Introduction to Neuroinformatics
Neuromorphic VLSI

Giacomo Indiveri

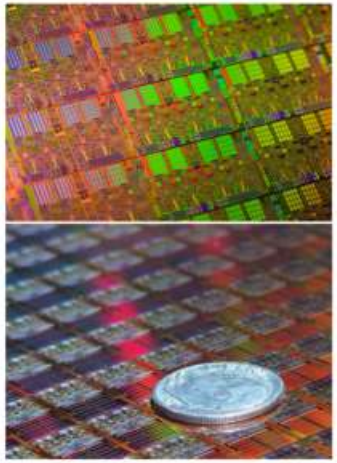
Neuromorphic Cognitive Systems
Institute of Neuroinformatics
University of Zurich and ETH Zurich

December 19, 2019

Outline

1. Introduction to neuromorphic VLSI
2. Neuromorphic subthreshold circuits
3. The differential pair circuit
4. Silicon neurons
5. Neuromorphic processors
6. How to program neuromorphic processors
7. Learning
8. Computational primitives
9. Neural State Machines
10. Conclusions

VLSI technology



Very Large Scale Integration technology
The technology that allows us to fabricate processor chips and memories. Today's computers, built using digital VLSI circuits, are:

- not analog
- not low power
- not fault tolerant
- not robust to inhomogeneities
- not asynchronous
- not massively parallel

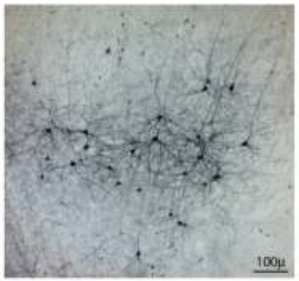
The term "neuromorphic"

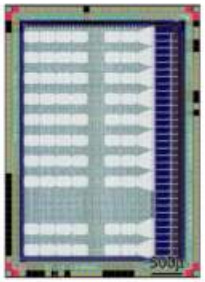
The term neuromorphic was coined by Carver Mead in the late '80s to describe VLSI systems containing electronic analog/digital circuits that *exploit the physics of silicon to reproduce the bio-physics of neural* circuits present in the nervous system.

It is a discipline characterized by two main goals.

1. To understand the computational properties of biological neural systems using standard CMOS VLSI technology as a tool.
2. To exploit the known properties of biological systems to design and implement efficient devices for engineering applications.

Neuromorphic VLSI neuron circuits





Nuno da Costa, INI, 2008

Goals:

- to reproduce the physics of neural computation using subthreshold analog circuits and asynchronous digital circuits.
- to build autonomous learning behaving systems that can interact with the environment in real–time.

Neuromorphic Engineering

The origins (late 1970s)

Carver Mead
(Caltech)

Max Delbruck
(Caltech)

Moshe Eisenberg (from UPenn)
Jim Hall (recalled early from Vietnam)
Peter Leuger (Konstanz)
Fred Sigworth

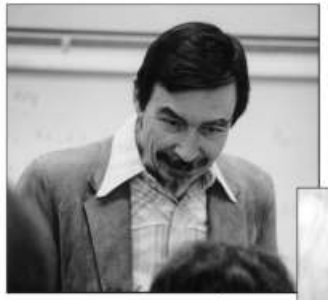
Paul Mueller
(UPenn)

Biophysics of membrane channels

Neuromorphic Engineering

Deeply rooted in biology . . .

1980s

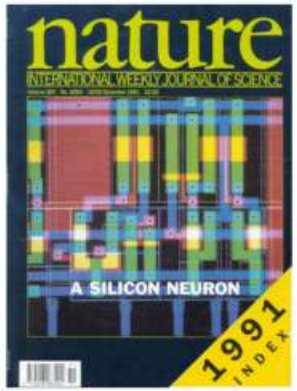


Carver Mead

Misha Mahowald

A silicon neuron

The '90s

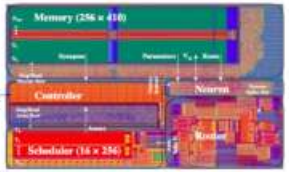


In 1991 Misha Mahowald and Rodney Douglas proposed a conductance-based silicon neuron and showed that it had properties remarkably similar to those of real cortical neurons.

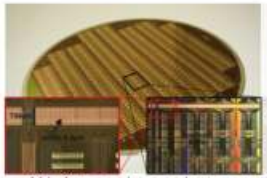
Neuromorphic computing today

ARM-based simulated spiking neural networks (SpiNNaKer).

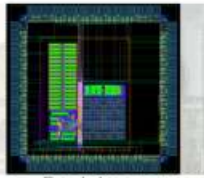
Asynchronous simulation of neurons and synapses (Loihi).



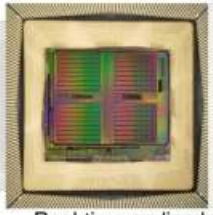
Large-scale asynchronous spiking neural network (TrueNorth).



Wafer-scale analog neural accelerators (BrainScaleS).



Real-time neuromorphic large-scale system (BrainDrop).



Real-time on-line learning neuromorphic chips (DYNAPs)

Neuromorphic computing

Basic research

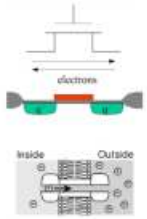
- Fundamental research.
- Emulation of neural function.
- Subthreshold analog
- Asynchronous digital.

Recent developments

- Dedicated VLSI hardware.
- High performance computing.
- Application driven.
- Conservative approaches.

The neuromorphic engineering approach

Learn to build artificial neural processing systems that can interact intelligently with the physical world



1. Combine multiple disciplines (neuroscience physics, computer science, electrical engineering,. . .)
2. Exploit device physics to directly emulate the biophysics of neural systems.
3. Let time represent itself.
4. Implement robust computation in autonomous agents that produce cognitive behavior.

Channel current-voltage relationships

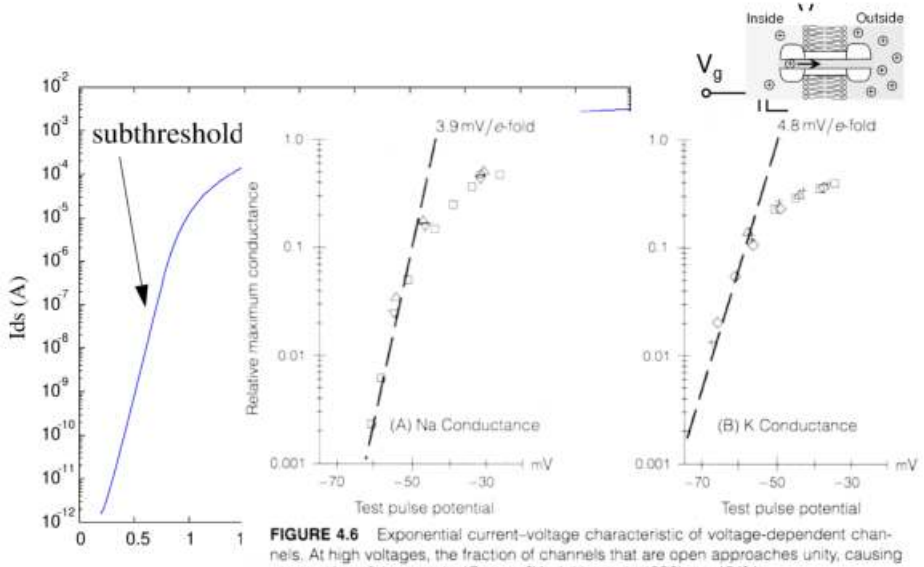
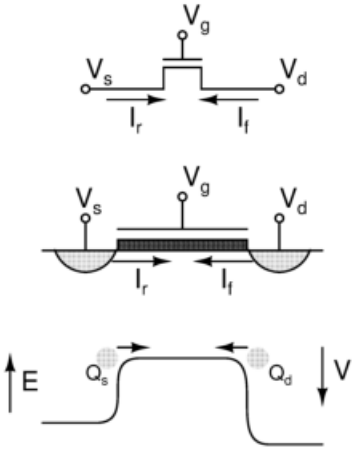


FIGURE 4.6 Exponential current–voltage characteristic of voltage-dependent channels. At high voltages, the fraction of channels that are open approaches unity, causing a saturation of the curves. (*Source:* [Hodgkin et al., 1952b, p. 464].)

Diffusion and saturation



$$I_{ds} = I_0 e^{\kappa_n V_g/U_T}\left(e^{-V_s/U_T} - e^{-V_d/U_T}\right)$$

is equivalent to:

$$\begin{aligned} I_{ds} &= I_0 e^{\kappa \frac{V_g}{U_T} - \frac{V_s}{U_T}} - I_0 e^{\kappa \frac{V_g}{U_T} - \frac{V_d}{U_T}} \\ I_{ds} &= I_f - I_r \end{aligned}$$

If $V_{ds} > 4U_T$ the I_r term becomes negligible, and the transistor is said to operate in the saturation regime:

$$I_{ds} = I_0 e^{\kappa_n V_g/U_T - V_s/U_T}$$

Body Effect

What is body effect?

Subthreshold

In subthreshold, for a constant I, a ΔV change in the source voltage means that the gate voltage has to increase by $\kappa \Delta V$ and not just ΔV.

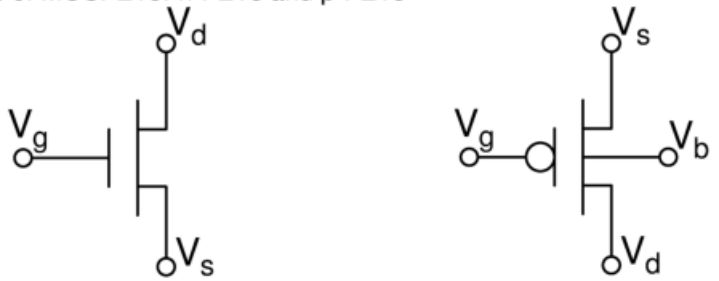
Above threshold

In above threshold, this effect is often taken to mean that the threshold voltage of the transistor increases with the source voltage.

$$\kappa = \frac{C_{OX}}{C_{OX} + C_{dep}}$$

n-FETs and p-FETs

In Complementary Metal-Oxide Semiconductor (CMOS) technology, there are two types of MOSFETs: n-FETs and p-FETs



In traditional CMOS circuits, all n-FETs have the common bulk potential (V_b) connected to Ground (Gnd), and all p-FETs have a common bulk potential (typically) connected to the power supply rail (V_{dd}).

Transistor Subthreshold Equations

nFET

$$I = I_{n0} e^{\kappa_n V_g/U_T}\left(e^{-V_s/U_T} - e^{-V_d/U_T}\right)$$

pFET

$$I = I_{p0} e^{\kappa_p (V_{dd} - V_g)/U_T}\left(e^{-(V_{dd} - V_s)/U_T} - e^{-(V_{dd} - V_d)/U_T}\right)$$

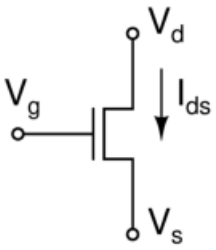
where

- I_{n0} and I_{p0} denote the nFET/pFET current-scaling parameter
- κ_n and κ_p denote the nFET/pFET subthreshold slope factor
- U_T the thermal voltage
- V_g the gate voltage, V_s the source voltage, and V_d the drain voltage.

The current is defined to be positive if it flows from the drain to the source.

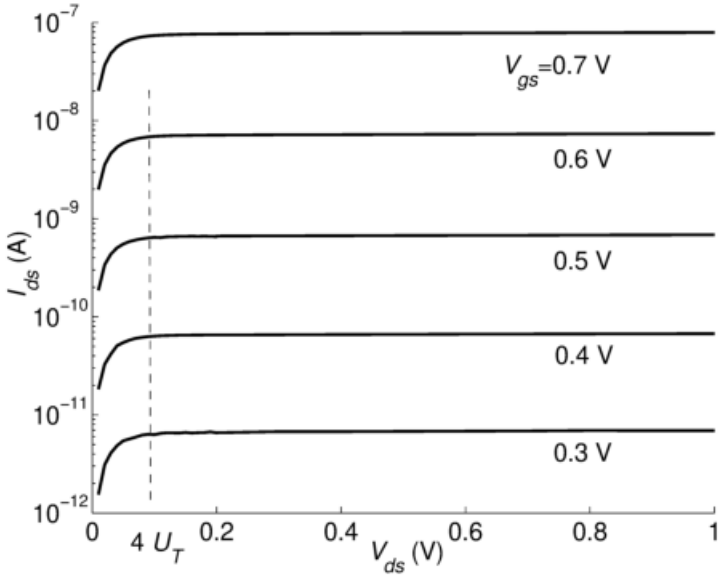
Current Source

If we can neglect the Early effect (use long transistors)

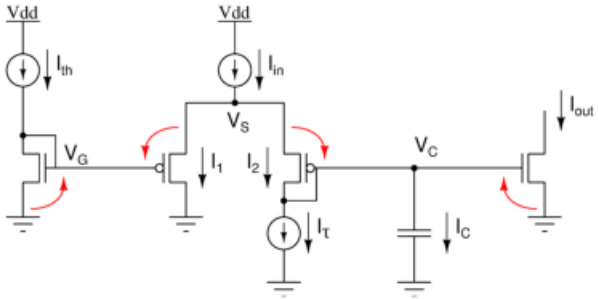


$$I = I_{n0} e^{(\kappa_n V_g - V_s)/U_T} \text{ for } |V_d - V_s| \geq 100mV$$

I_d vs V_{ds}



A current-mode differential-pair integrator (DPI)



$$I_{th} \cdot I_1 = I_2 \cdot I_{out}$$

$$I_{th} \cdot (I_{in} - I_\tau - I_C) = (I_\tau + I_C) \cdot I_{out}$$

$$\tau \left(1 + \frac{I_{th}}{I_{out}}\right) \frac{d}{dt} I_{out} + I_{out} = \frac{I_{th} I_{in}}{I_\tau} - I_{th}$$

$$I_{out} = I_0 e^{\frac{\kappa V_C}{U_T}}$$

$$I_1 + I_2 = I_{in}$$

$$I_2 = I_\tau + I_C$$

$$I_C = C \frac{d}{dt} V_C$$

$$I_C = C \frac{U_T}{\kappa I_{out}} \frac{d}{dt} I_{out}$$

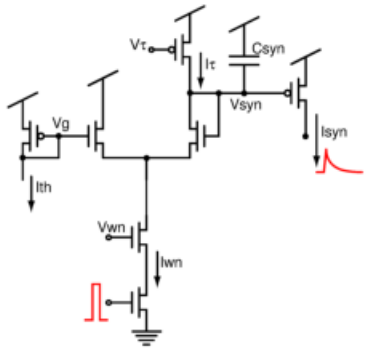
$$\tau = \frac{C U_T}{\kappa I_\tau}$$

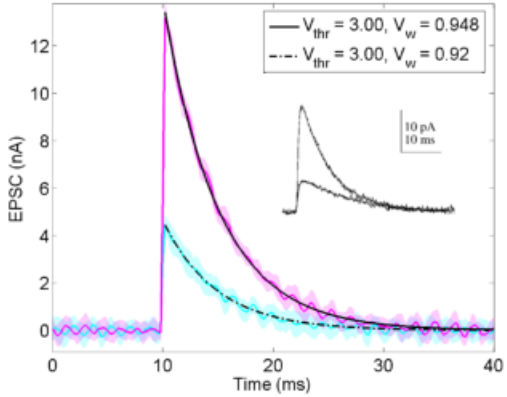
$$\text{if } I_{in} \gg I_\tau$$

$$\tau \frac{d}{dt} I_{out} + I_{out} = \frac{I_{th}}{I_\tau} I_{in}$$

Silicon synapses

The Differential-Pair Integrator synapse



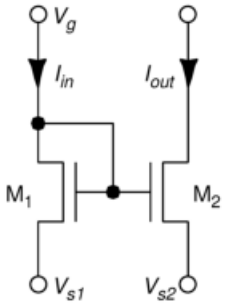


$$\tau \frac{d}{dt} I_{syn} + I_{syn} = \frac{I_{thr} I_{wn}}{I_\tau}$$

The diff-pair integrator (DPI) circuit [Bartolozzi and Indiveri, Neural Computation, 2007]

The Current Mirror

The output current is a *mirrored* copy of the input current.



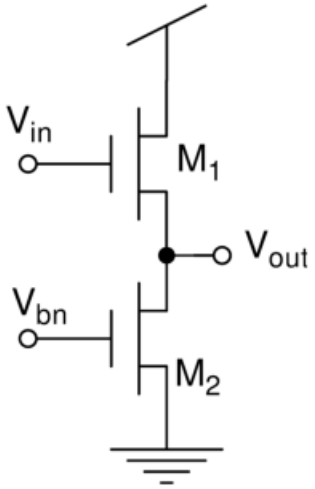
If both MOSFETs are of the same size and have the same source voltage, they source the same current, which is why the device is called a *current mirror*. The input current I_{in} through the diode-connected transistor M_1 sets the common gate voltage V_g and hence the output current I_{out} of the second transistor M_2.

The output current can be scaled by choosing different transistor sizes, or by choosing different source potentials V_{s1} and V_{s2} for the two MOSFETs.
If M_2 is in saturation:

$$I_{out} = e^{(V_{s1}-V_{s2})/U_T} I_{in}.$$

The Source Follower

n-type



$$I_{M1} = I_0 e^{\kappa V_{in}/U_T - V_{out}/U_T}$$
$$I_{M2} = I_0 e^{\kappa V_{bn}/U_T}$$
$$I_{M1} = I_{M2}$$

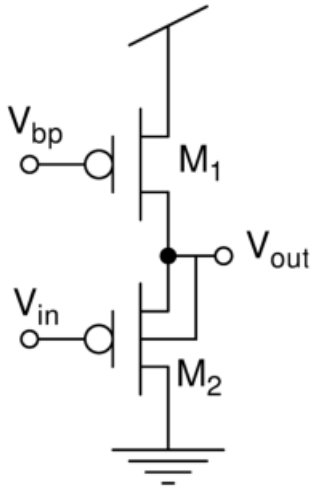
$$\boxed{V_{out} = \kappa(V_{in} - V_{bn})}$$

Saturation condition for `M2`:

$$V_{out} > 4U_T$$

The Source Follower

p-type



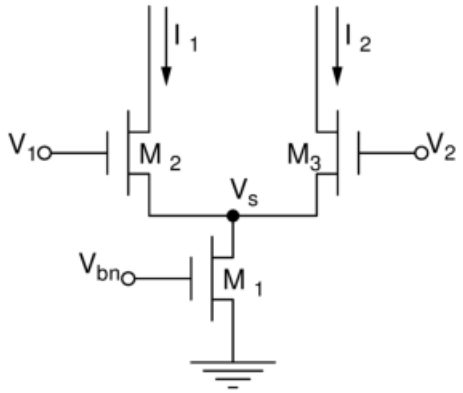
$$I_{M1} = I_0 e^{-\kappa(V_{bp}-V_{dd})/U_T}$$
$$I_{M2} = I_0 e^{-\kappa(V_{in}-V_{out})/U_T}$$
$$I_{M1} = I_{M2}$$

$$\boxed{V_{out} = (V_{dd} - V_{bp}) + V_{in}}$$

Saturation condition for `M1`:

$$V_{out} < V_{dd} - 4U_T$$

The differential pair



- Input signal: $\Delta V = V_1 - V_2$
- Output signals: I_1 and I_2, if saturated.
- Bias parameter: V_b

The differential-pair

$$I_1 = I_0 e^{\frac{\kappa V_1 - V_s}{U_T}}$$

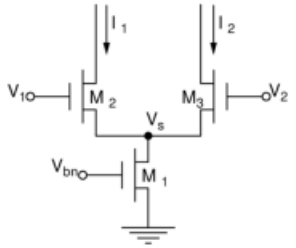
$$I_2 = I_0 e^{\frac{\kappa V_2 - V_s}{U_T}}$$

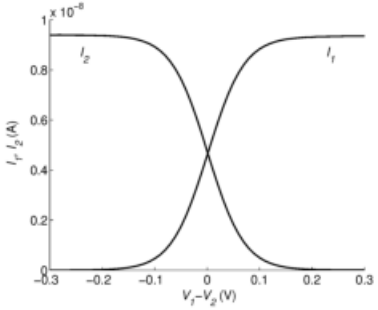
$$I_b = I_1 + I_2 = I_0 e^{\frac{\kappa V_b}{U_T}}$$

$$e^{-\frac{V_s}{U_T}} = \frac{I_b}{I_0} \frac{1}{e^{\frac{\kappa V_1}{U_T}} + e^{\frac{\kappa V_2}{U_T}}}$$

$$I_1 = I_b \frac{e^{\frac{\kappa V_1}{U_T}}}{e^{\frac{\kappa V_1}{U_T}} + e^{\frac{\kappa V_2}{U_T}}}$$

$$I_2 = I_b \frac{e^{\frac{\kappa V_1}{U_T}}}{e^{\frac{\kappa V_1}{U_T}} + e^{\frac{\kappa V_2}{U_T}}}$$





Sigmoids (contd)

Diff-pair output currents

The output currents of the diff-pair can be rewritten in the canonical sigmoid form:

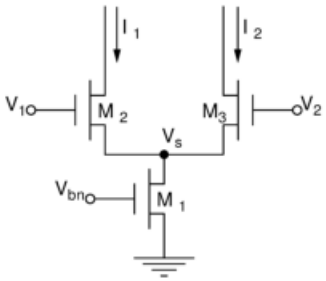
$$I_1 = I_b \frac{1}{1 + e^{\frac{\kappa}{U_T}(V_2 - V_1)}} \qquad I_2 = I_b \frac{1}{1 + e^{\frac{\kappa}{U_T}(V_1 - V_2)}}$$

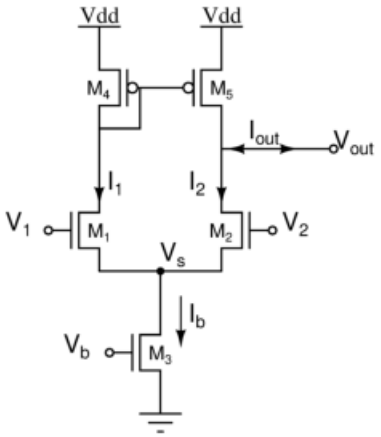
Difference of diff-pair currents

$$I_1 - I_2 = I_b \frac{e^{\frac{\kappa V_1}{U_T}} - e^{\frac{\kappa V_2}{U_T}}}{e^{\frac{\kappa V_1}{U_T}} + e^{\frac{\kappa V_2}{U_T}}} = I_b \tanh\left(\frac{\kappa}{2U_T}(V_1 - V_2)\right)$$

Difference of currents

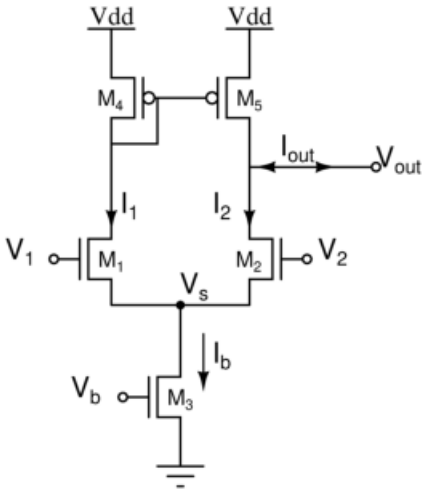
To implement the difference of currents $(I_1 - I_2)$ we can use ...

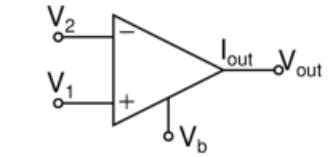




... a current-mirror

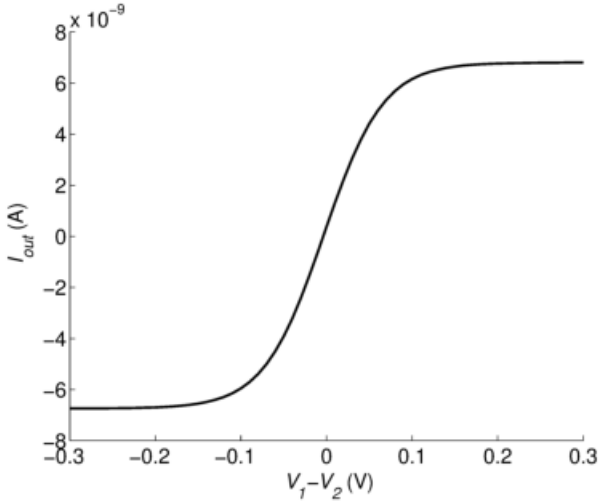
The Transconductance Amplifier





$$I_{out} = I_b \tanh\left(\frac{\kappa}{2U_T}(V_1 - V_2)\right)$$

The Transconductance Amplifier



The Transconductance Amplifier

For small differential voltages (*e.g.* $|V_1 - V_2| < 200mV$), the $\tanh(\cdot)$ relationship is approximately linear and the equation

$$I_{out} = I_b \tanh\left(\frac{\kappa}{2U_T}(V_1 - V_2)\right)$$

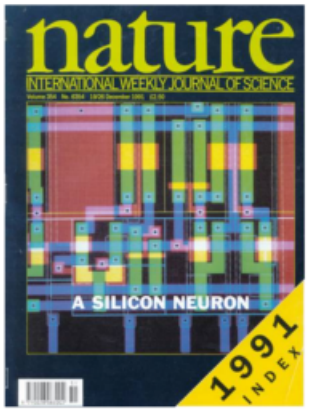
can be reduce to:

$$I_{out} \approx g_m(V_1 - V_2)$$

where

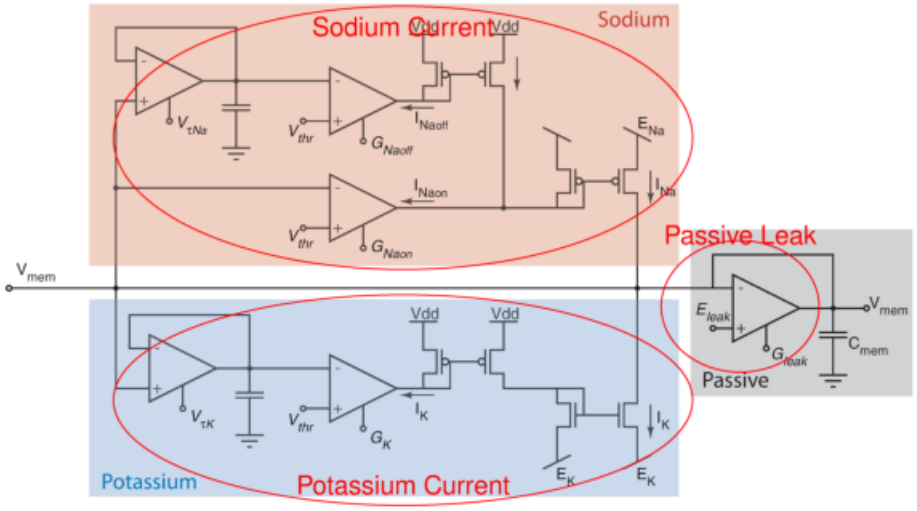
$$\boxed{g_m = \frac{I_b \kappa}{2U_T}}$$

A conductance-based silicon neuron

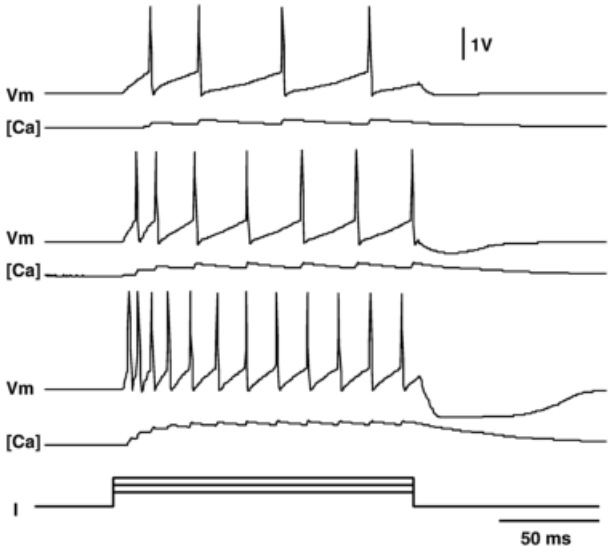


In 1991 Misha Mahowald and Rodney Douglas proposed a conductance-based silicon neuron and showed that it had properties remarkably similar to those of real cortical neurons.

The conductance based Si-Neuron



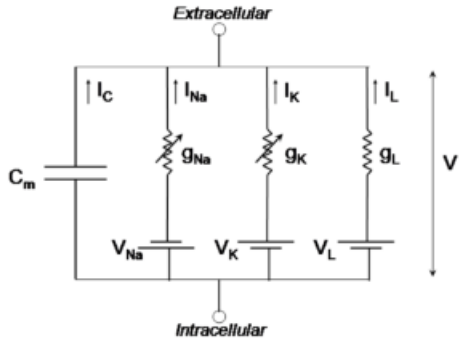
Silicon neuron's measurements



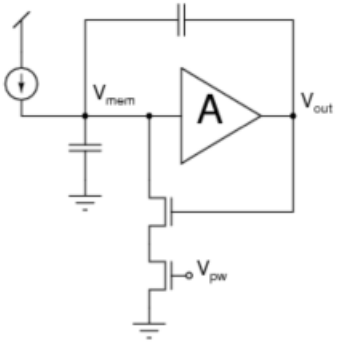
Conductance-based models

Integrate and Fire vs Hodgkin-Huxley

Traditionally there have been two main classes of neuron models:



Conductance-based (R-C)



Integrate and fire (I-C)

Conductance-based models

Integrate and Fire vs Hodgkin-Huxley

But recently proposed models bridge the gap between the two:

J Neurophysiol 92: 959–976, 2004;
10.1152/jn.00190.2004.

Generalized Integrate-and-Fire Models of Neuronal Activity Approximate Spike Trains of a Detailed Model to a High Degree of Accuracy

Renaud Jolivet,[1,*] Timothy J. Lewis,[2,*] and Wulfram Gerstner[1,*]

J Neurophysiol 99: 656–666, 2008.
First published December 5, 2007; doi:10.1152/jn.01107.2007.

Dynamic *I-V* Curves Are Reliable Predictors of Naturalistic Pyramidal-Neuron Voltage Traces

Laurent Badel,[1] Sandrine Lefort,[2] Romain Brette,[3] Carl C. H. Petersen,[2] Wulfram Gerstner,[1] and Magnus J. E. Richardson[1,4]

Biol Cybern (2008) 99:361–370
DOI 10.1007/s00422-008-0259-4

ORIGINAL PAPER

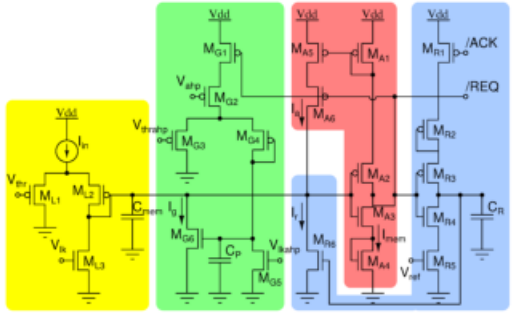
Biological Cybernetics

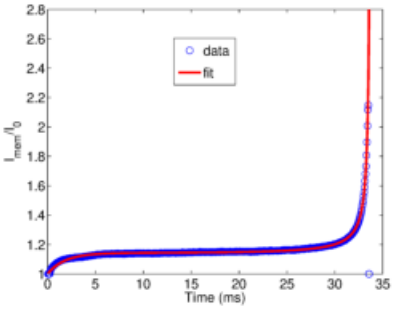
Extracting non-linear integrate-and-fire models from experimental data using dynamic *I–V* curves

Laurent Badel · Sandrine Lefort · Thomas K. Berger · Carl C. H. Petersen · Wulfram Gerstner · Magnus J. E. Richardson

Silicon neurons

The low power I&F neuron





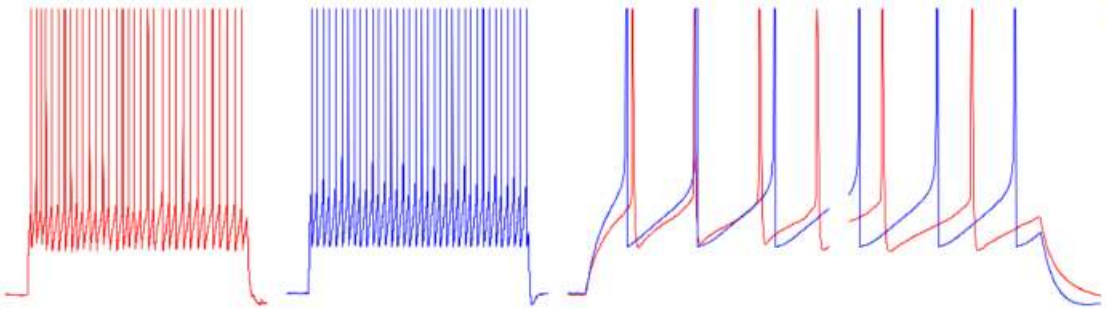
$$\tau \frac{d}{dt} I_{mem} + I_{mem} \approx \frac{I_{th} I_{in}}{I_\tau} - I_g + f(I_{mem})$$

$$\tau_{ahp} \frac{d}{dt} I_g + I_g = \frac{I_{thr} I_{ahp}}{I_{\tau_{ahp}}}$$

[Indiveri et al., 2010] [Brette and Gerstner, 2005]

Model neurons

The adaptive exponential I&F neuron model

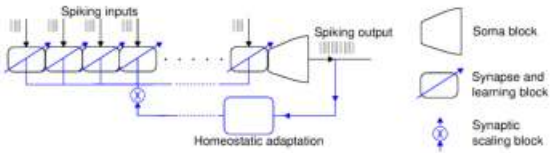


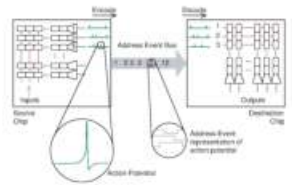
$$C\frac{d}{dt}V + g_L(V - E_L) = I - w + f(V)$$

$$\tau_w \frac{d}{dt}w + w = a(V - E_L)$$

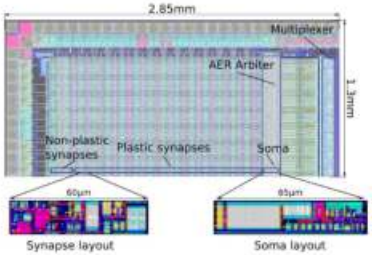
[Brette and Gerstner, 2005]

Neuromorphic Processors

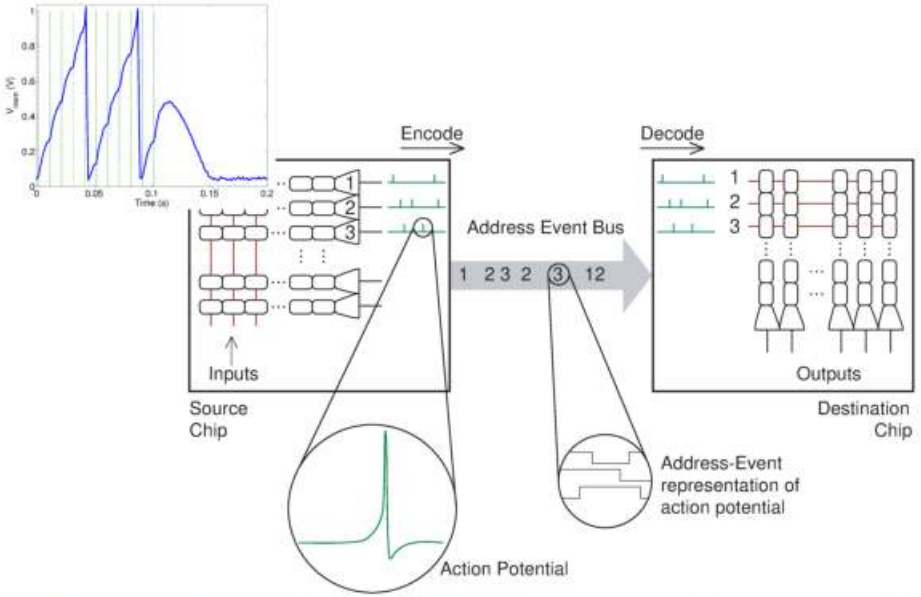




- Analog subthreshold circuits.
- Slow temporal, non-linear dynamics.
- Massively parallel operation.
- Inhomogeneous, imprecise, and noisy.
- Adaptation and learning at multiple time scales.
- Fast asynchronous digital routing circuits.
- Re-programmable network topology.
- Fault tolerant and mismatch insensitive by design.



Spikes and the Address-Event Representation



Neuromorphic processing systems

existence proof

Bee brain specs

weight:	1 mg
volume:	1 mm^3
# neurons:	960'000
energy/op:	10^{-15} J/spik

Neuromorphic agents

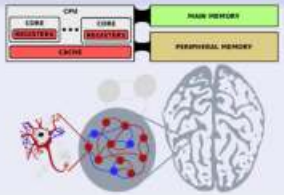
- Interact with the environment in real-time closed-loop settings
- Use both analog and digital computing elements.
- Exploit non-linearities and temporal dynamics.
- Leverage noise, variability, and stochasticity to achieve robust computation.
- Processing complex (dynamic and noisy) spatio-temporal signals.

Neuromorphic computing

A radical paradigm shift

Exploit physical space

- Multiple instances of similar computing elements.
- Memory and computation co-localized.
- Sparse activation, massive parallelsim.
- Continuous time. Data driven processing.



Let time represent itself

- For interacting with the environment in real-time.
- Dynamics with time constants matched to the input signals.
- Inherently synchronized with the real-world “natural” events.
- To process sensory signals efficiently (low power, low bandwidth).

Neuromorphic vs conventional processors

Pros

- Low latency
- Ultra low-power (**<1mW**)

Cons

- Limited resolution (<4bits)
- High variability, noisy

What are they good for?

- Real-time sensory-motor processing
- Sensory-fusion and on-line classification
- Low-latency decision making

What are they bad at?

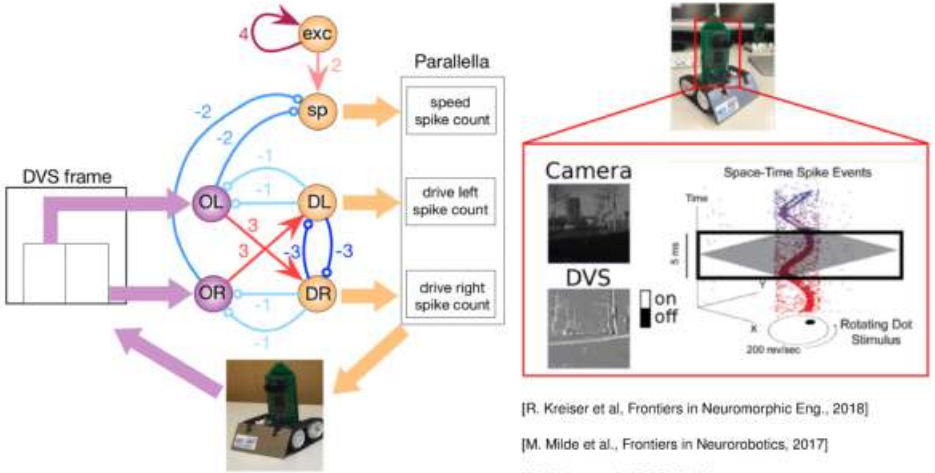
- High accuracy pattern recognition
- High precision number crunching
- Batch processing of data sets

How to program a neuromorphic processor?

- Configure network structure and parameters
- Train the network with different learning methods
- Define neural computational primitives
- Compose multiple primitives for context dependent processing

Configuring network and circuit parameters

to “program” robotic behavior

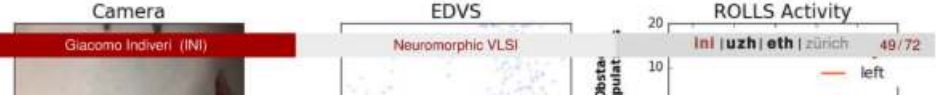


[R. Kreiser et al, Frontiers in Neuromorphic Eng., 2018]

[M. Milde et al., Frontiers in Neurorobotics, 2017]

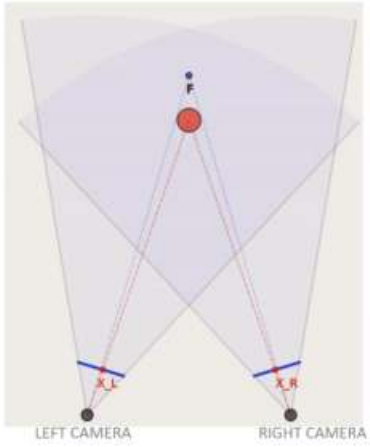
[R. Kreiser et al., IROS, 2018]

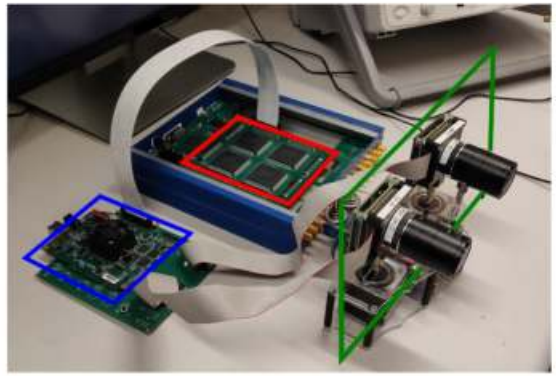
[S. Glatz et al., arXiv:1810.10801, 2018]



Configuring network and circuit parameters

to implement vergence control in active stereo vision setups

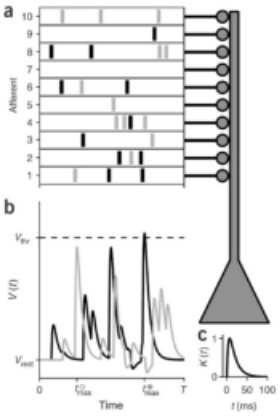




[Osswald and Indiveri, 2017][Nicoletta Risi, (in preparation)]

Learning: training the network

with on-chip on-line plasticity mechanism



Recent spike-driven learning algorithm

Spike-driven weight change depends on the timing of the pre-synaptic input, and on the value of the post-synaptic neuron's state variables.

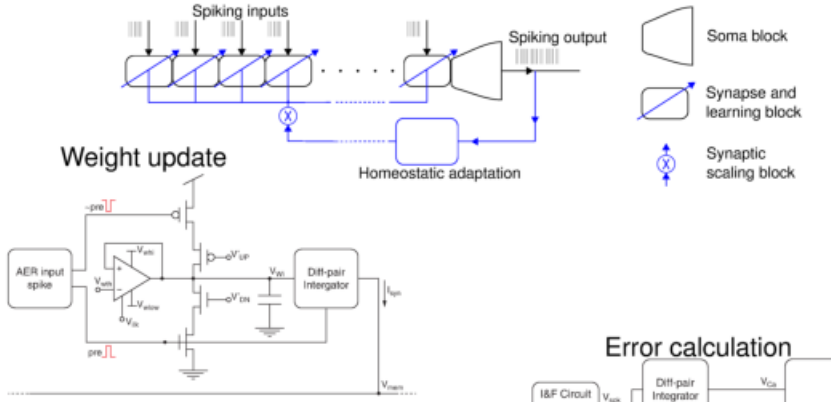
W. Senn, S. Fusi, N. Brunel, S. Sheik, E. Neftci, R. Zecchina, M. Memmesheimer, etc.

Requirements for efficient implementation

- bistability: use two synaptic states;
- redundancy: implement many synapses that see the same pre- and post-synaptic activity
- stochasticity & inhomogeneity: induce LTP/LTD only in a subset of stimulated synapses.

Spike-based learning circuits

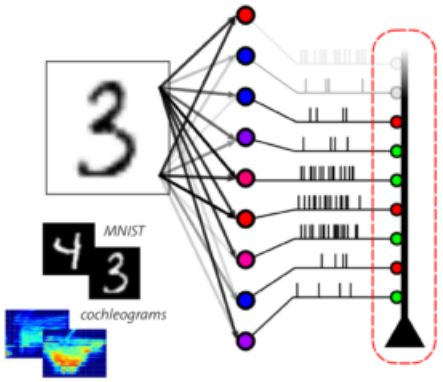
implementing third-factor and stochastic learning mechanisms

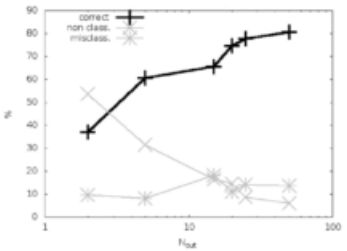


[Brader et al., 2007][Payvand and Indiveri, 2019]

Robust classification

to "program" classification/recognition behaviors





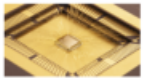
MNIST	deep/CNN (Hinton et al. 2012)	98.4%
	random + bistable synapses	~ 85%
	random + bistable synapses + (mod. protocol)	~ 96%
TIMIT	deep/CNN (Hinton et al. 2012)	77%
	VLSI cochlea + bistable synapses	~ 60%

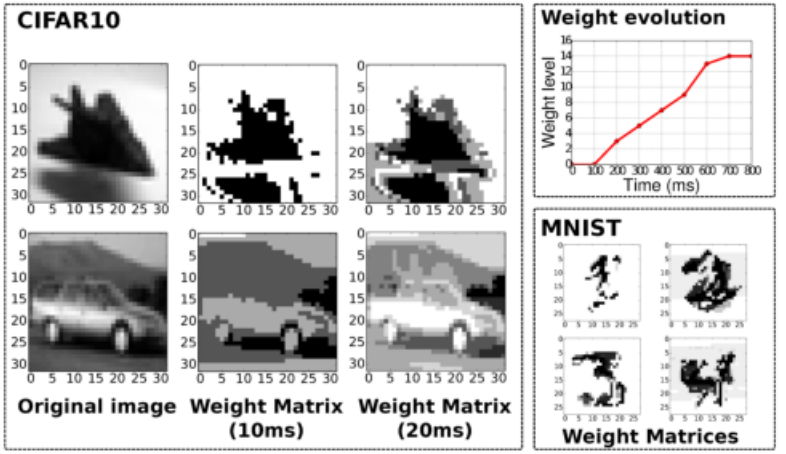
Ensemble learning techniques

Spike-based learning with inhomogeneous synapses exploits variability to enable an on-line bagging technique.

- AdaBoost theorem: $1-\text{error}(H_{final}) \geq 1-e^{-2\gamma^2 N}$ [Y. Freund And R. E. Schapire, 1995]

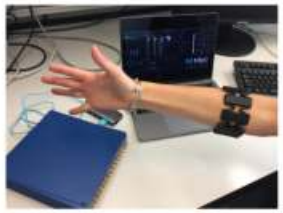
On-line on-chip spike-based learning

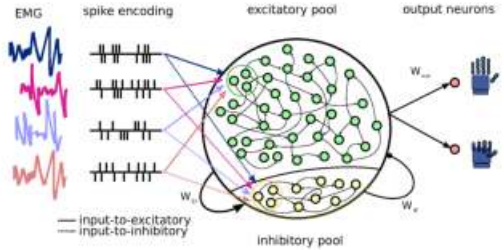


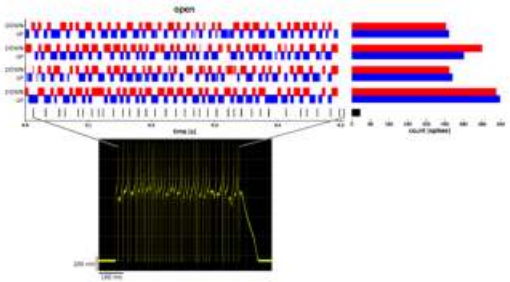


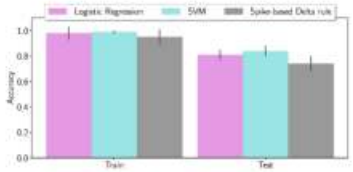
[Qiao et al, (in preparation)]

Learning to solve practical applications





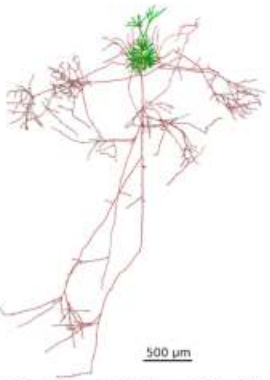




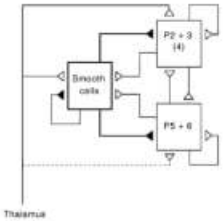
[Donati et al, 2019]

Canonical micro-circuits

to "program" state-dependent procedures



Pyramidal Cell of Layer 3 of Cat Visual Cortex Showing Dendrite (Green) and Axon (Red) Forming Multiple Clusters of Boutons (Black) in Layer 3 and 5.



Canonical Cortical Circuit Based on Electrophysiological and Modeling Studies in the Cat Visual Cortex (from [Douglas and Martin, 1989]).

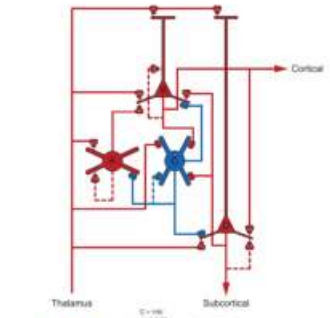
Winner-Take-All networks

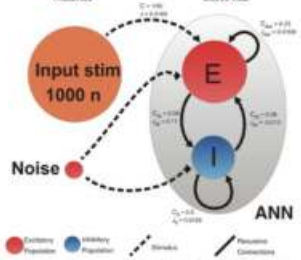
[Marcus et al., "The Atoms of Neural Computation", Science 2014]

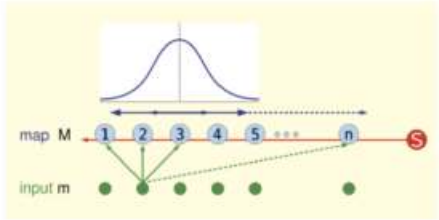
Hence we propose that the ubiquitous microcircuit motif [...] provides an important atomic computational operation to large-scale distributed brain computations.

[Jonke et al. J. Neurosci. 2017]

Winner-Take-All and Attractor networks

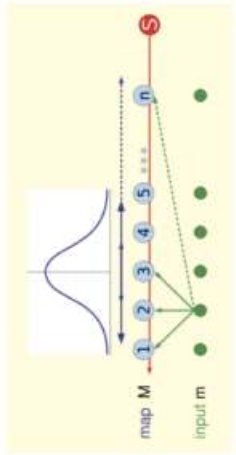


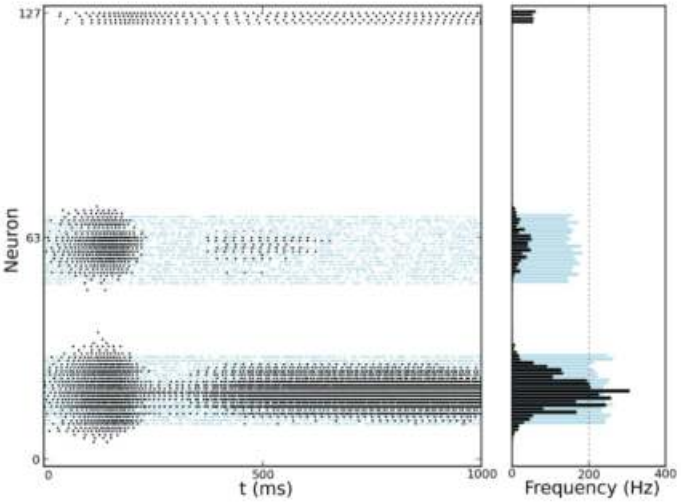




[Hahnloser et al, Science 2000] [Douglas and Martin, 2007, 2010] [Sandamriskaya, 2014] [Jonke et al. J. Neurosci. 2017]

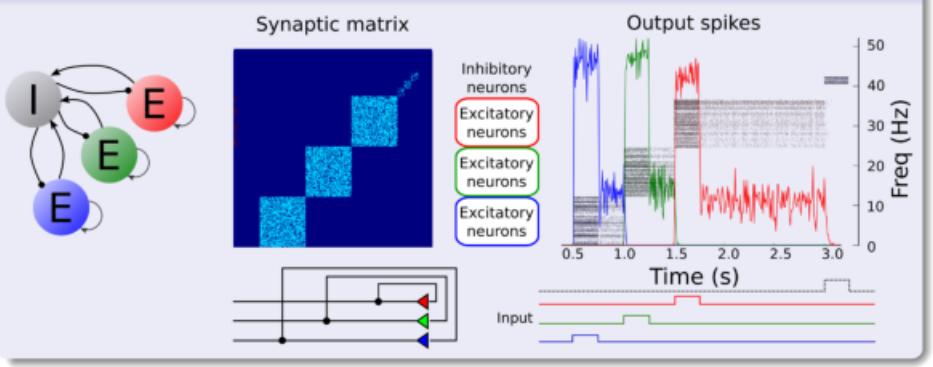
Winner-take-all networks in neuromorphic hardware





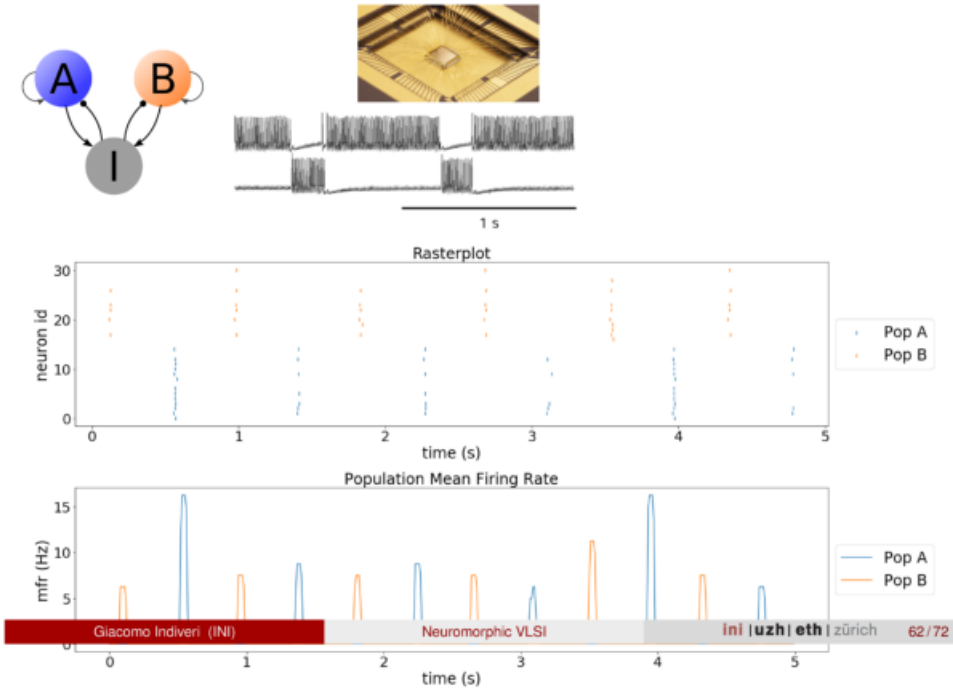
Learning and Winner-Take-All networks

Forming memories with attractor networks



[Indiveri Liu, 2017] [F. Corradi et al., 2014]

Intrinsic oscillators and Central Pattern Generators

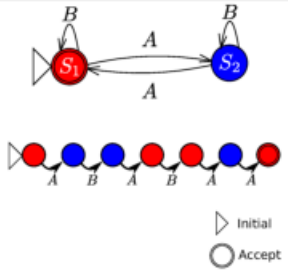


Neural State Machines (NSMs)

Robust computation with the Finite State Machine formalism

Finite State Machines

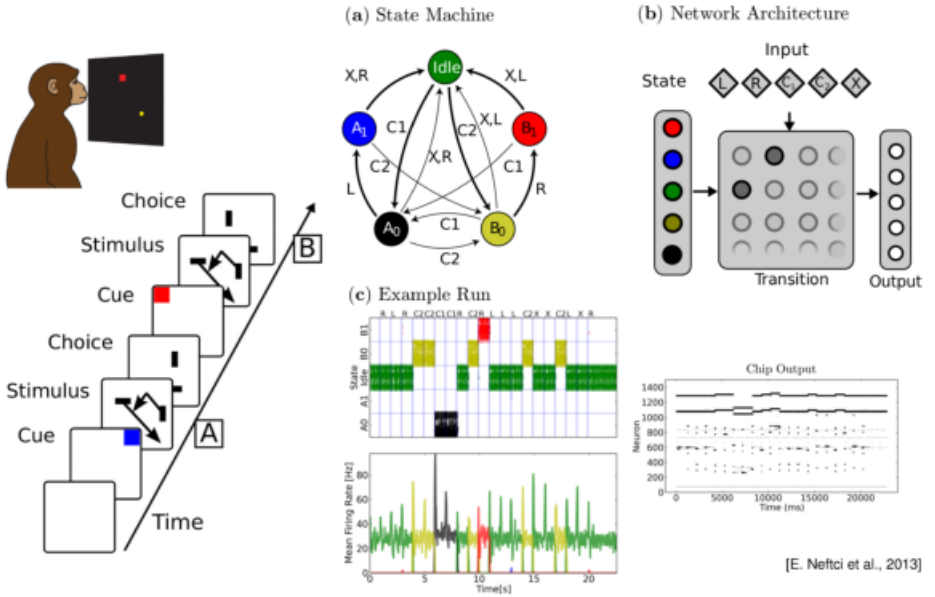
A finite-state machine (FSM) is a mathematical model of computation used to design both computer programs and sequential logic circuits. It is conceived as an abstract machine that can be in one of a finite number of states. [Wikipedia]



- Recognizes regular expression $B^*[AB^*A]^*$

Minsky, 1967

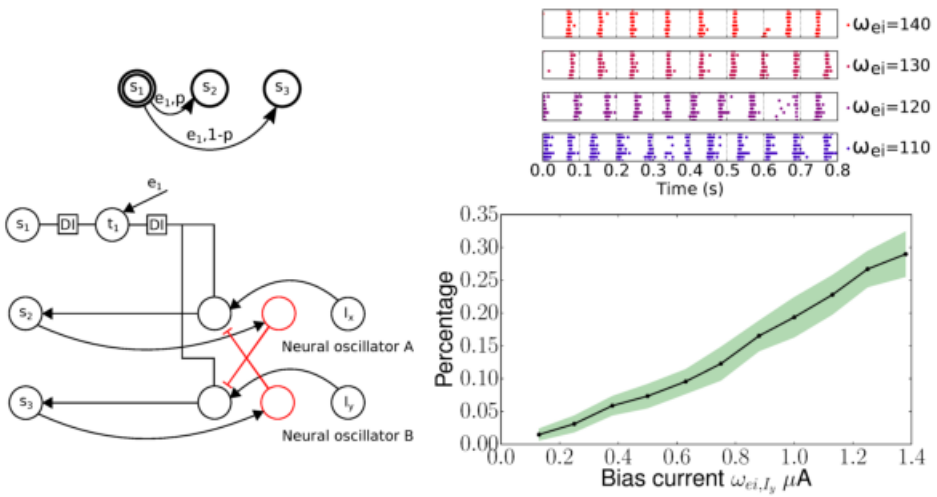
Synthesizing cognition using NSMs



[E. Neftci et al., 2013]

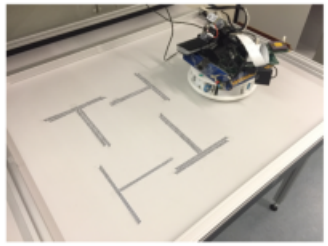
Probabilistic Neural State Machines

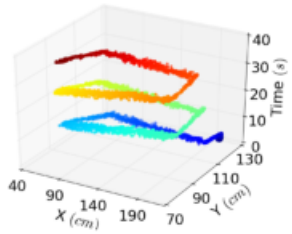
with intrinsic oscillators



[Mostafa et al 2015][Liang and Indiver, 2018]

Neuromorphic circuits for robust sensory-motor processing



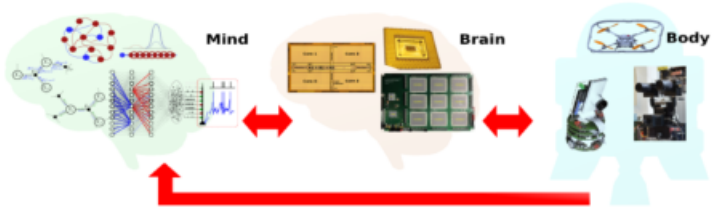


Promising results

By combining a small number of key computational primitive circuits with on-chip learning circuits existing and future mixed-signal neuromorphic processors can be used to implement:

- fast and robust visual, auditory, and multi-modal sensory processing
- adaptive motor control [Glaz et al. 2019]
- context dependent procedural tasks [Liang et al. 2019]
- map formation and pose estimation [Kreiser et al. 2018]
- on-line self-calibration of system parameters [Kreiser et al. 2019]

Conclusions



On-going research: iterative refinement

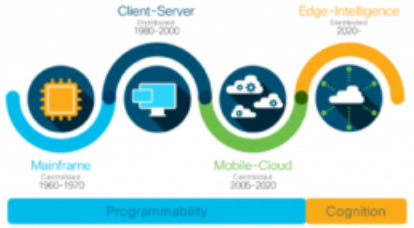
- We study the principles of computation of cortical circuits and validate them on neuromorphic systems that interact intelligently with the environment.
- We exploit progress in technology to develop mixed-signal neuromorphic electronic circuits for emulating neural dynamics and learning in real-time (2, 3 tape-outs/year, new learning mechanisms, new memory technologies, new architectures/protocols).
- We build analog/digital neural processing systems interfaced to sensors and robotic platforms that can (learn to) produce intelligent behavior.

The Future of Computing

for sustainable "big-data" processing

Neuromorphic computing application domains

We are now entering the era of *neuromorphic intelligence* in which dedicated task-specific "chiplets" will be used to provide intelligence to a multitude of edge-computing devices.



[https://techoverlook.com/]

- Intelligent "watchdogs"
- Auditory scene analysis
- Environmental sensing
- Prosthetic controllers
- Health monitoring
- Human body area networks

The end

Thank you for your attention

The CapoCaccia Workshops for Neuromorphic Intelligence

http://capocaccia.cc/

- Interdisciplinary, international, diverse
- Morning lectures, afternoon hands-on work-groups
- Active and lively discussions (no powerpoint)
- Concrete results, establishment of long-term collaborations

Capo Caccia, Sardinia, Italy. April 26 - May 9, 2020