what were they good for...? Nice plots...!?



Neuromorphic Engineering

RESEARCH

A side note on “circuits”:

- for me circuit is a word to describe connected components rather than a "rigid" unchangeable structure
- circuits in neuromorphics **must be parametric or tunable** or they cannot be used in computation
- trained neural networks are rigid and only useful to the things that they have been trained - they are set in stone. These are useful things but they probably have dead-ends - scaling issues, saturating performance and so on. They probably aren’t the best models if we want to understand and build a brain!

Neuromorphic Engineering

RESEARCH

Problems:

- Really, really difficult to scale. And how do we deal with all the connections???
- Low-power circuits (particularly using transistors in the subthreshold region) are too noisy and have too much mismatch between elements (even on the same chip). These were issues that the "neuromorphic approach" was supposed to overcome.
- Digital is so cheap, reliable, and power consumption continues to improve...
- Asynchronous, often spike-based, computation is really difficult to interface with. In the end, despite fancy, neuromorphic circuits and systems, outputs are often "framed" and much of the temporal detail is lost.

Neuromorphic Engineering

RESEARCH

But still neuromorphic approaches are appealing:

- parallel computation is expensive in digital
- matrix multiplication is expensive in digital
- these things we need for truly intelligent computation
- and (if you'll indulge me a little bit here)...digital "machine learning" is not *elegant*. It is brute force and lacks the artistry of an analog solution.

Stochastic Electronics

INSPIRATIONS

The work of Jonathan Tapson:

- autocorrelation properties of single neurons
- cross-correlation using noisy neurons
- Stochastic/Neural ADC

Stochastic Electronics

INSPIRATIONS

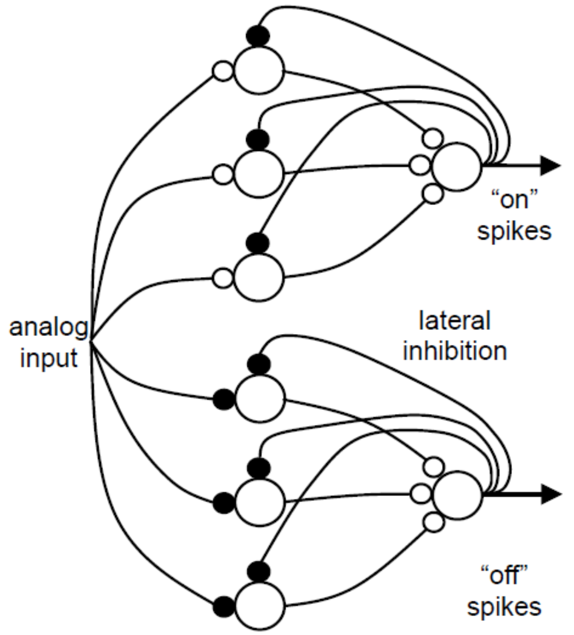
That (rather long) time I fell off the Neuromorphic wagon:

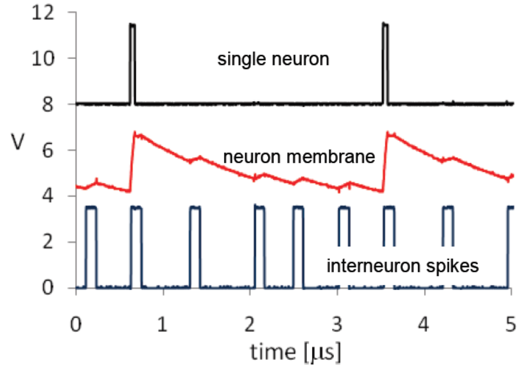
- Industry collaboration
- Time-to-Digital Converters (up to 10GHz)
  - Dynamic Element Matching
  - Random Element Switching (like injecting quantization noise)
- Nauta-Amplifier
- Using a cochlea as a spectrum analyser to measure impledance of bi-lipid layers

Stochastic Electronics

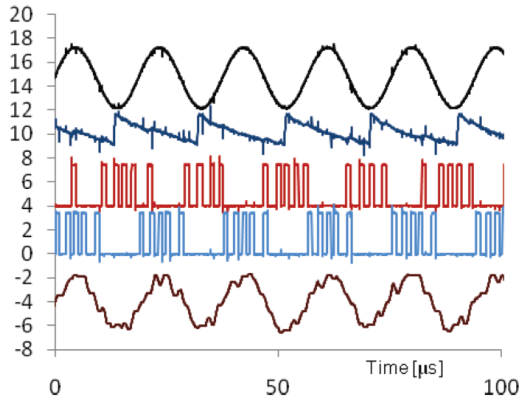
INSPIRATIONS AND EARLY WORK

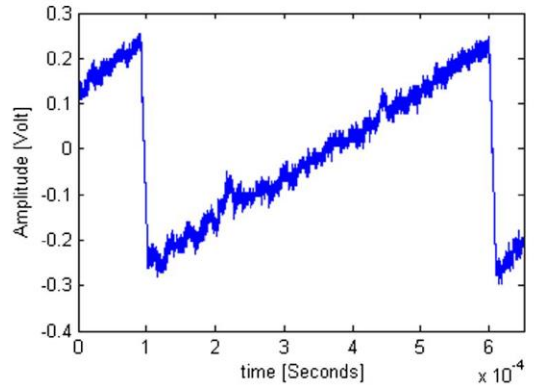
Stochastic/Neural ADC





Note: In this case the neuron "fires" when the membrane potential falls below a particular threshold (rather than the usual above threshold firing)

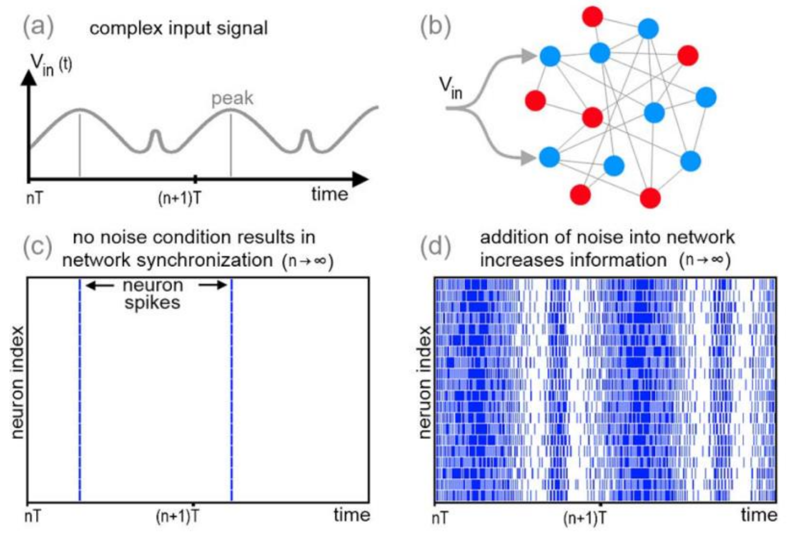


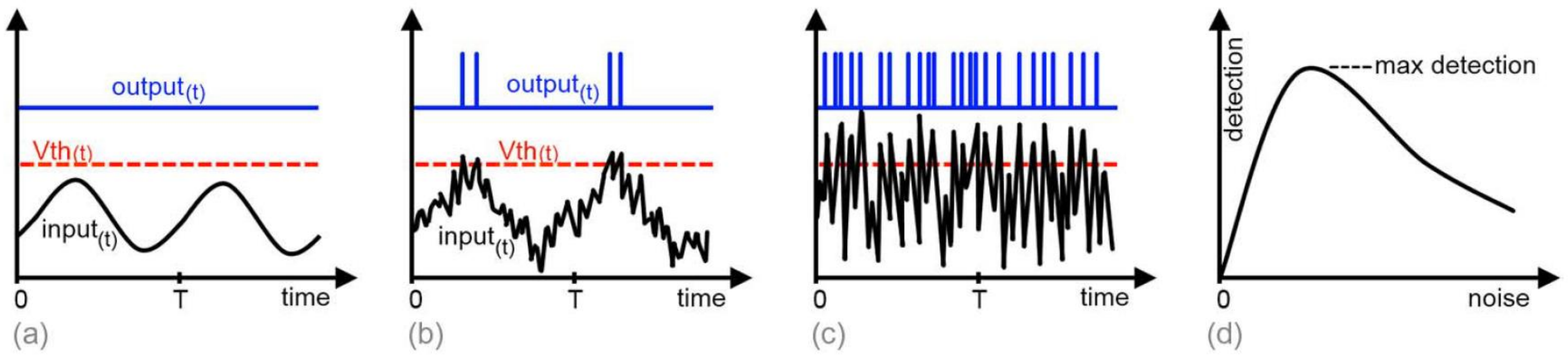


Four neurons with approximately 10 Hz spiking rate can sample a 2kHz waveform. Take that Nyquist!

Stochastic Electronics

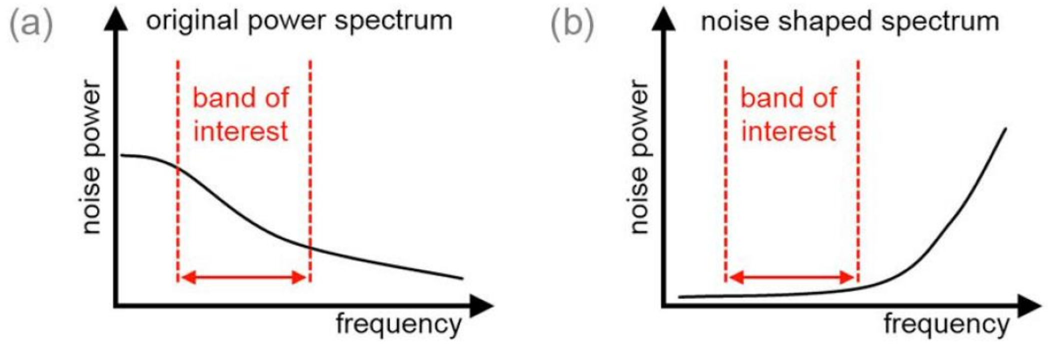
THE MANIFESTO

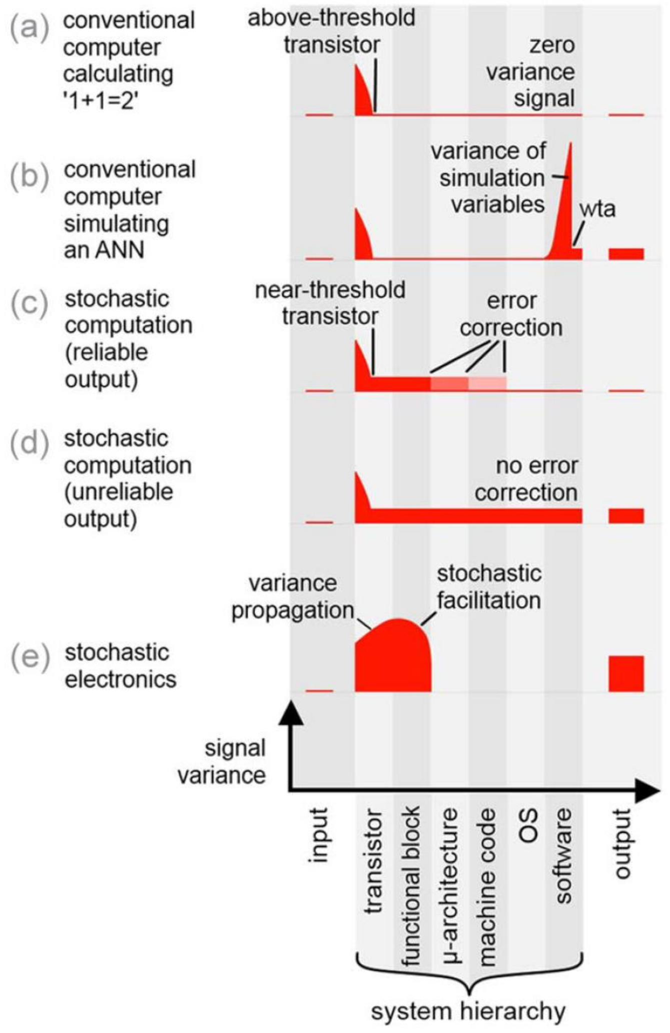




Stochastic Electronics

THE MANIFESTO

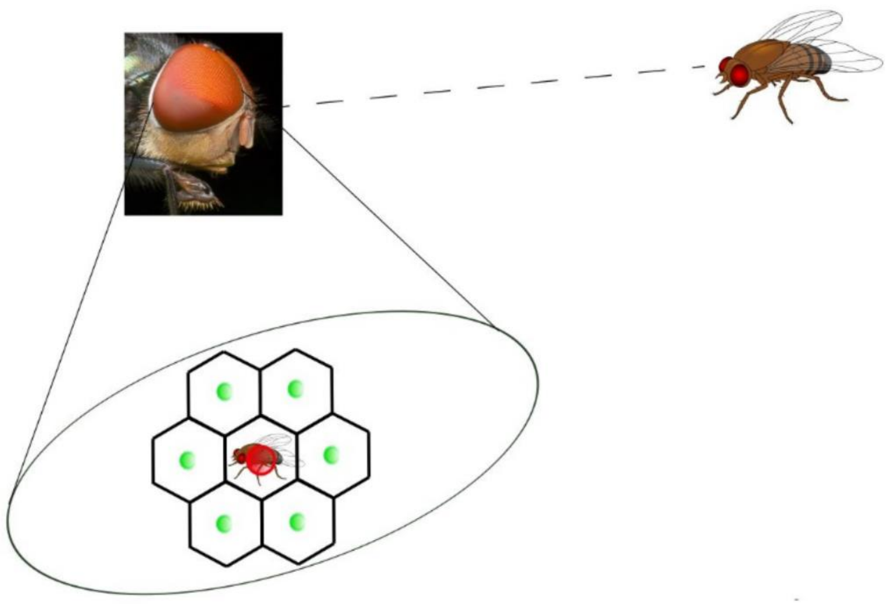


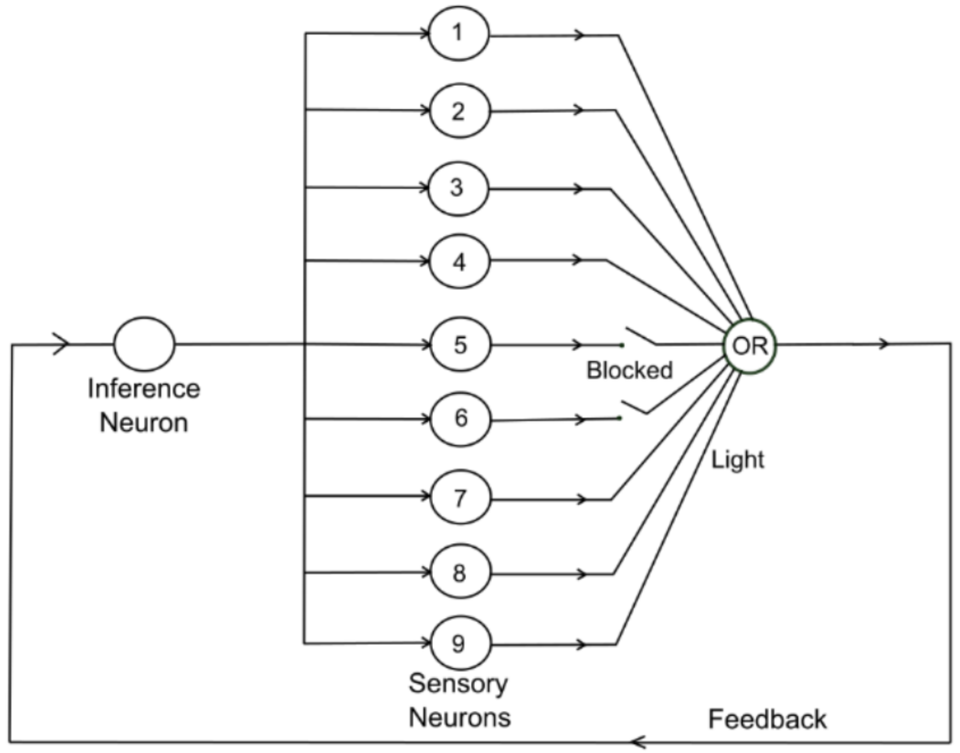

(a) conventional computer calculating '1+1=2'
above-threshold transistor
zero variance signal
(b) conventional computer simulating an ANN
variance of simulation variables
wta
(c) stochastic computation (reliable output)
near-threshold transistor
error correction
(d) stochastic computation (unreliable output)
no error correction
(e) stochastic electronics
variance propagation
stochastic facilitation
signal variance
input
transistor
functional block
μ-architecture
machine code
OS
software
output
system hierarchy

Stochastic Electronics

SOME COOL ALGORITHMS

Neuromorphic Competitive Control

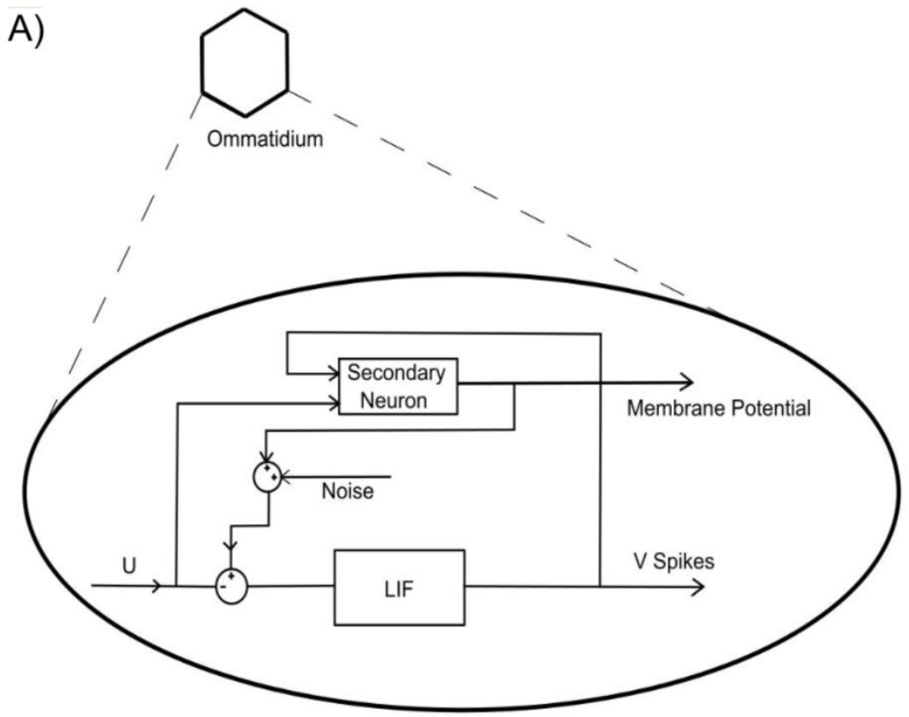


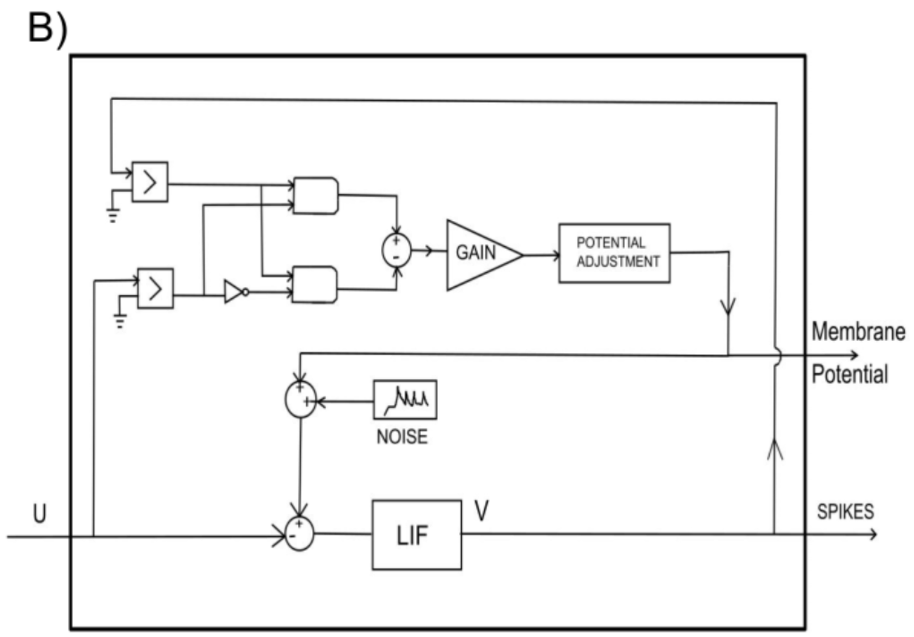


Stochastic Electronics

SOME COOL ALGORITHMS

Neuromorphic Competitive Control

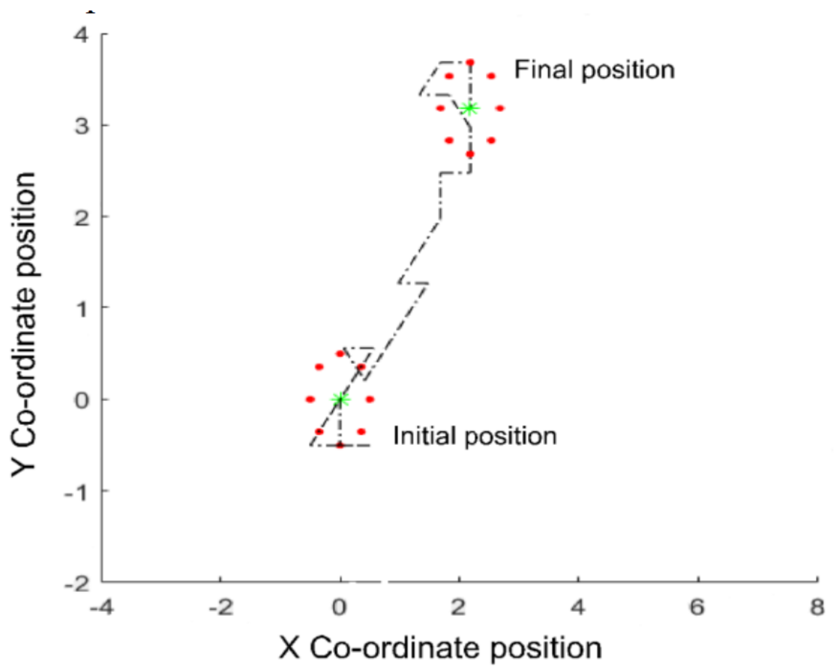


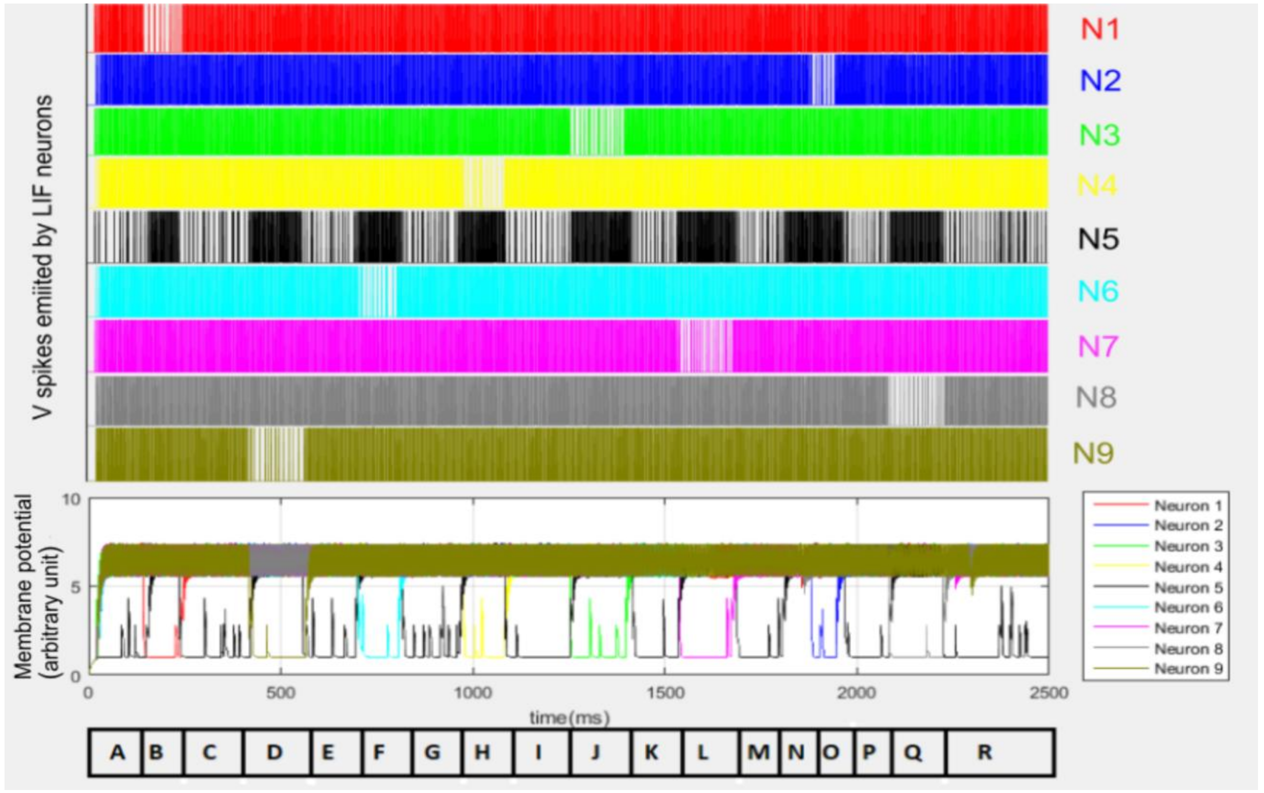


Stochastic Electronics

SOME COOL ALGORITHMS

Neuromorphic Competitive Control

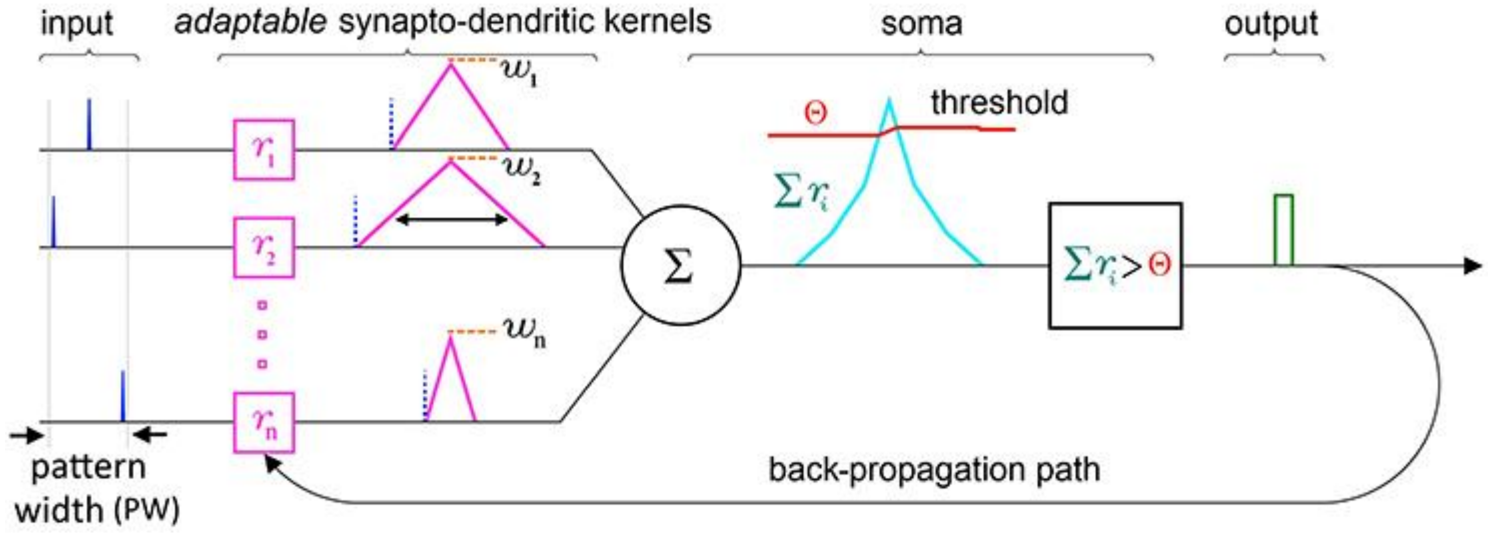


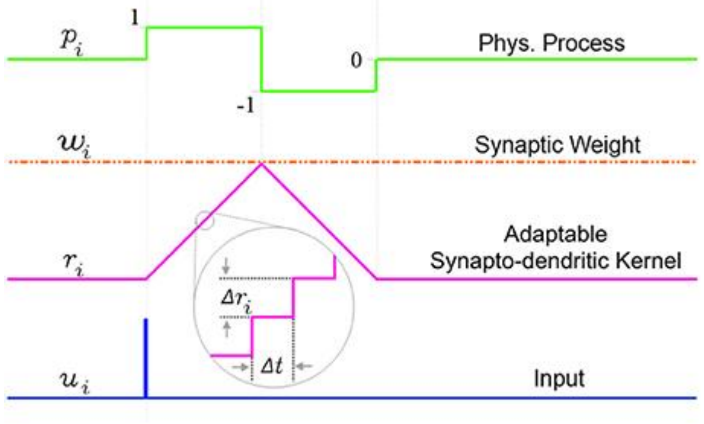


Stochastic Electronics

SOME COOL ALGORITHMS

Unsupervised Learning with the Synaptic Kernel Adaptation Neuron (SKAN)

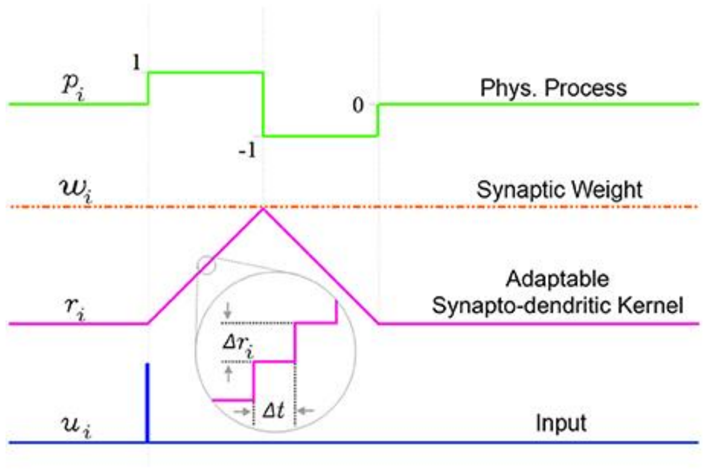


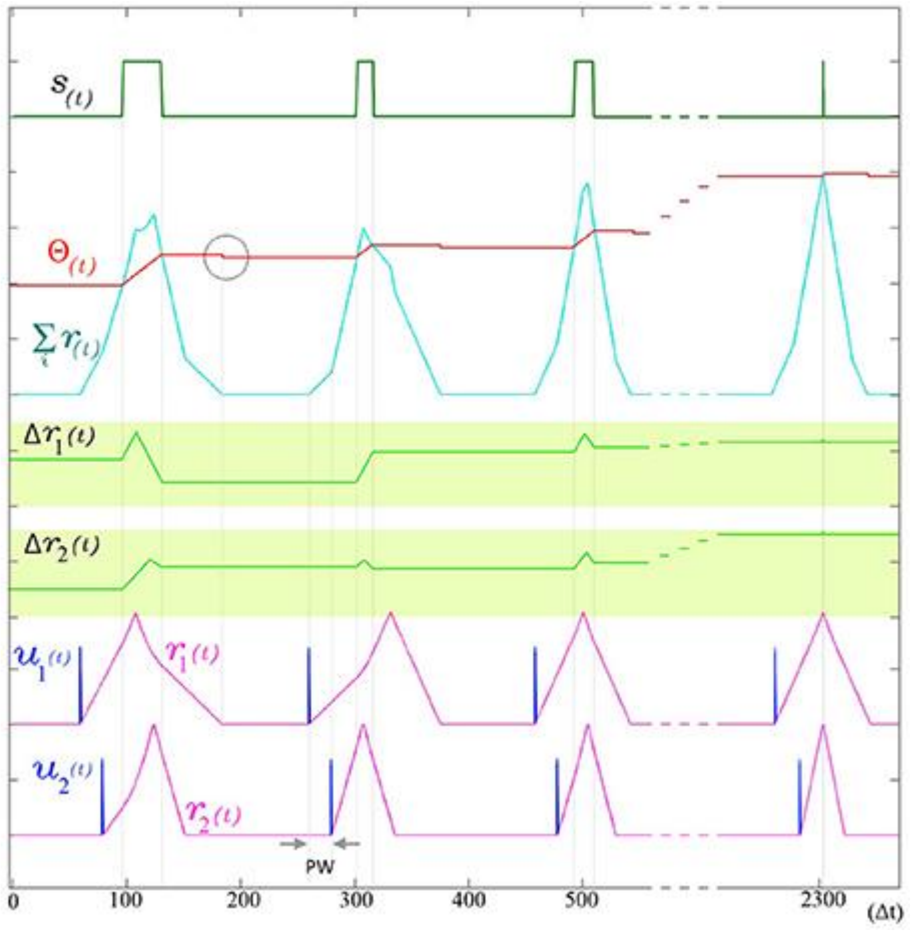


Stochastic Electronics

SOME COOL ALGORITHMS

Unsupervised Learning with the Synaptic Kernel Adaptation Neuron (SKAN)



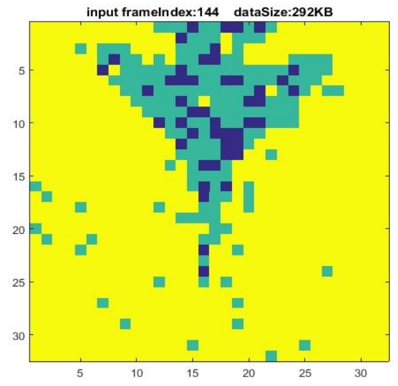


Stochastic Electronics

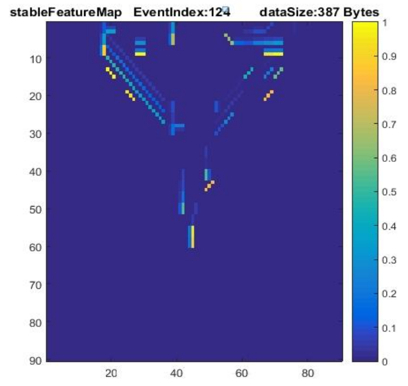
SOME COOL ALGORITHMS

Unsupervised Learning with the Synaptic Kernel Adaptation Neuron (SKAN)

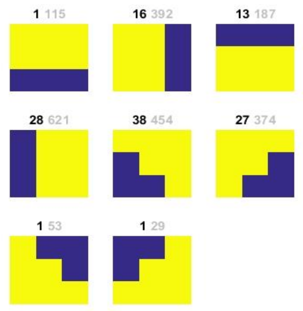
(a) Raw Input



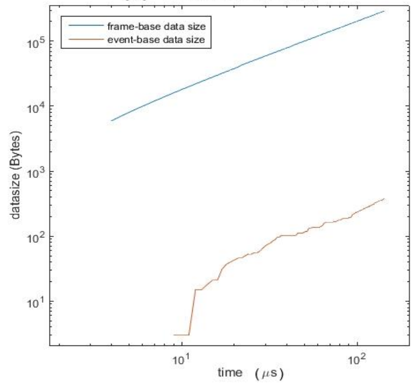
(b) Event Feature Surface

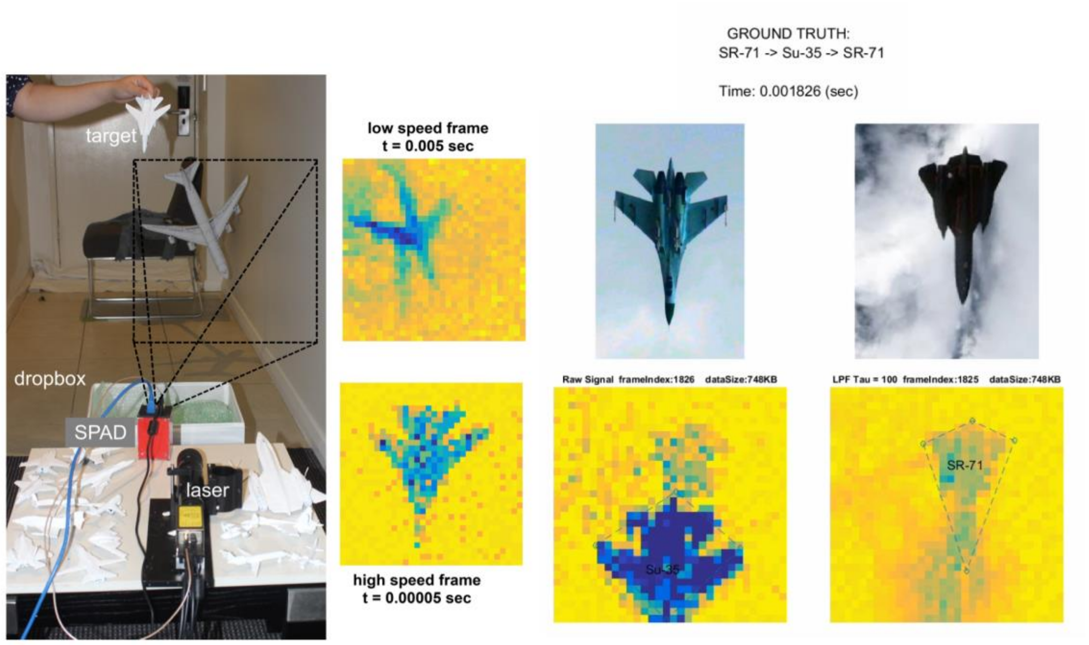


(c) 3x3 Features and Activations



(d) Generated Data

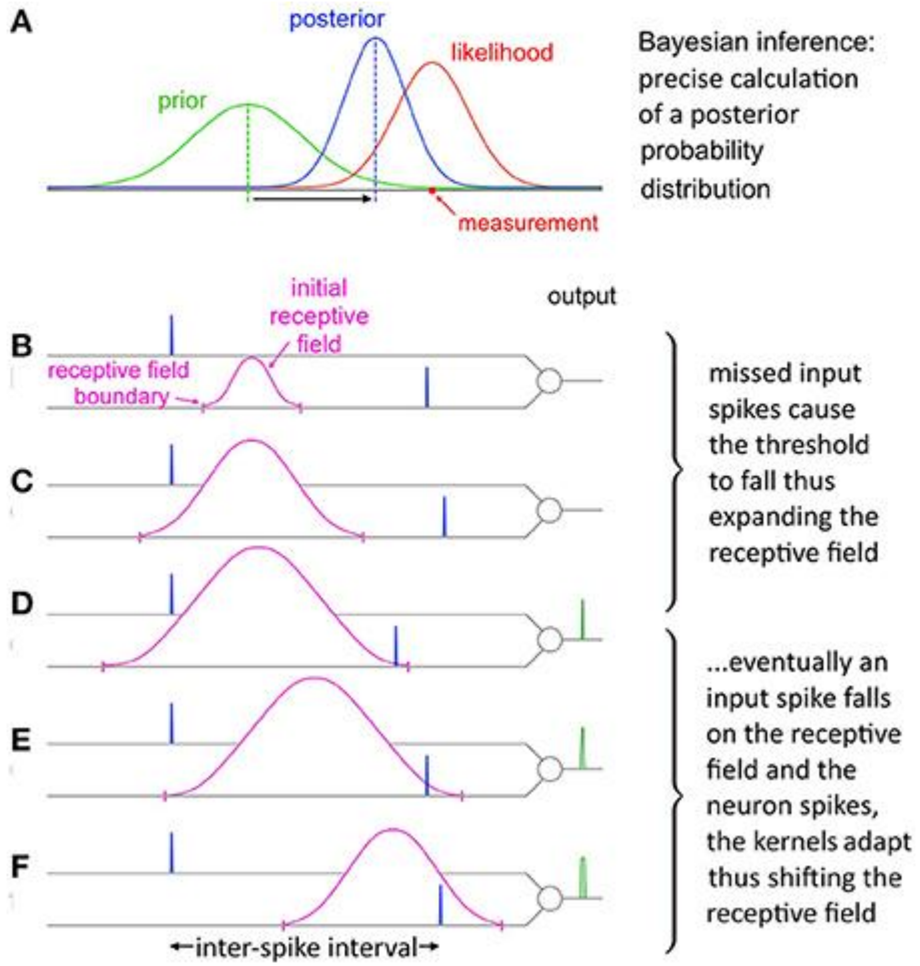




Stochastic Electronics

SOME COOL ALGORITHMS

Unsupervised Learning with the Synaptic Kernel Adaptation Neuron (SKAN)


A
posterior
likelihood
prior
measurement
Bayesian inference: precise calculation of a posterior probability distribution
initial receptive field
output
B
receptive field boundary
C
D
E
F
inter-spike interval
missed input spikes cause the threshold to fall thus expanding the receptive field
...eventually an input spike falls on the receptive field and the neuron spikes, the kernels adapt thus shifting the receptive field

Stochastic Electronics

THEMES

Stochastic Electronics principles:

- decoherence - we do not want neurons to synchronize – this is where the noise and mismatch helps!
- random spreading - neurons need to fire randomly and approximately cover the sample space
- Inhibition – turning off paths

Stochastic Computation

ORIGINS

The Work of Gaines in the 1960s

- Stochastic Computation was borne out of the need to do pattern recognition and parallel learning tasks (sound familiar???) back in the day when computers were enormous and took up entire rooms
- Fell out of favour due to the introduction of Integrated Circuits and particularly CMOS – binary had won the day!

Stochastic Computation

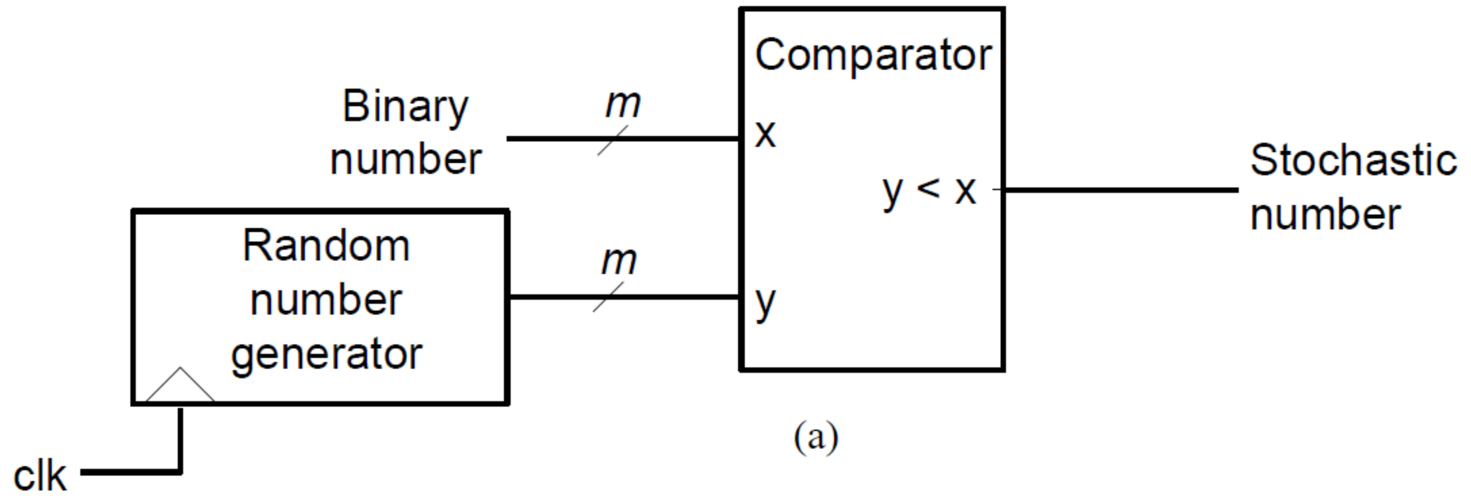
THE STOCHASTIC COMPUTING CONCEPT

- Numbers are interpreted as probabilities, they fall naturally into the interval [0,1].
- Numbers are represented by bit-streams (0,1,1,1,0,0,1,0,0,0,0,1,1,0,1,..) that can be processed by very simple digital circuits.
- E.g. 0.5 = 0,1,1,1,0,0,1,0 or 0,0,0,0,1,1,1,1 and so on (order does not matter).

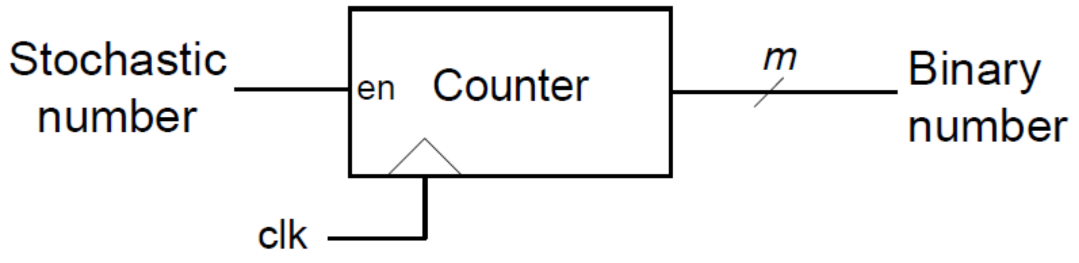
Stochastic Computation

ELEMENTS OF STOCHASTIC COMPUTATION

- Randomizer and derandomizer



(a)

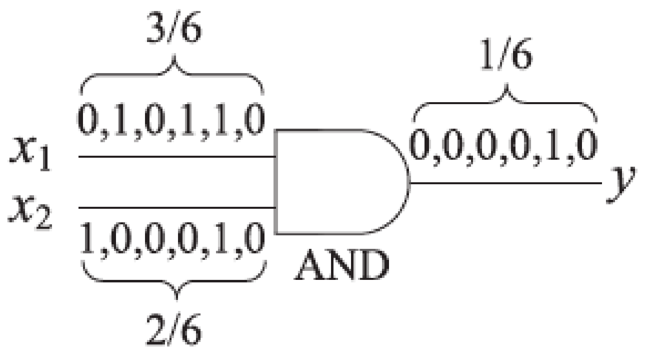


(b)

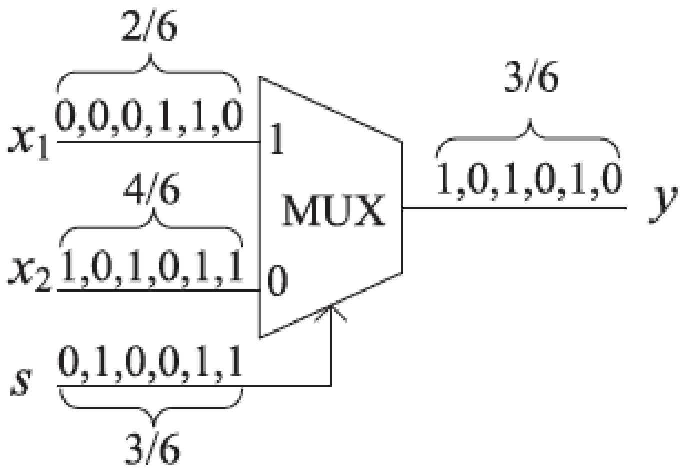
Stochastic Computation

ELEMENTS OF STOCHASTIC COMPUTATION

- Multiplication and Addition



(a)

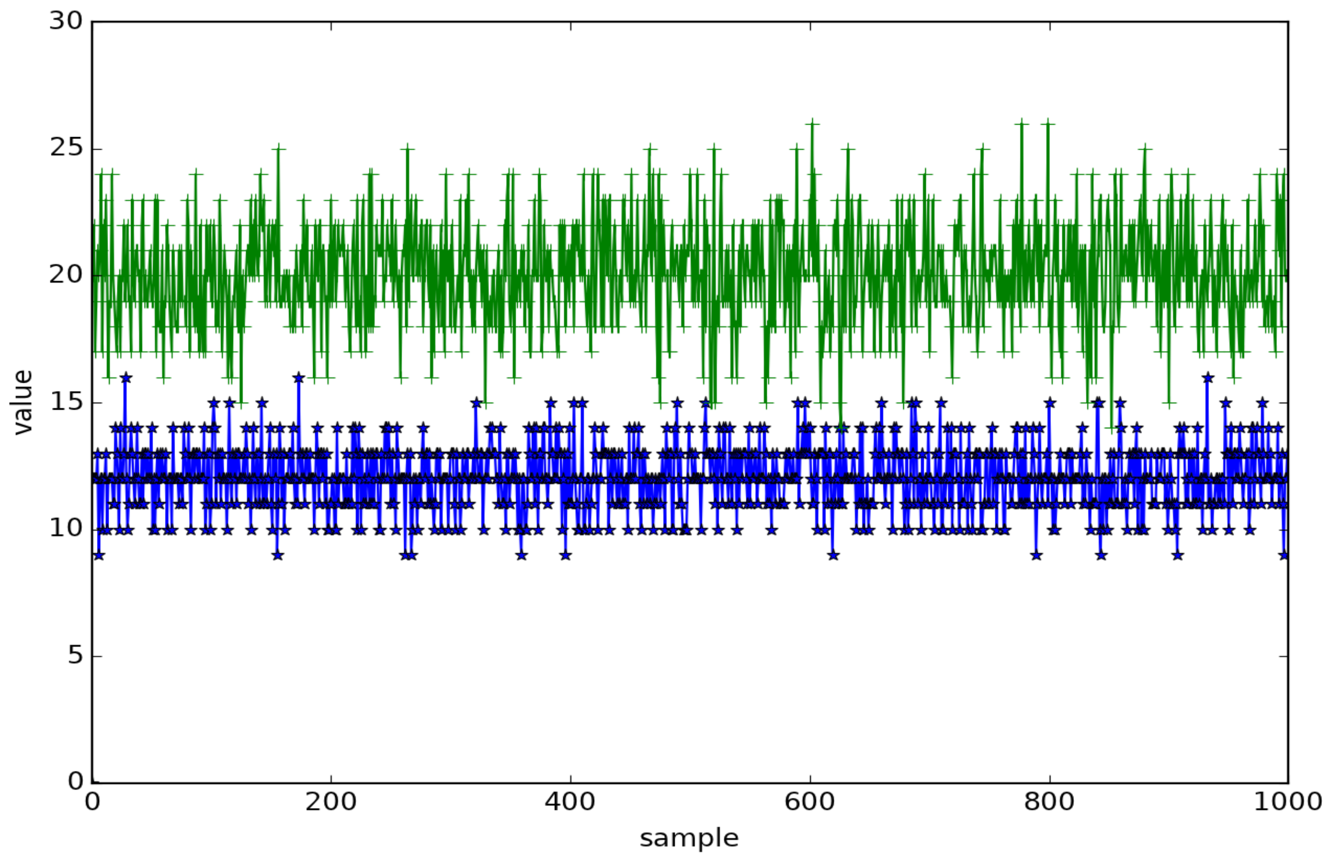


(b)

Stochastic Computation

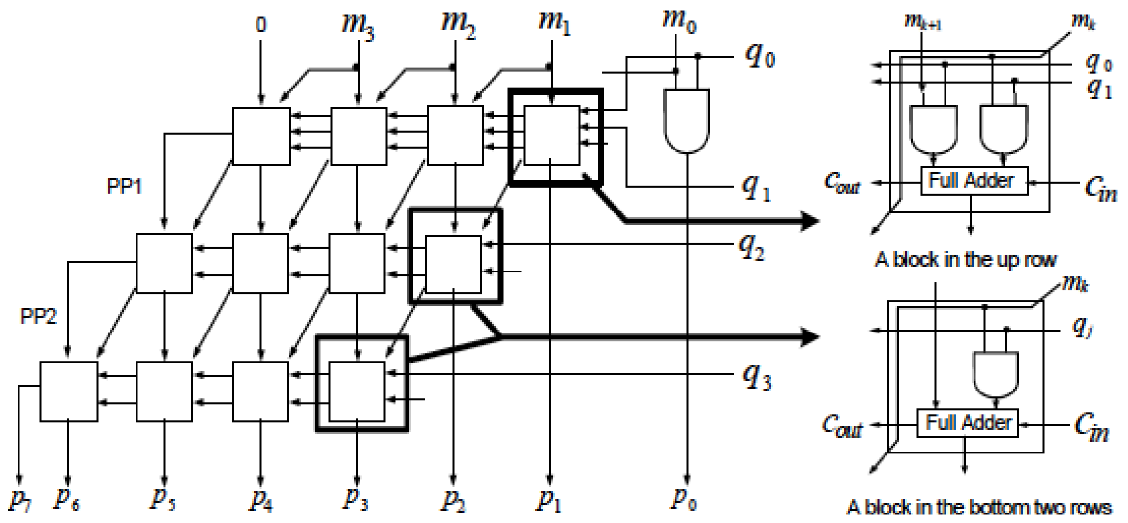
ELEMENTS OF STOCHASTIC COMPUTATION

- Multiplication in blue
  - X = 0.5, Y = 0.75
  - Z = 0.375
  - With 32 bits = 12
- Addition in green
  - X = 0.5, Y = 0.75
  - Z = 0.5(0.5+0.75) = 0.625
  - (note output is scaled by 0.5)
  - With 32 bits = 20

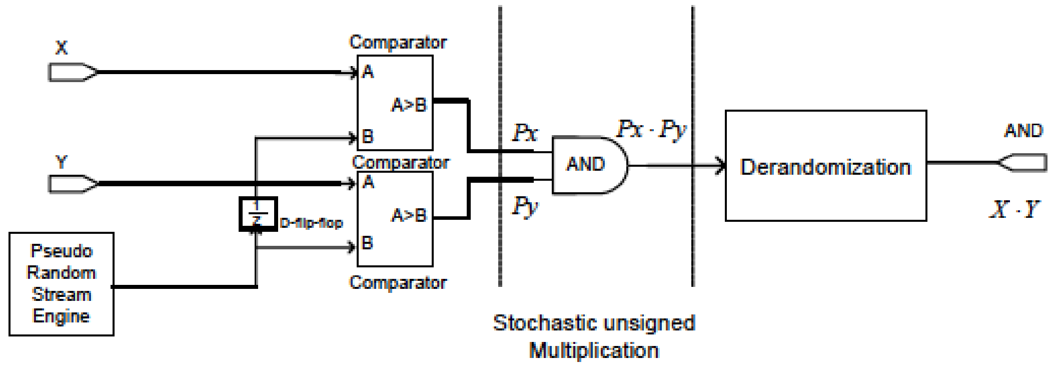


Stochastic Computation

ELEMENTS OF STOCHASTIC COMPUTATION



(a) A traditional 4×4 unsigned multiplier circuit [25]

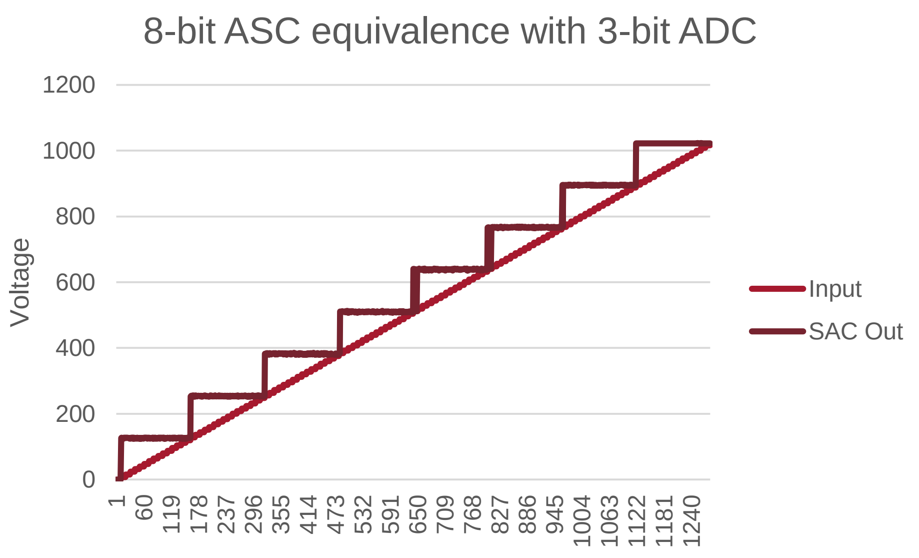


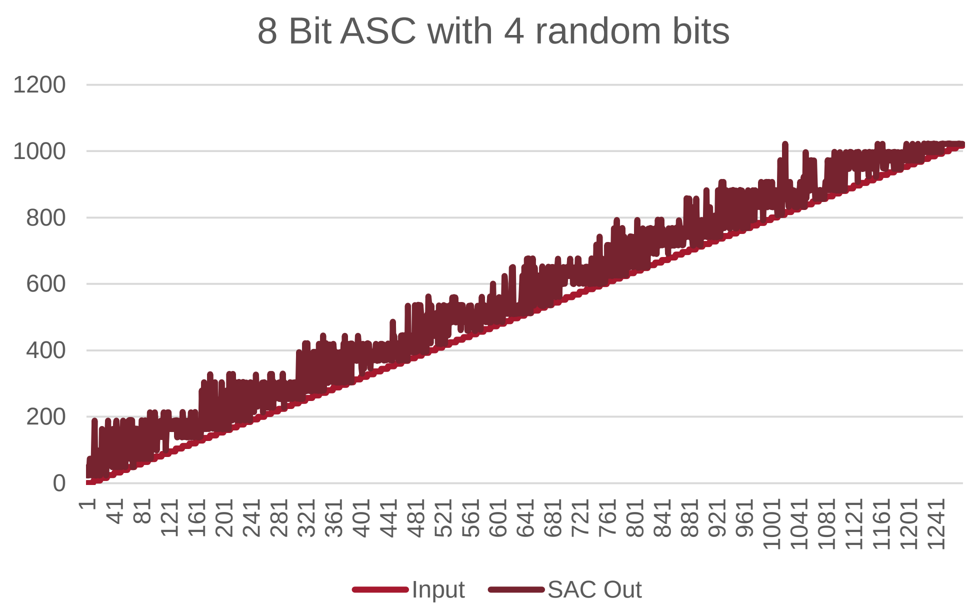
(b) The Stochastic implementation of unsigned multiplication

Stochastic Computation

ELEMENTS OF STOCHASTIC COMPUTATION

- Save digital resources in hardware, where area is precious
- High-fault tolerance capability and high reliability
- Noise tolerant…in fact noise can be a benefit





Conclusions

- Neuromorphic Engineering is at a cross-roads
- Hardware needs algorithms to be developed with the hardware in mind – regardless of whether it is CMOS or Spintronics or Memristors or…”the next thing”
- Stochastic Electronics algorithms are very efficient in area and power because they were developed with a hardware mindset
- Stochastic Computation is quite similar to Stochastic Electronics but goes into the digital domain – it can remove the multiplier problem but is capable of so much more!

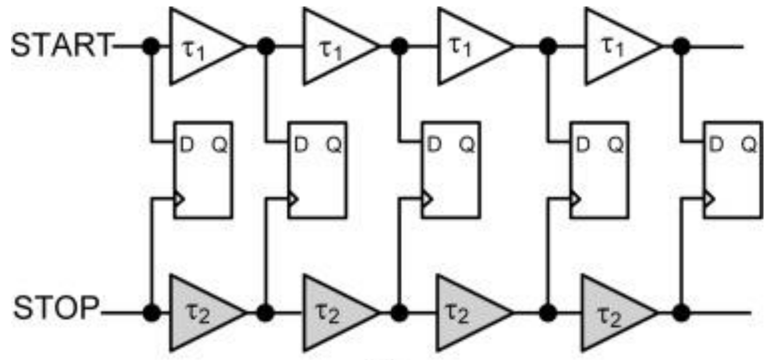
Acknowledgements

- All the Western Sydney University Team in particular:
  - Saeed Afshar
  - Chetan Singh Thakur (now at IIS)
  - Jonathan Tapson
  - Andrew Nicholson
  - Gaetano Gargiulo

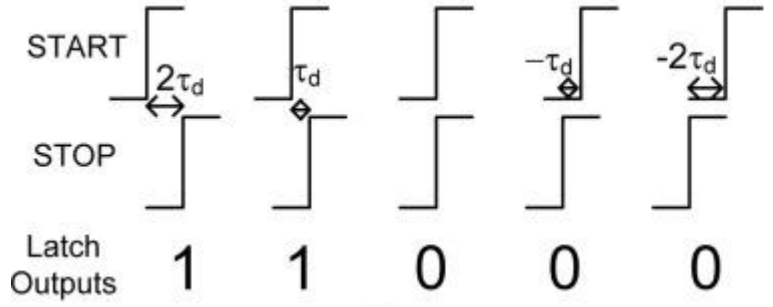
Stochastic Electronics

INSPIRATIONS

Time-to-Digital Converters (TDCs)



(a)

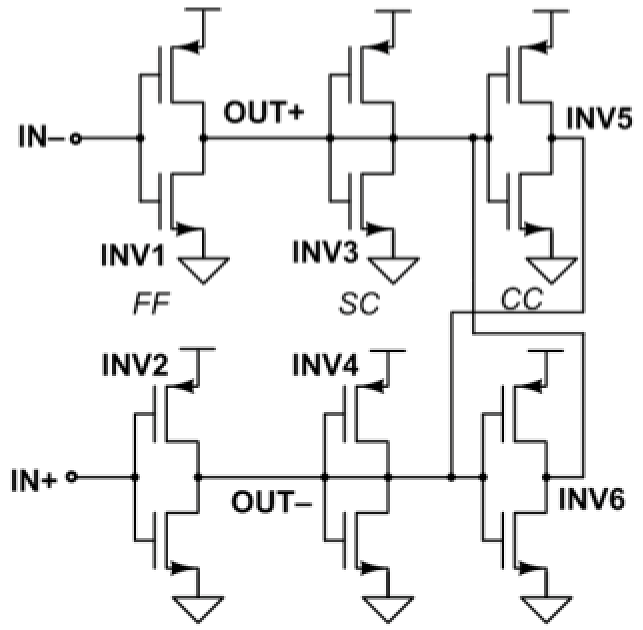


(b)

Stochastic Electronics

INSPIRATIONS

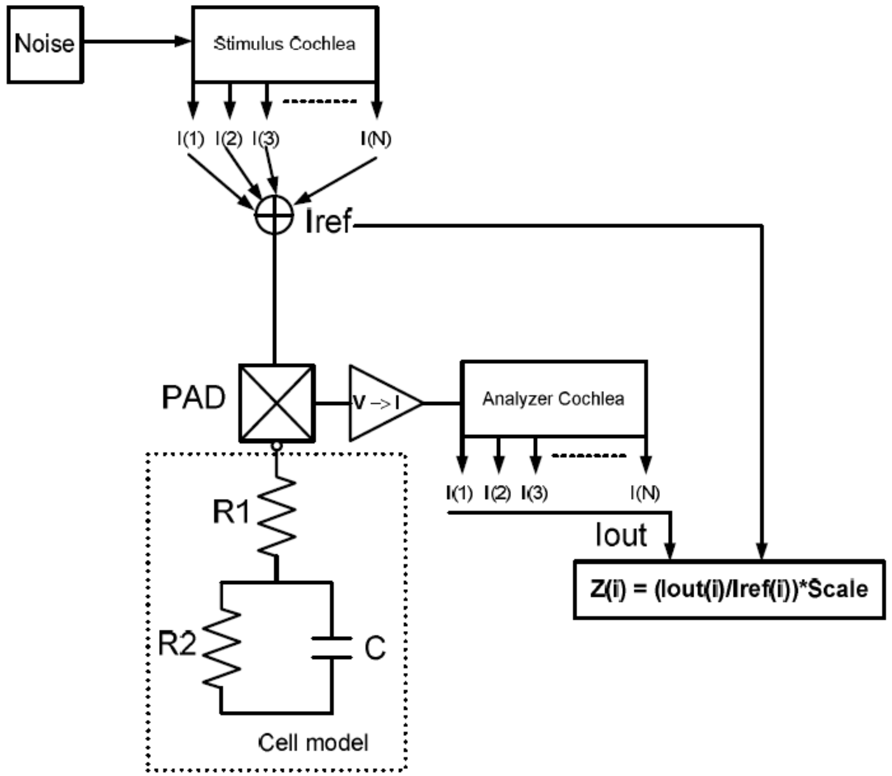
Nauta Amplifier



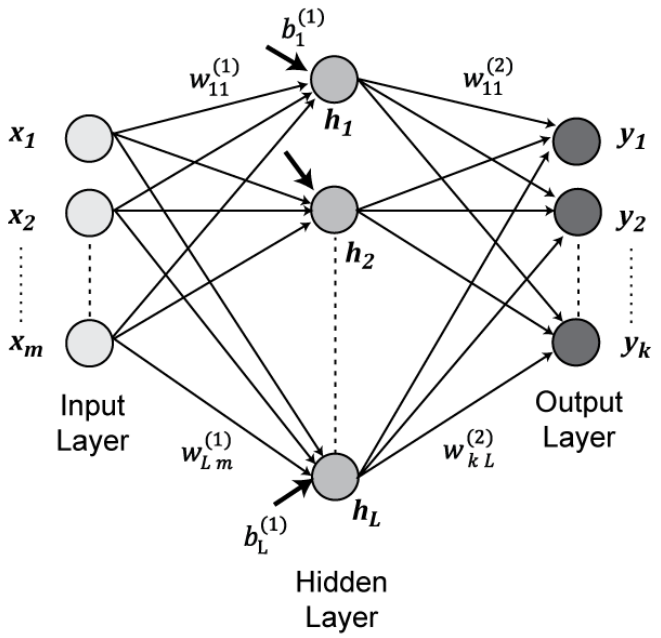
Stochastic Electronics

INSPIRATIONS

Cochlea as a spectrum analyser

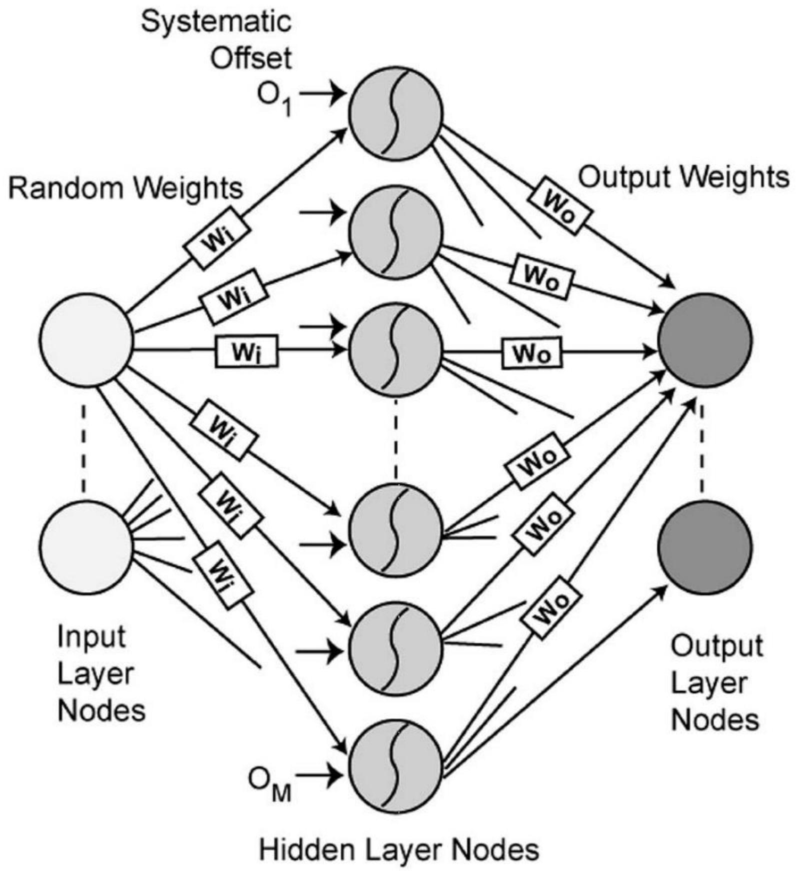


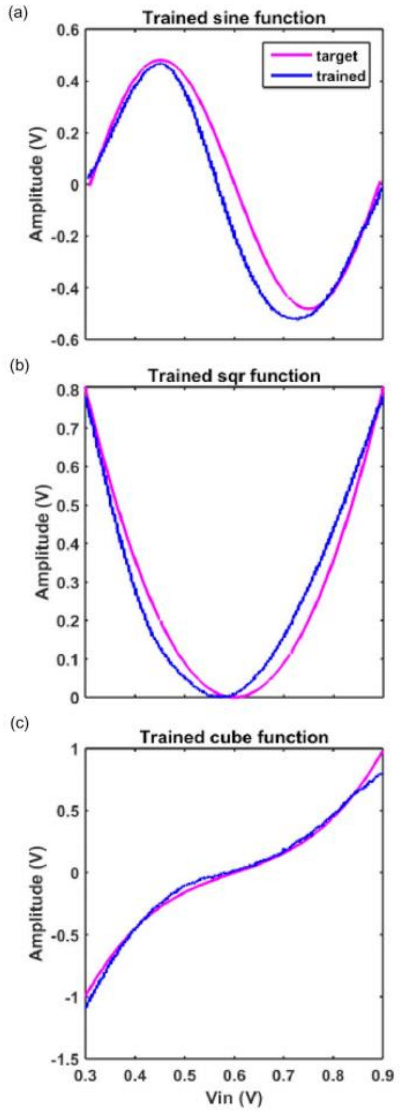
Extreme Learning Machines



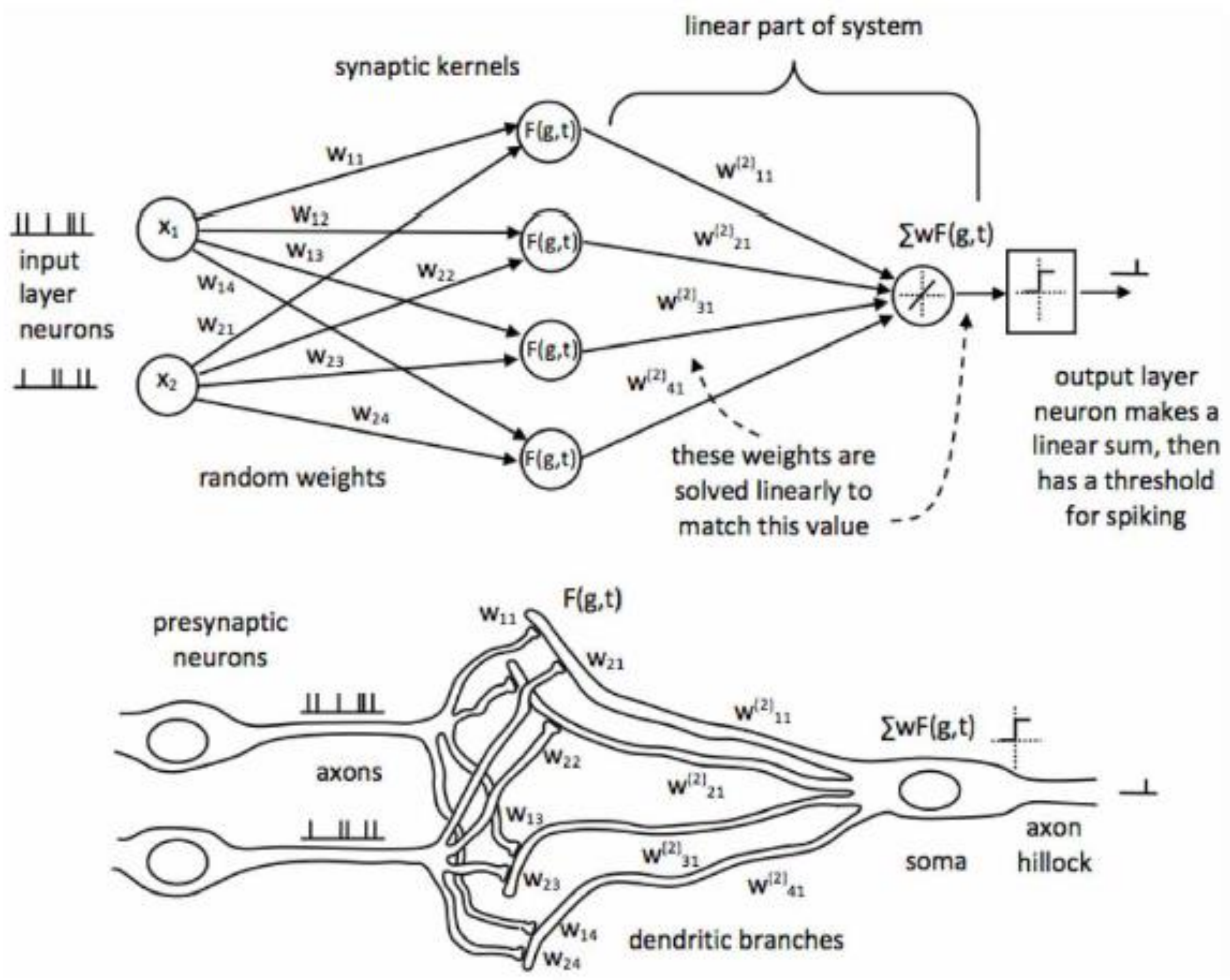
- $y_n = \sum_{i=1}^{L} w_i^{(2)} G\left(w_i^{(1)}, b_i^{(1)}, x_n\right) \quad \forall_n = 1,2,\dots M$
- $HW^{(2)} = Y$
- $H_{MxL} = \begin{Bmatrix} G\left(w_1^{(1)}, b_1, x_1\right) \dots\dots G\left(w_L^{(1)}, b_L, x_1\right) \\ \vdots \qquad\qquad \vdots \\ \vdots \qquad\qquad \vdots \\ \vdots \qquad\qquad \vdots \\ G\left(w_1^{(1)}, b_1, x_M\right) \dots\dots G\left(w_L^{(1)}, b_L, x_M\right) \end{Bmatrix}$
- $W^{(2)}{}_{LxK} = \begin{Bmatrix} w_1^{(2)} \\ \vdots \\ \vdots \\ \vdots \\ w_L^{(2)} \end{Bmatrix}, \; Y_{MxK} = \begin{Bmatrix} y_1 \\ \vdots \\ \vdots \\ \vdots \\ y_M \end{Bmatrix}$
- Calculate : $W^{(2)} = H^{+}Y$

Trainable Analog Blocks

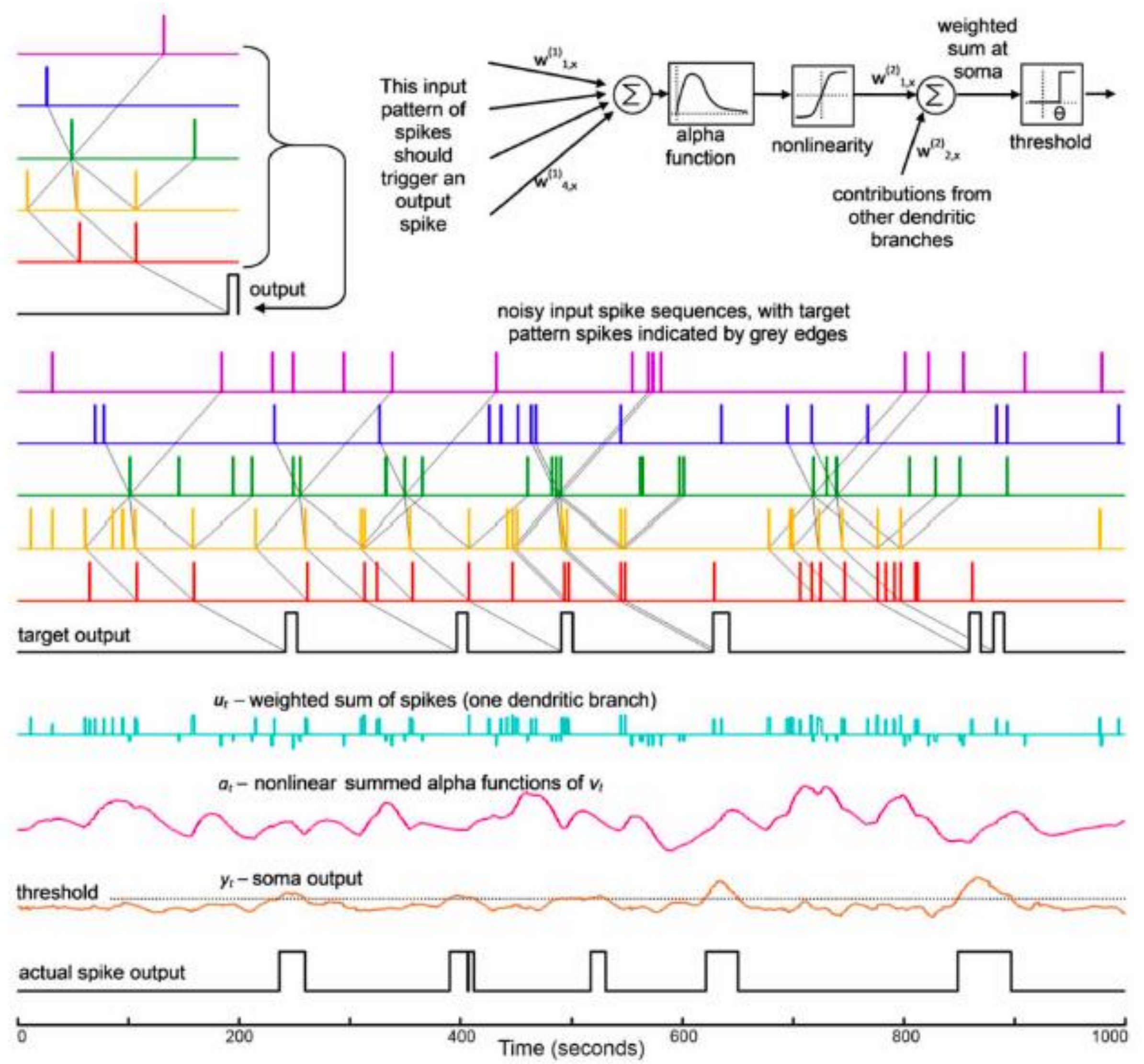




Synaptic Kernel Inverse Method (SKIM)



Synaptic Kernel Inverse Method (SKIM)



output
This input pattern of spikes should trigger an output spike
$w^{(1)}_{1,x}$
$w^{(1)}_{4,x}$
alpha function
nonlinearity
$w^{(2)}_{1,x}$
$w^{(2)}_{2,x}$
weighted sum at soma
threshold
θ
contributions from other dendritic branches
noisy input spike sequences, with target pattern spikes indicated by grey edges
target output
u_t – weighted sum of spikes (one dendritic branch)
a_t – nonlinear summed alpha functions of v_t
threshold
y_t – soma output
actual spike output
0
200
400
600
800
1000
Time (seconds)