Neuromorphic computing for robotics

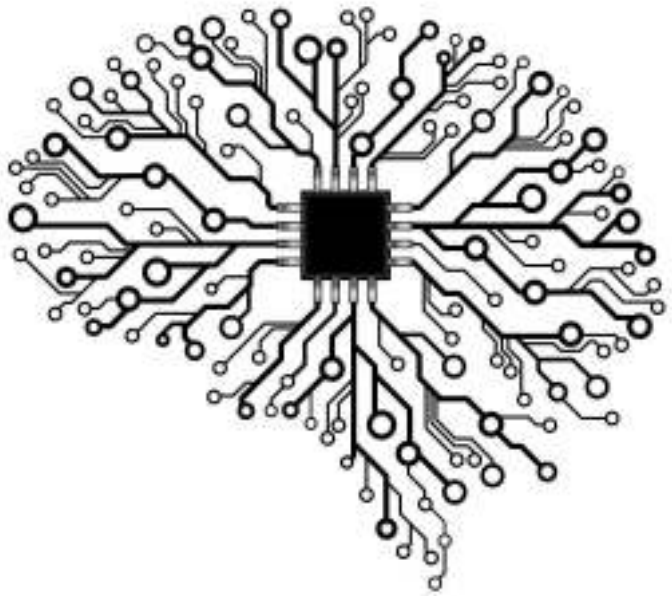
ROBIN lab meeting 26.10.2023

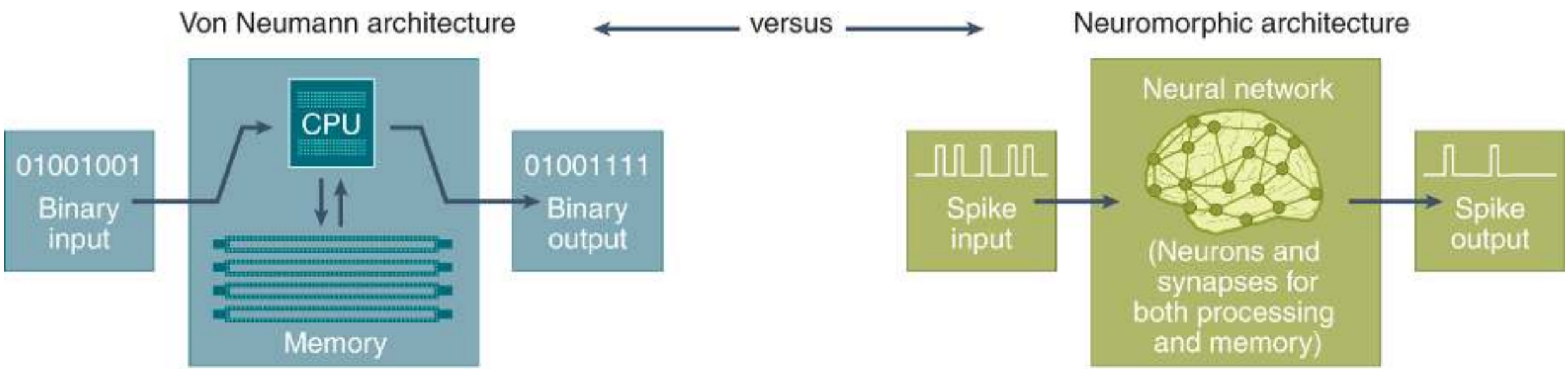
Plan for today

- What is it and why we should care?
- Parallels between neuromorphic chips and the brain
- Algorithms for neuromorphic hardware
  - Leaky integrate-and-fire neuron model
  - Spiking Neural Networks
  - Brian2 library
- Applications in robotics
- Akida BrainChip

What is it and why we should care?

- Moore’s law reaching its limits
- AI models growing more complex
- Biological systems are much more efficient
- Types of neuromorphic circuits
- Potential for robotics





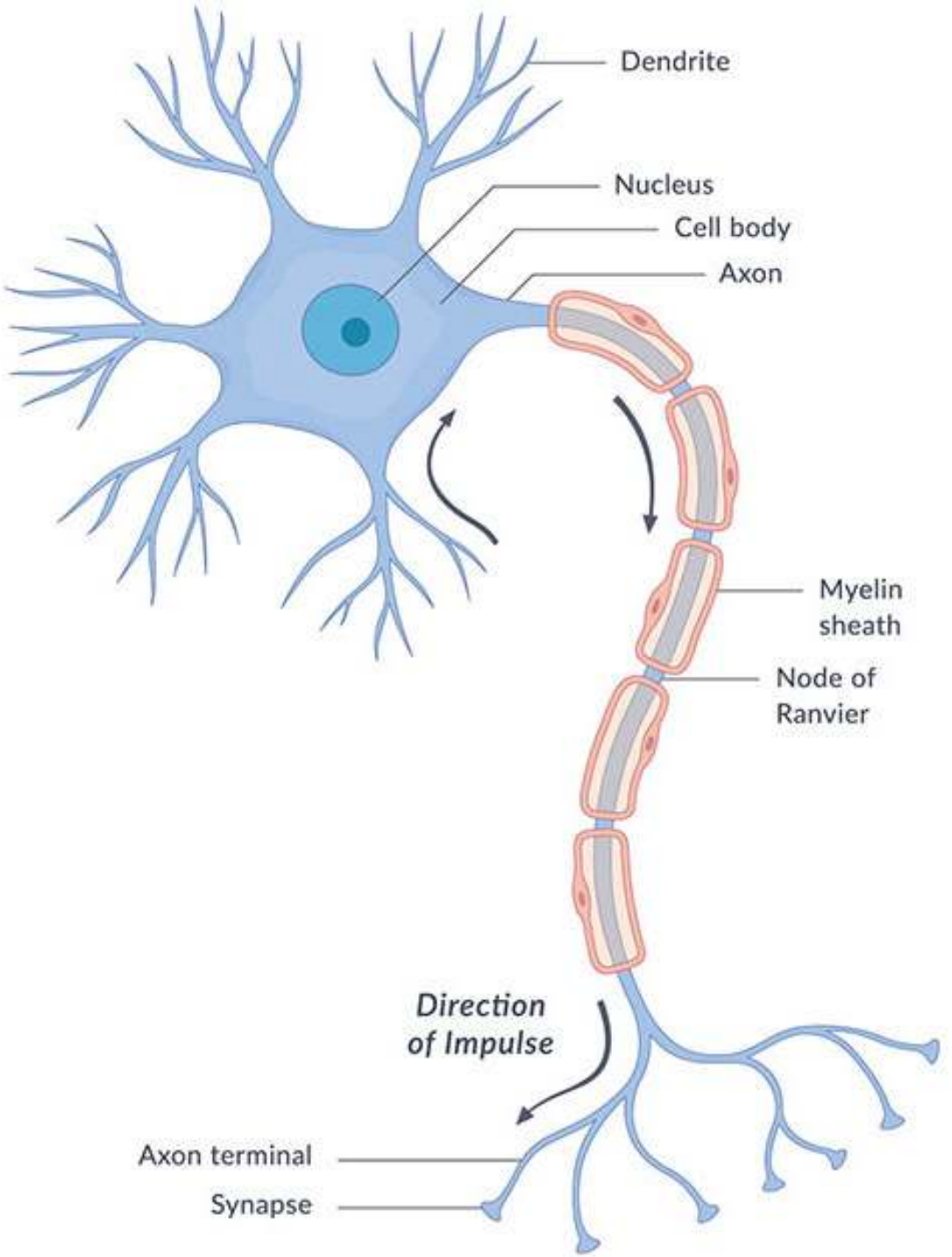
Sequential processing	Operation	Massively parallel processing
Separated computation and memory	Organization	Collocated processing and memory
Code as binary instructions	Programming	Spiking neural network
Binary data	Communication	Spikes
Synchronous (clock-driven)	Timing	Asynchronous (event-driven)

Biological neuron

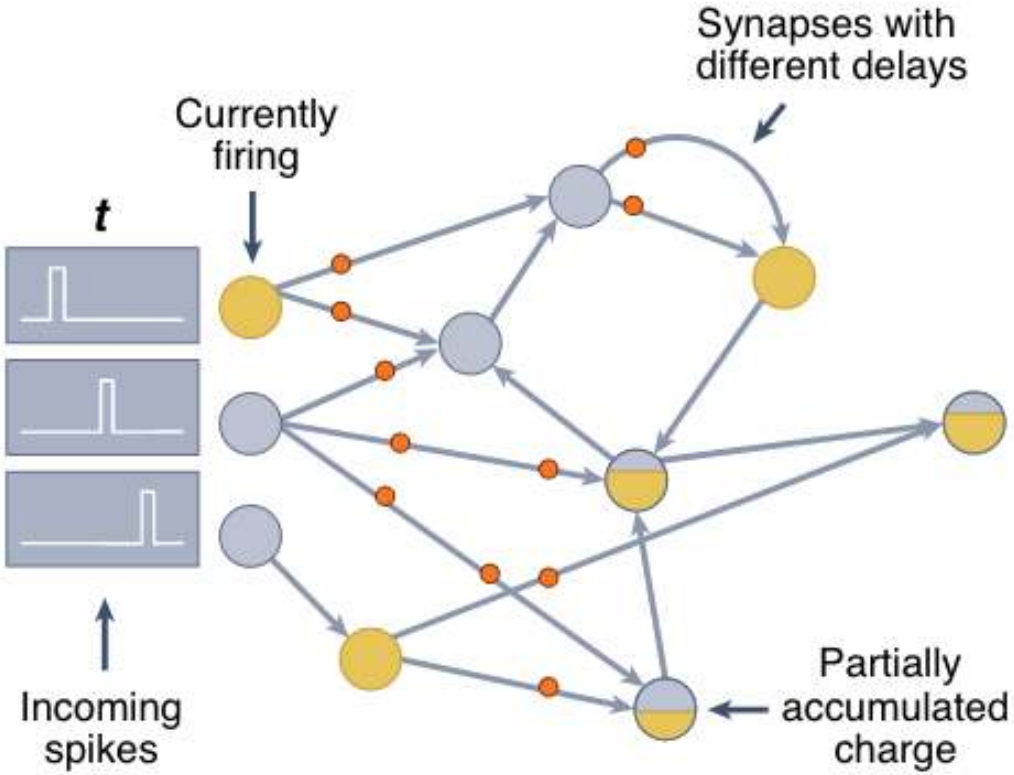
Leaky integrate-and-fire model

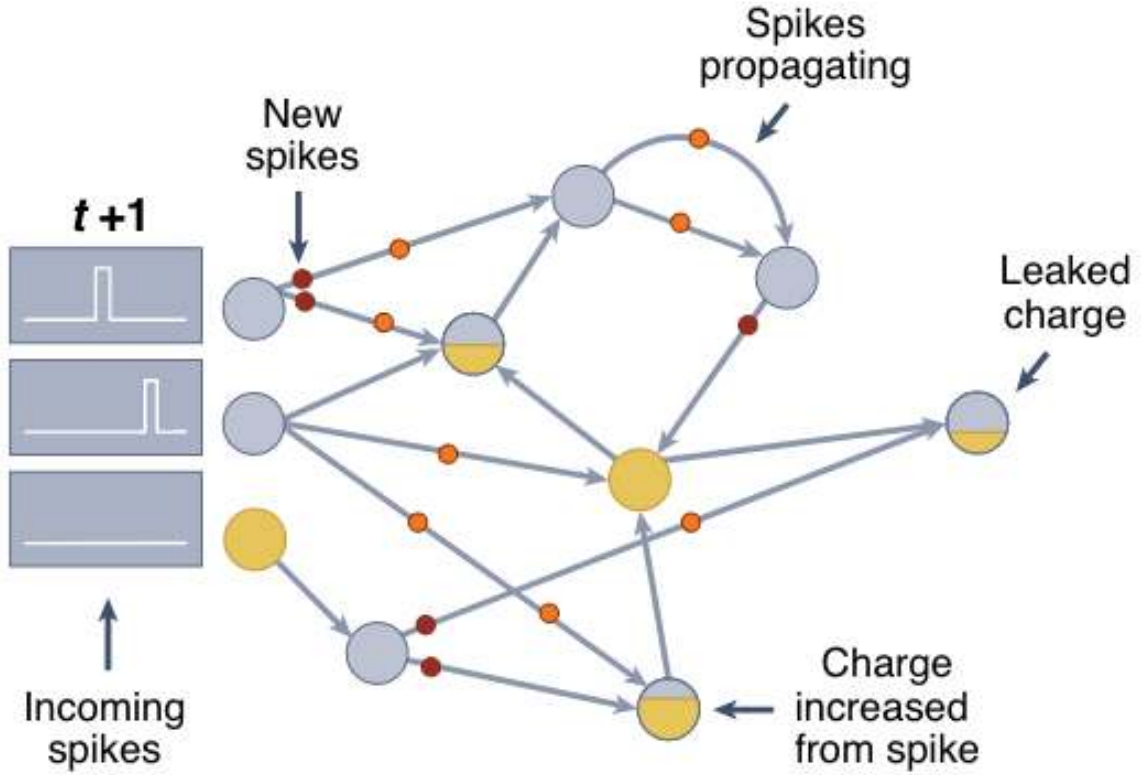
$$C_{\mathrm{m}} \frac{dV_{\mathrm{m}}(t)}{dt} = I(t) - \frac{V_{\mathrm{m}}(t)}{R_{\mathrm{m}}}$$

$V_m > V_{\text{threshold}} \rightarrow$ SPIKE!



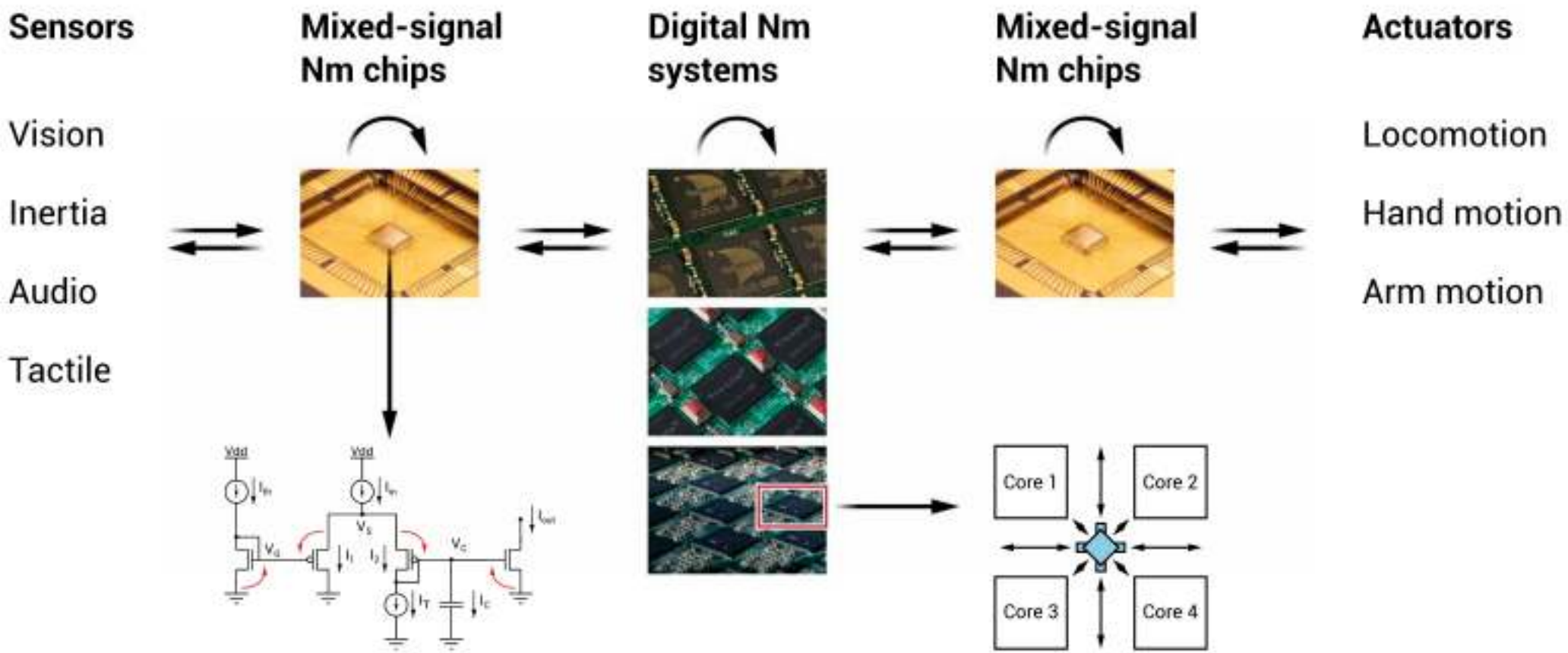
Spiking Neural Networks





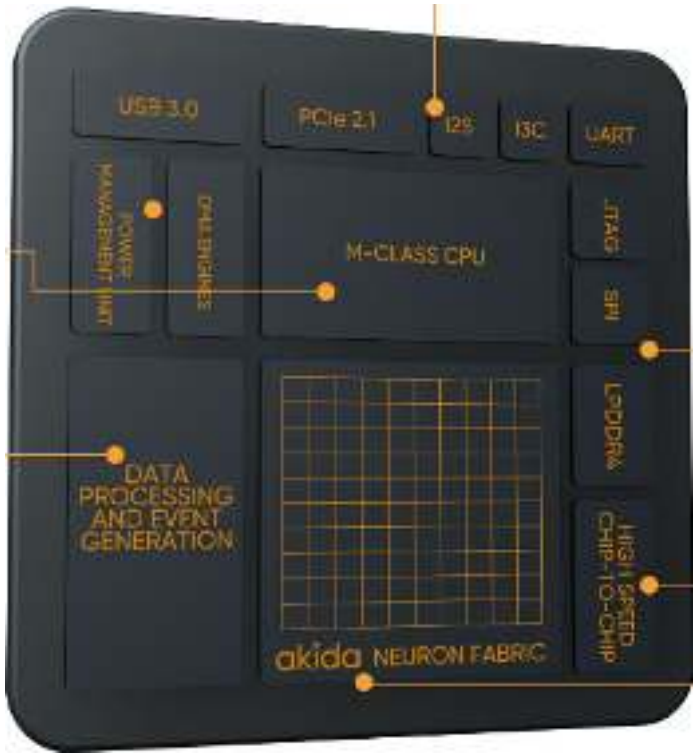
Large-scale neuromorphic computers

Distributed “thinking” in robotics



Try it out on ex3!

akida
AKD1000



Create and train

Leveraging TensorFlow/Keras or Akida model zoo

Quantize

Leveraging QuantizeML toolkit

Optional Quantization Aware Training step

Convert

Leveraging CNN2SNN toolkit

Code examples!

```
eqs_v = '''
dv/dt = (-g_leak*v + I_inj + I_rec + wnsigma*xi + I_d)/Cm :volt
dI_rec/dt = -I_rec/tau_rec : amp
I_d : amp
'''

eqs_h = '''
dv/dt = (-g_leak*v + I_inj + I_rec + wnsigma*xi)/Cm :volt
dI_rec/dt = -I_rec/tau_rec : amp
'''
neuron_group_rvisible = b2.NeuronGroup(
            N_v+N_c,
            model = eqs_v,
            threshold = 'v>theta',
            refractory = t_ref,
            reset = 'v=0*volt'
            )

neuron_group_rhidden = b2.NeuronGroup(
        N_h,
        model = eqs_h,
        threshold = 'v>theta',
        refractory = t_ref,
        reset = 'v=0*volt'
        )
```

```
Srs=b2.Synapses(neuron_group_rvisible, neuron_group_rhidden,
           model='''Afre : amp
                    Afost : amp
                    g : 1 (shared)
                    w : amp
                    lastupdate : second''',
                on_pre ='''Afre=Afre*exp((lastupdate-t)/tau_learn)
                    Afost=Afost*exp((lastupdate-t)/tau_learn)
                    Afre+=deltaA
                    I_rec_post+= w*int(randn()<0.5)
                    w=w+g*Afost
                    lastupdate = t''',
                on_post='''Afre=Afre*exp((lastupdate-t)/tau_learn)
                    Afost=Afost*exp((lastupdate-t)/tau_learn)
                    Afost+=deltaA
                    I_rec_pre+= w*int(randn()<0.5)
                    w=w+g*Afre
                    lastupdate = t'''
                    )
Srs.connect()
```

References and useful links

- Schuman, Catherine D., et al. "Opportunities for neuromorphic computing algorithms and applications." Nature Computational Science 2.1 (2022)
- Sandamirskaya, Yulia, et al. "Neuromorphic computing hardware and neural architectures for robotics." Science Robotics 7.67 (2022)
- Brian2 library for simulating Spiking Neural Networks: brian2.readthedocs.io
- Ex3 cluster with Akida BrainChip: ex3.simula.no
- Akida's API documentation: doc.brainchipinc.com

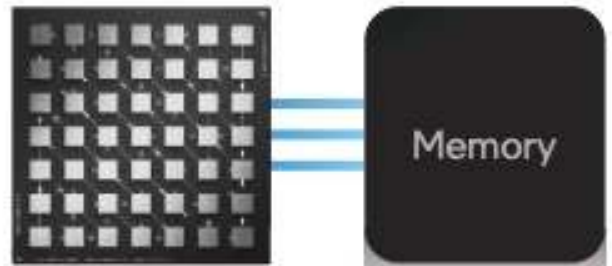
Backup

Today's Computing Architectures

Conventional Computing

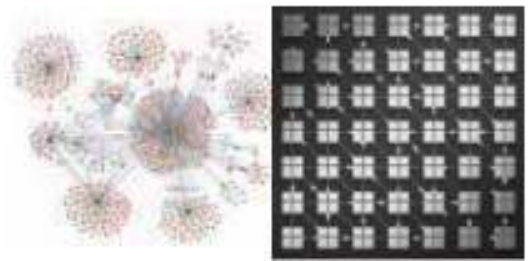
- Programming by Encoding Algorithms
- Synchronous Clocking
- Sequential Threads of Control

Parallel Computing

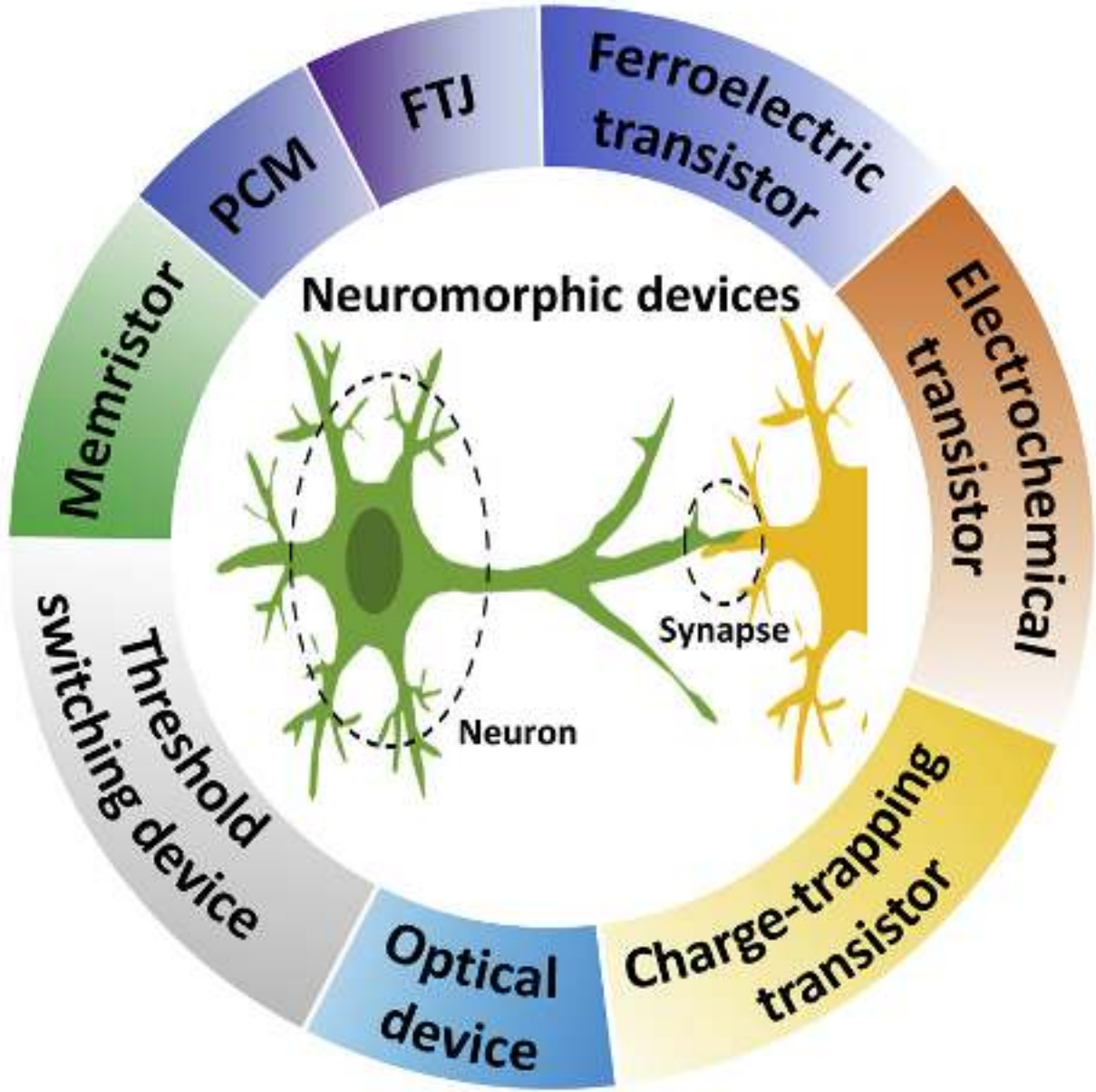


- Offline Training Using Labeled Datasets
- Synchronous Clocking
- Parallel Dense Compute

Neuromorphic Computing



- Learn On-the-Fly Through Neuron Firing Rules
- Asynchronous Event-Based Spikes
- Parallel Sparse Compute


Neuromorphic devices
Synapse
Neuron
Memristor
PCM
FTJ
Ferroelectric transistor
Electrochemical transistor
Charge-trapping transistor
Optical device
Threshold switching device

Human

Behavior

Nervous system

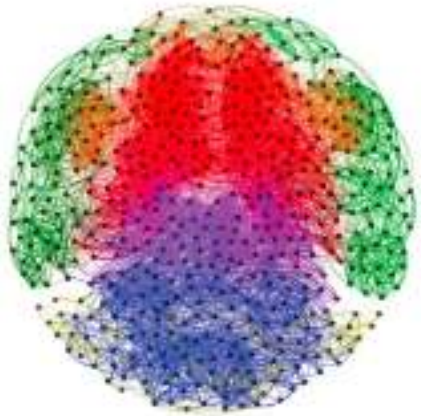
Sensory, cognitive, motor maps

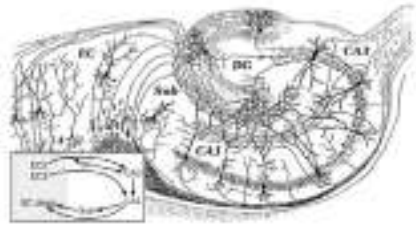
Local networks

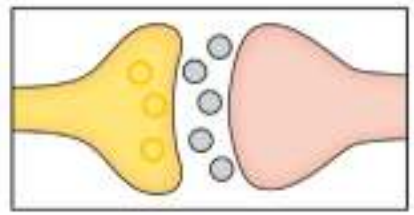
Neurons

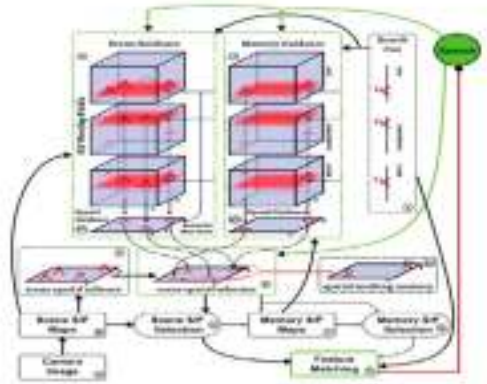
Synapses

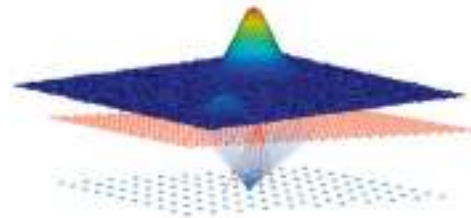
Molecules

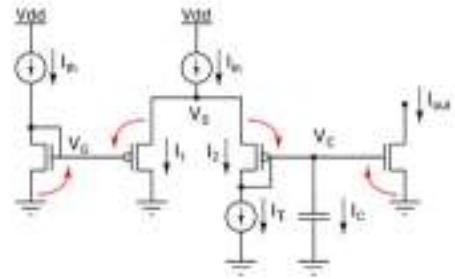












Robot

Behavior

Architecture

Modules

Networks

Neurons

Synapses

Transistors, memristors