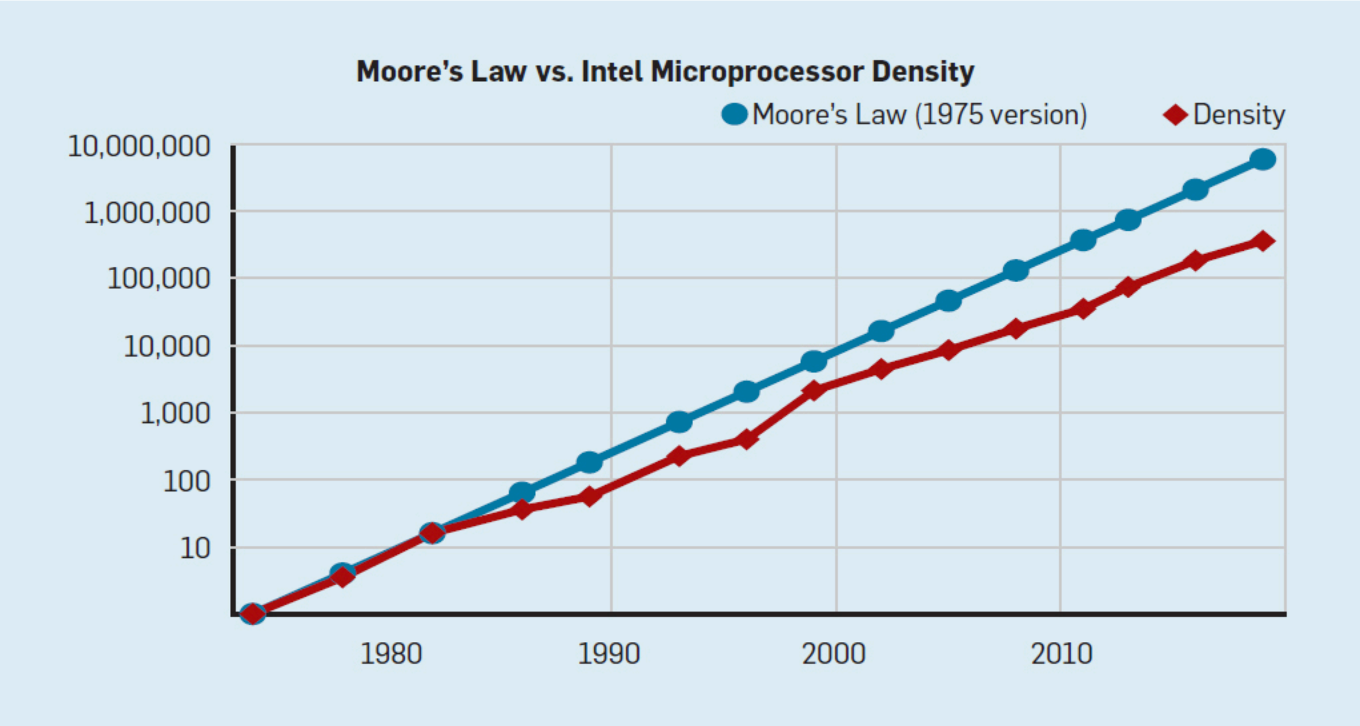
Batch << 1: Why Neuromorphic Computing Architectures Suit Real-Time Workloads

Jonathan Tapson

University of Technology Sydney

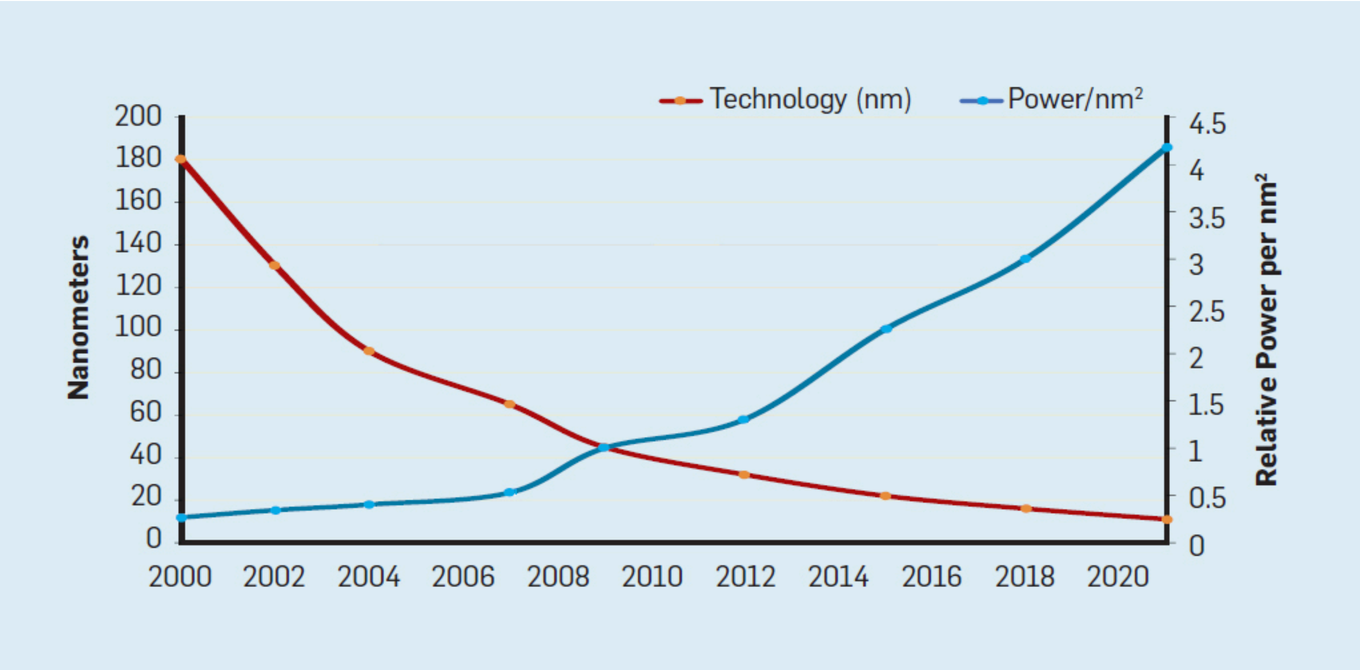
CSO, GrAI Matter Labs 2018-2020

Limitations of current technology: The end of Moore's Law



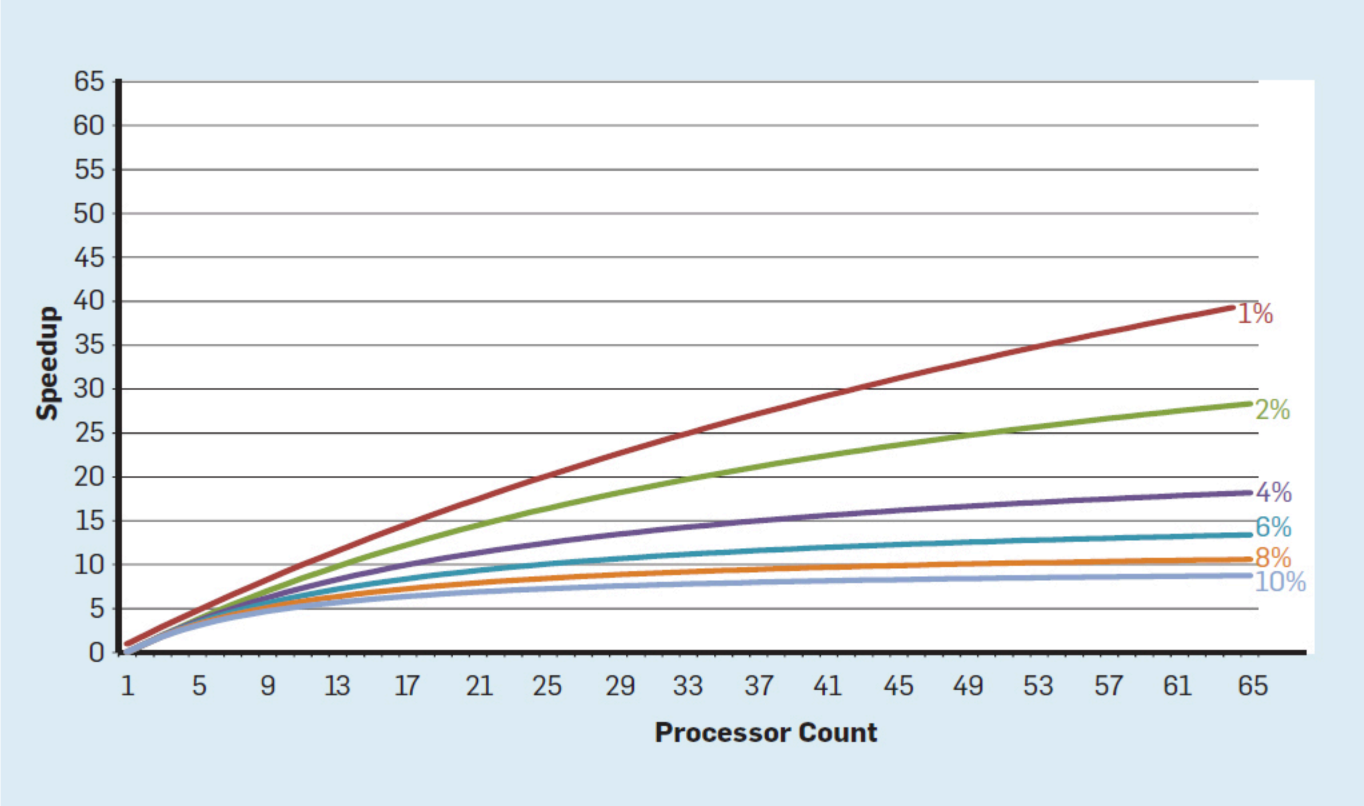
A New Golden Age for Computer Architecture
By John L. Hennessy, David A. Patterson
Communications of the ACM, February 2019, Vol. 62 No. 2, Pages 48-60

The End of Dennard Scaling



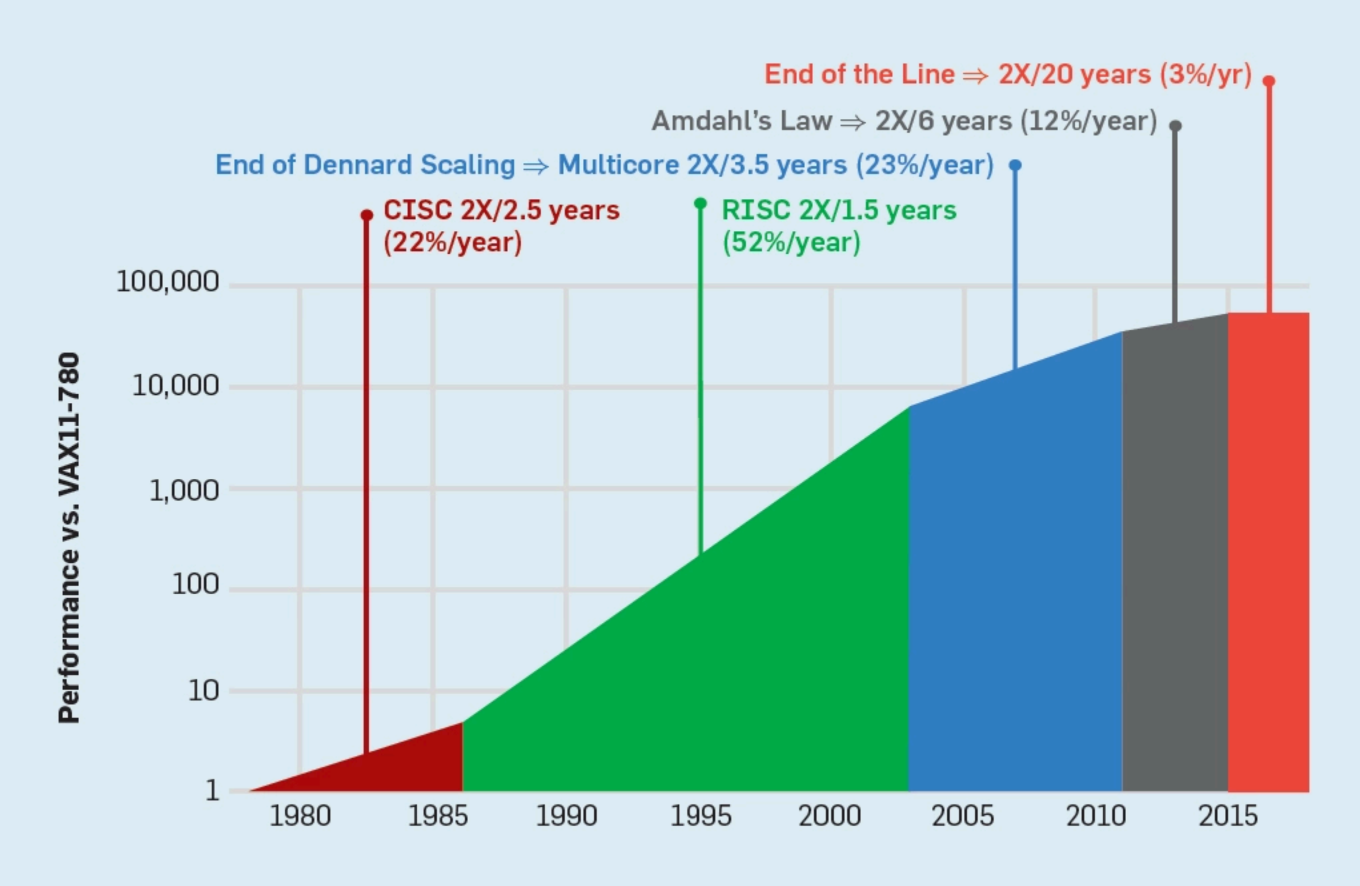
A New Golden Age for Computer Architecture
By John L. Hennessy, David A. Patterson
Communications of the ACM, February 2019, Vol. 62 No. 2, Pages 48-60

The End of Amdahls' Law



A New Golden Age for Computer Architecture
By John L. Hennessy, David A. Patterson
Communications of the ACM, February 2019, Vol. 62 No. 2, Pages 48-60

The End of the Line



A New Golden Age for Computer Architecture
By John L. Hennessy, David A. Patterson
Communications of the ACM, February 2019, Vol. 62 No. 2, Pages 48-60

The New Golden Age

- The end of Dennard scaling and Moore's Law … are not problems that must be solved but facts that, recognized, offer **breathtaking opportunities**.

- High-level, **domain-specific languages and architectures** … will usher in a new golden age for computer architects.

- The next decade will see **a Cambrian explosion of novel computer architectures**, meaning exciting times for computer architects in academia and in industry.

A New Golden Age for Computer Architecture
By John L. Hennessy, David A. Patterson
Communications of the ACM, February 2019, Vol. 62 No. 2, Pages 48-60

The New Golden Age

- The end of Dennard scaling and Moore's Law … are not problems that must be solved but facts that, recognized, offer **breathtaking opportunities**.

- High-level, **domain-specific languages and architectures** … will usher in a new golden age for computer architects.

- The next decade will see **a Cambrian explosion of novel computer architectures**, meaning exciting times for computer architects in academia and in industry.

A New Golden Age for Computer Architecture
By John L. Hennessy, David A. Patterson
Communications of the ACM, February 2019, Vol. 62 No. 2, Pages 48-60

The New Golden Age

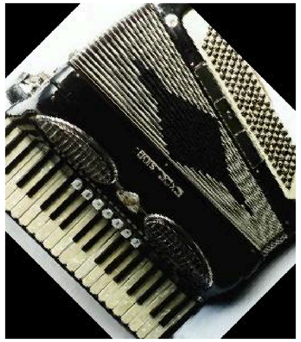
- The end of Dennard scaling and Moore's Law … are not problems that must be solved but facts that, recognized, offer **breathtaking opportunities**.

- High-level, **domain-specific languages and architectures** … will usher in a new golden age for computer architects.

Which neuromorphic architectures are suitable for which domains?

A New Golden Age for Computer Architecture
By John L. Hennessy, David A. Patterson
Communications of the ACM, February 2019, Vol. 62 No. 2, Pages 48-60

Workloads: Cloud ≠ Edge



Cloud Workload
(Caltech-101 as used for ResNet50)

Correlation = 0.027
Unchanged pixels: 0.2%
Pixels Δ < 0.05: 5.0%

Correlation = 0.085
Unchanged pixels: 0.3%
Pixels Δ < 0.05: 8.0%

Edge Workload
(NVIDIA PilotNet)
10 frames /sec

Correlation = 0.98
Unchanged pixels: 64%
Pixels Δ < 0.05: 87%

Correlation = 0.96
Unchanged pixels: 64%
Pixels Δ < 0.05: 86%

The Edge is Different

Edge workloads involve real-time processes

- Smart devices responding to input
- User interfaces with video / audio
- Closely-coupled feedback loops
  - Autonomous systems

Input data streams are continuous

- Video feeds
- Audio feeds
- Industrial sensor ensembles
- Bio signals (EEG, EKG, movement)

Characteristics of Edge Data Streams

- The data rate is much higher than the real information rate

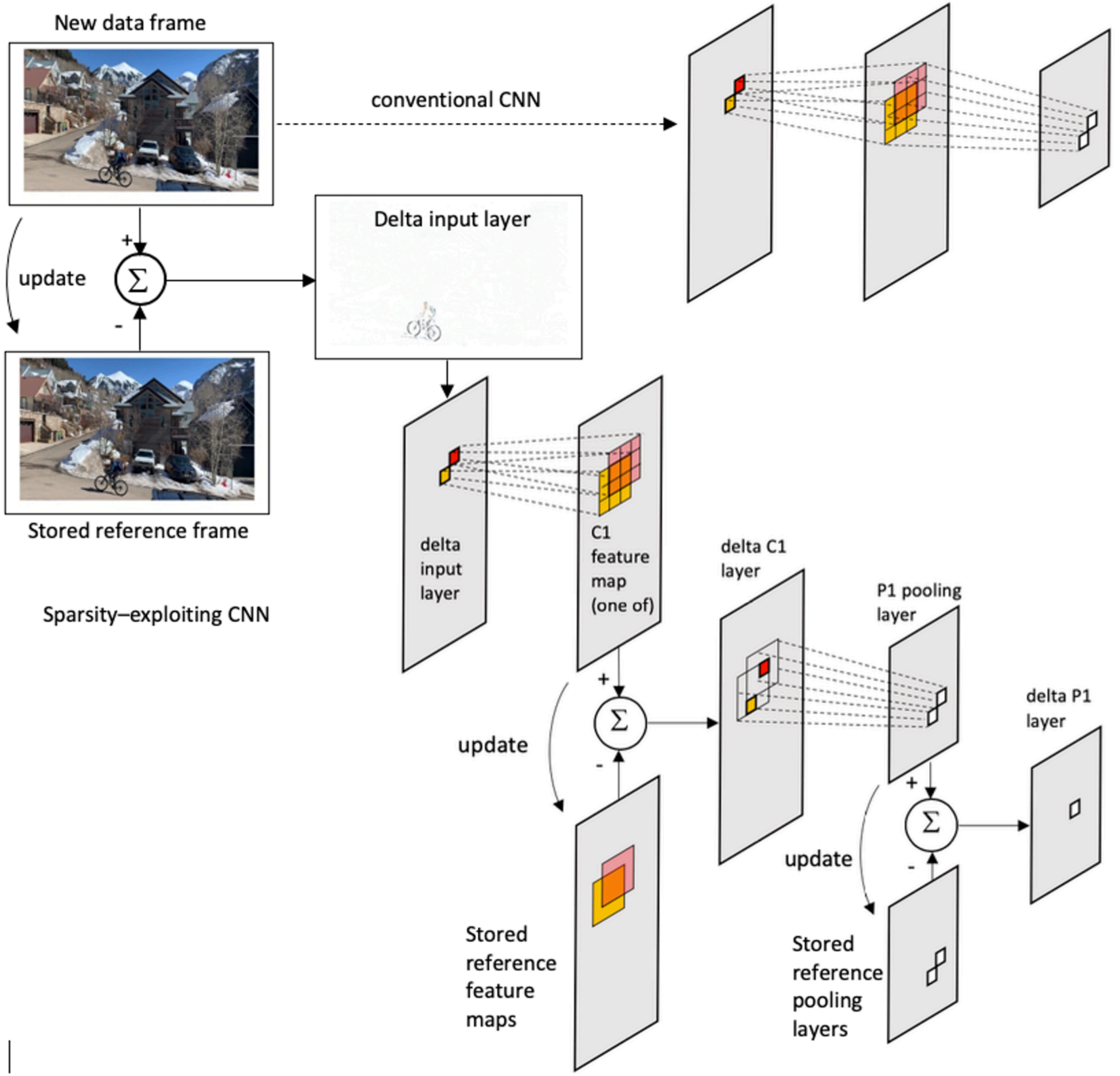
Data rate: 2 x microphones, 16 ksamples/s at 16bits -> 512 kbits/s

Information: Human speech = 39 bits/s av. (when speaking!)

Data rate: Always-on UXGA video – 79 MB/s

Information: Zero when no caller present
Lossless compression > 95%

Maintaining State is Expensive



Exploiting Sparsity

- Sparsity in **space**
- Sparsity in **time**
- Sparsity in **connectivity**
- Sparsity in **activation**

Exploiting Sparsity

- Sparsity in **space**
  - “Curse of dimensionality” - as data dimension grows, the proportion of null data points grows exponentially
- Sparsity in **time**
  - Real world signals have sparse changes in time
- Sparsity in **connectivity**
  - Compute only graph edges with significant weight (exploit “small world” connectivity)
- Sparsity in **activation**
  - Less than 40% of neurons may be activated by an upstream change

Advantages of Neuromorphic Computing

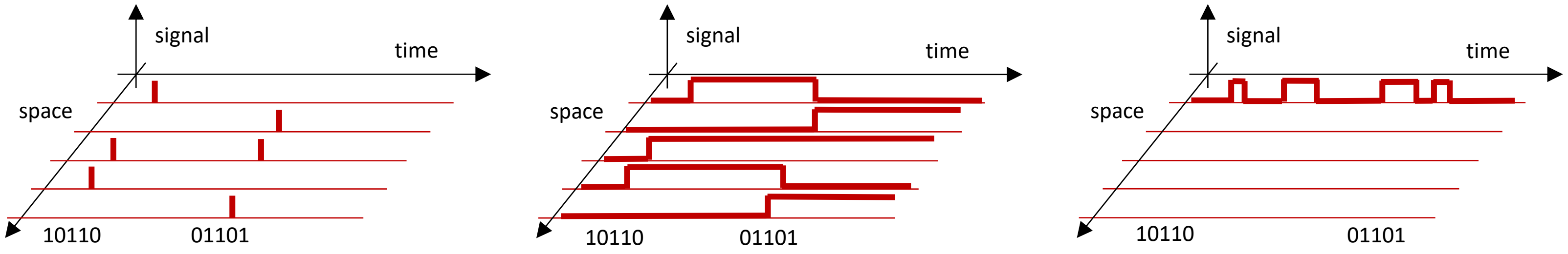
- Computational advantages are derived from:
  - **Spikes** – minimal power per signal event, noise immunity
  - **Events** - process only when change is occurring, time represents itself
  - **Sparsity** – Natural (real?) information is sparse relative to dimensionality
  - **Asynchrony** – no power consumed by clocking and clocked processing
  - **Analog signals** – infinite resolution
  - **Stochasticity** – robustness to noise, mismatch, process errors, probabilistic computation
  - **Compute-in-Network** - structure as computation
  - **Multiscale Connectivity** – at multiple scales

Coding of Numerical Input

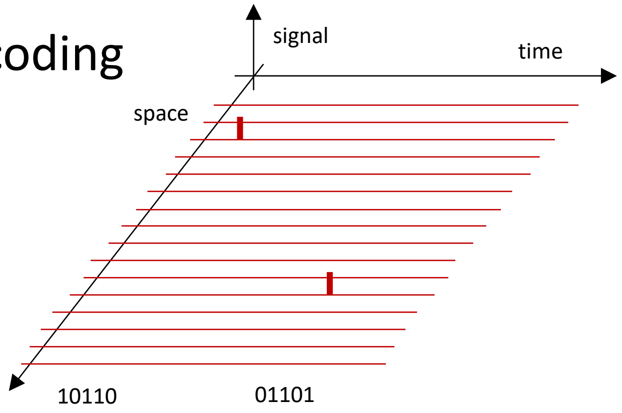
- Biological neuroscience
  - does not use explicit representation of numerical variables
  - Is robust to low precision in representation of variables
- Machine Learning
  - requires explicit use of numerical variables
  - Requires significant precision at some levels of representation
- How do we encode numbers in a spiking neuromorphic system?
    - Spike time or interval encoding
    - Spike rate encoding
    - Population encoding
    - Ensemble coding

Binary Encoding

- When we use binary digital encoding, there is little difference in power consumption between ensemble spike coding and conventional binary digital coding, either serial or parallel (when compared to other coding schemes) – a bit is a bit.

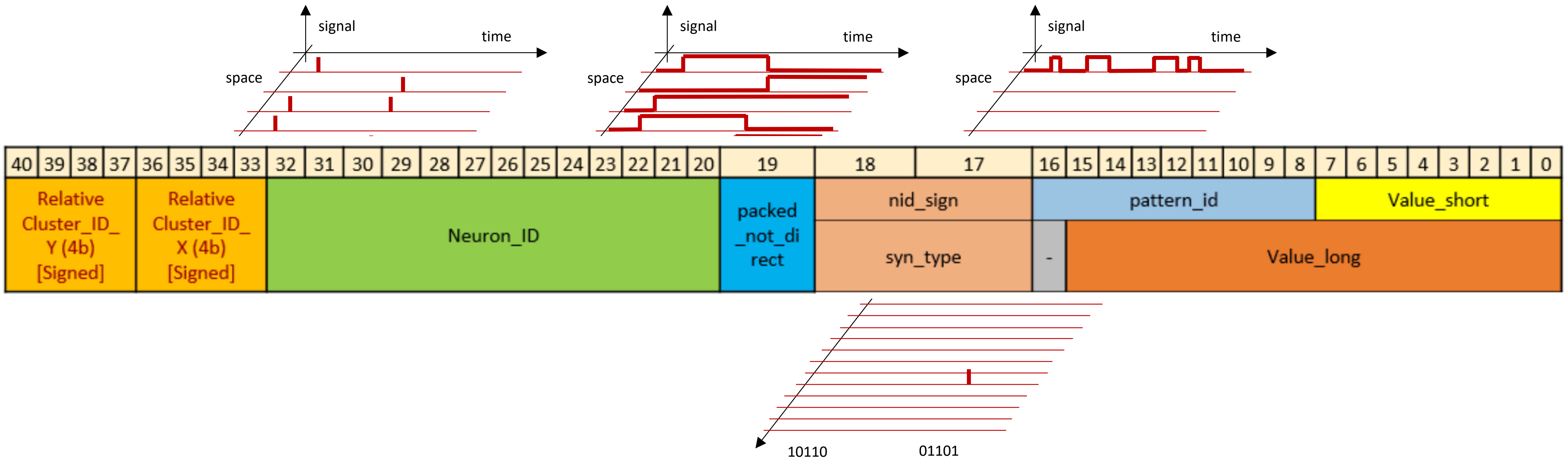


- An exception is ensemble (spatio-temporal) coding



Binary Encoding

- When we use binary digital encoding, there is little difference in power consumption between ensemble spike coding and conventional binary digital coding, either serial or parallel (when compared to other coding schemes) – a bit is a bit.



Digital Neuromorphic Computing

- Computational advantages are derived from:
  - ~~**Spikes** – minimal power per signal event, noise immunity~~
  - **Events** - process only when change is occurring, time represents itself
  - **Sparsity** – Natural (real?) information is sparse relative to dimensionality
  - ~~**Asynchrony** – no power consumed by clocking and clocked processing~~
  - ~~**Analog signals** – infinite resolution~~
  - ~~**Stochasticity** – robustness to noise, mismatch, process errors, probabilistic computation~~
  - **Compute-in-Network** - structure as computation
  - **Multiscale Connectivity** – at multiple scales

GrAI Matter Labs' NeuronFlow Architecture

- NeuronFlow
  - Is an architecture for **edge processing**

  - Designed for multiple types of computation load
    - Machine learning inference
    - Digital signal processing
    - Procedural computation
    - Mixtures of the above

  - Features
    - Very **low latency**
    - High **efficiency**

- Processes only changing signals, in real time (Batch << 1)

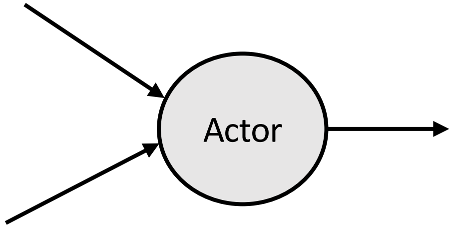
Why NeuronFlow?

- NeuronFlow is a hybrid of *Neuromorphic* and *Dataflow* architectures

- From Neuromorphic Computation we use:
    - Event-based processing
    - Data sparseness
    - Compute in network

- From (Fine-grained, dynamic) Dataflow Computation we use:
    - Compute on demand
    - Compute in memory

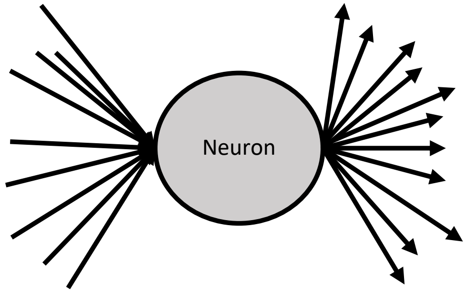
Dataflow and NN Similarities



inputs per construct: 1-3
output destinations per construct: 1-2
size of data packet: 8-64 bits
Apps: Procedural computation with heavy data processing.

Commonalities:

- Reactivity
- Sparsity of activity



inputs per construct: 100-10K
output destinations per construct: 100-10K
size of data packet: 1-8 bits
Apps: Pattern recognition: classification...

Different parameters require different design decisions to obtain a competitive solution.

GML platform uses configurable clusters that can combine data flow or SNN behaviour.

Architectural Features

- Neuronflow is a network of neuron clusters
- Neurons are connected by a network-on-chip (NoC)
  - The neural network is packet-switched
- The neurons process 8, 16 or 32-bit data
  - No spike-to-data coding problem
- All processing is event-triggered
  - No scheduled processing
  - No “pull” data
  - Processing only happens when new data is “pushed” to the neuron

GrAI ONE ACCELERATOR

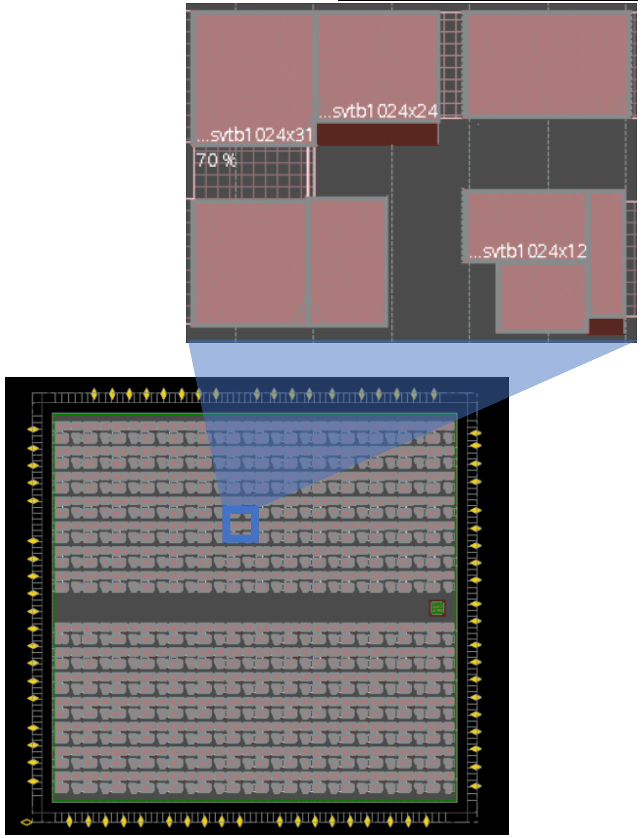
Flexible

Fully programmable / SDK
C++ / Python / TensorFlow

Self-Contained

No external DRAM

Neuron Core - Detailed

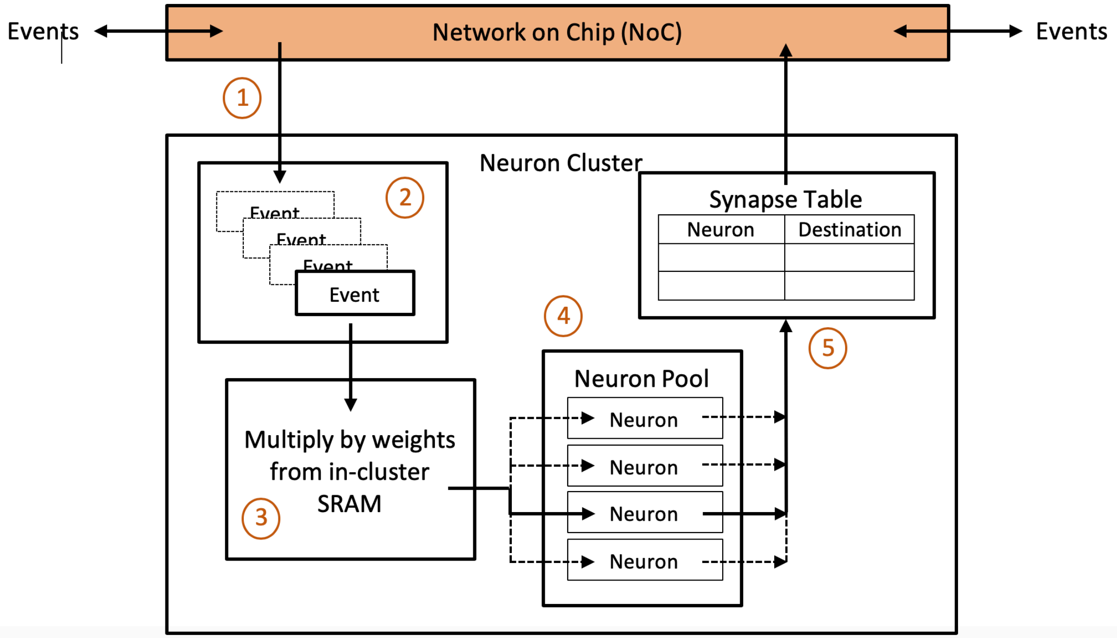


[14 x 14] Array of Neuron Cores

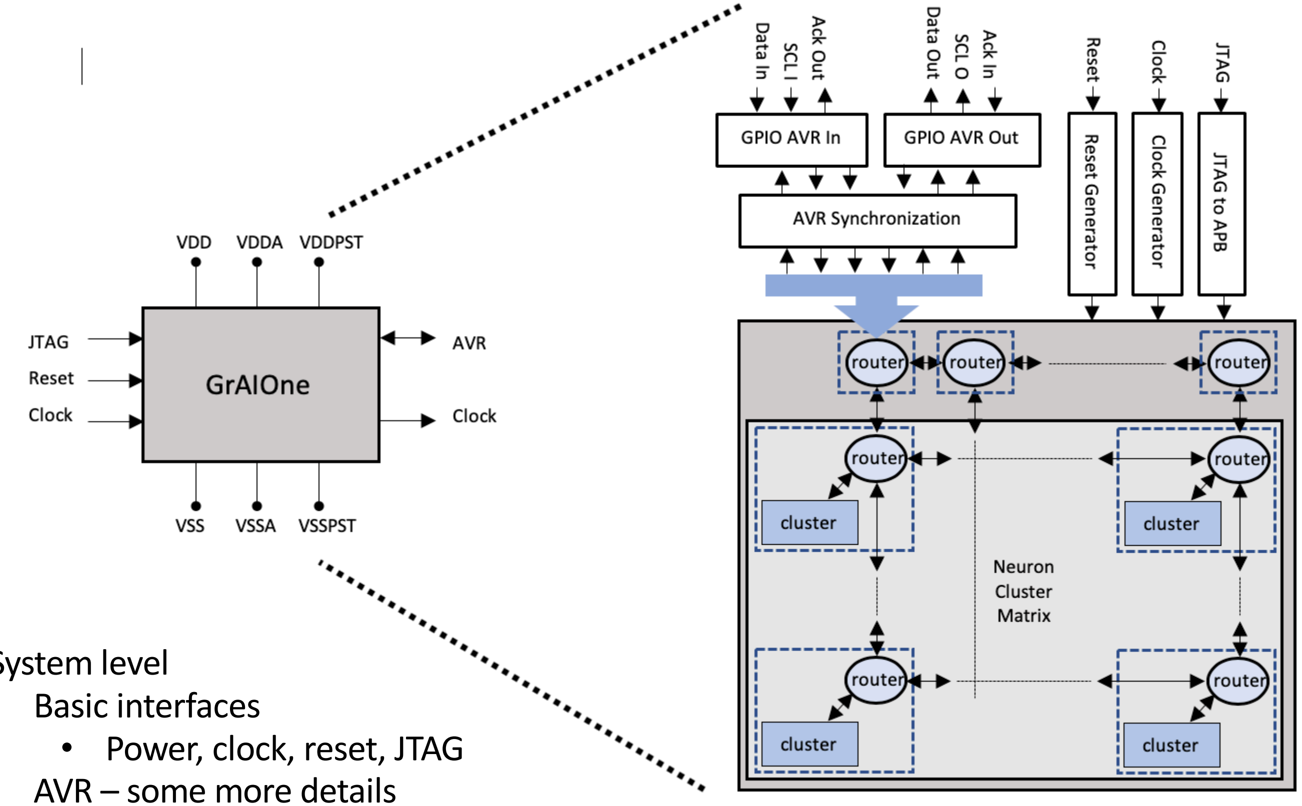
Configuration	GrAI One
# of Neuron Cores	196
# of Neurons	up to 200,704
Technology	TSMC 28 HPC+
Silicon Size	20 mm2
Package Size	8 x 8 mm2

Inside a Neuron Cluster

1. Events arrive via NoC based on destination address
2. Events are processed in FIFO queue
3. Weights are stored in local SRAM and can be shared. Data is weighted and passed to neurons
4. Neurons have state in local SRAM and perform basic neural and ALU functions
5. Events and mathematical output values are sent to destinations via synapse table and NoC



Top-level view



System level

- Basic interfaces
  - Power, clock, reset, JTAG
- AVR – some more details

Accelerator consists of a fabric of cores, each of which represents 1024 neurons. Cores are connected by a proprietary NoC.

AUTONOMOUS

NAVIGATION

Low latency low power
path planning
& steering control
in dynamic environments.

AFWERX CHALLENGE

PilotNet* on GrAI One

End-to-end learning system for self-driving cars

Utilization

< 200,000 neurons

Latency

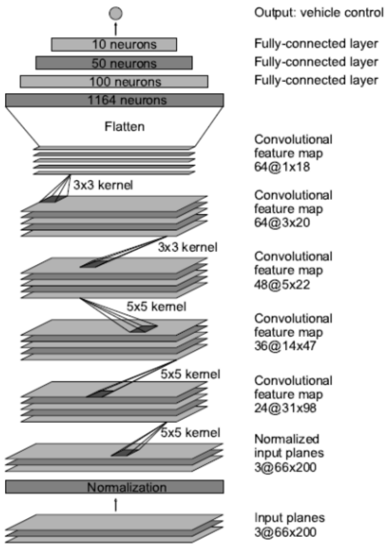
< 20 μs

[< 10,000 cycles @ 500MHz]

Power [dynamic]

< 10 mW

[< 1mJ / frame @ 10 fps]



* https://arxiv.org/pdf/1704.07911.pdf, adapted for implementation on GrAI One L

COGNITIVE

VOICE & VIDEO

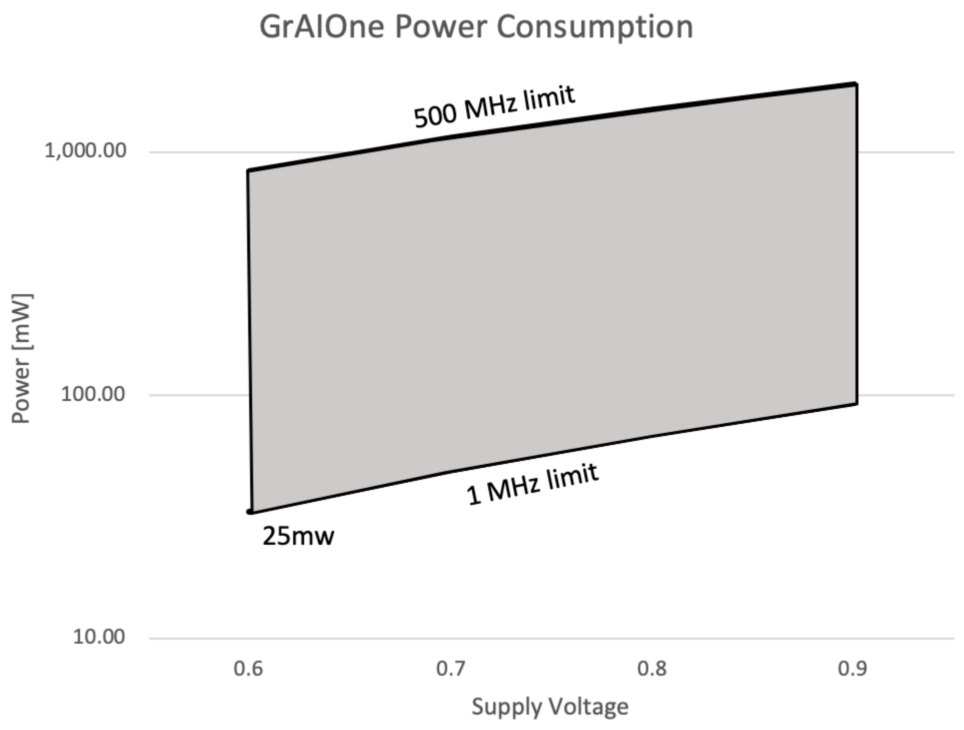
ASSISTANT

Low latency low power understanding of human speech and gestures.

Keyword Spotting + Hand Gesture Recognition on GrAI One

Keyword Spotting	Hand Gesture Recognition
Use Case 32 Keywords 16kHz + 512-FFT RNN 40/64/32	**Use Case** 10 Gestures “SparseNet’ > 90% accuracy
Latency < **3** µs	**Latency** < **1** µs
Power [dynamic] < **10** mW	**Power** [dynamic] < **25** mW

POWER AND PERFORMANCE

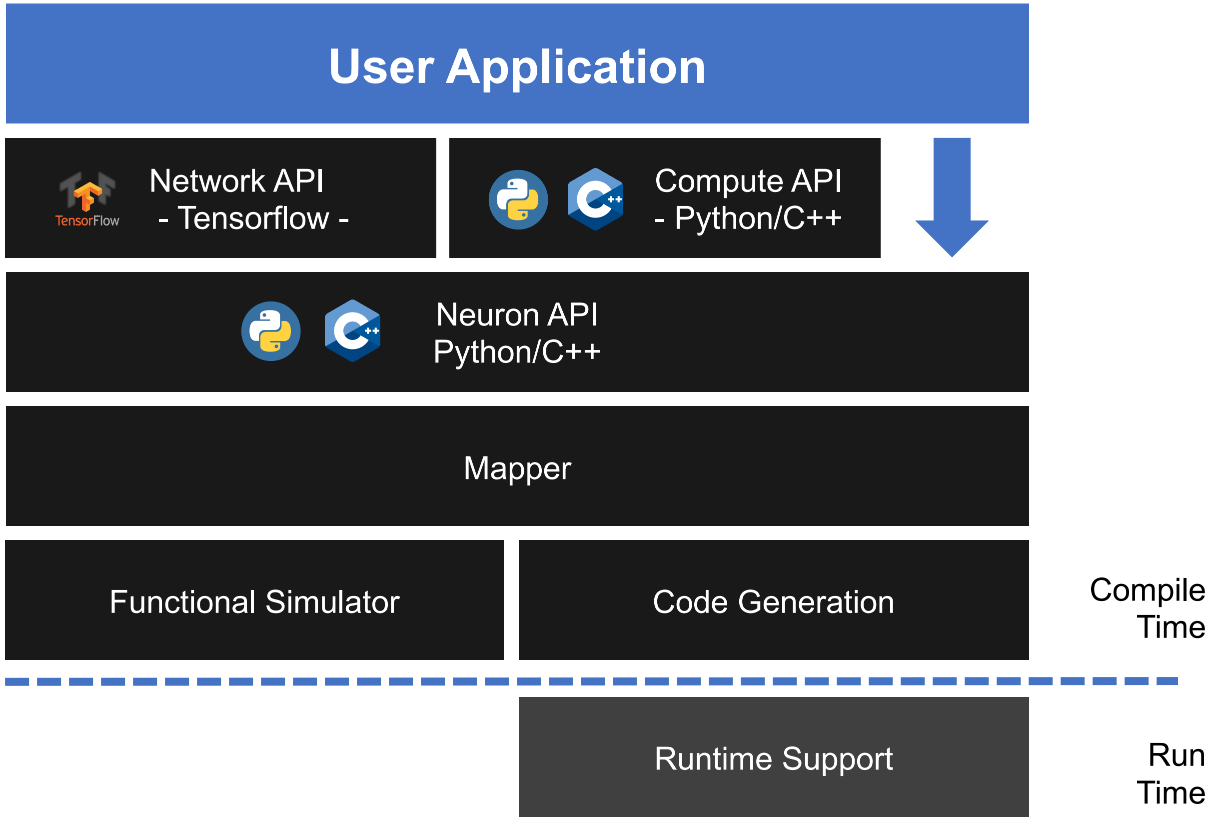


GrAIFLOW

SDK

Key Features

Conventional Programming & Machine Learning
Direct Network Import
Integrated Simulator
Graphical Editor



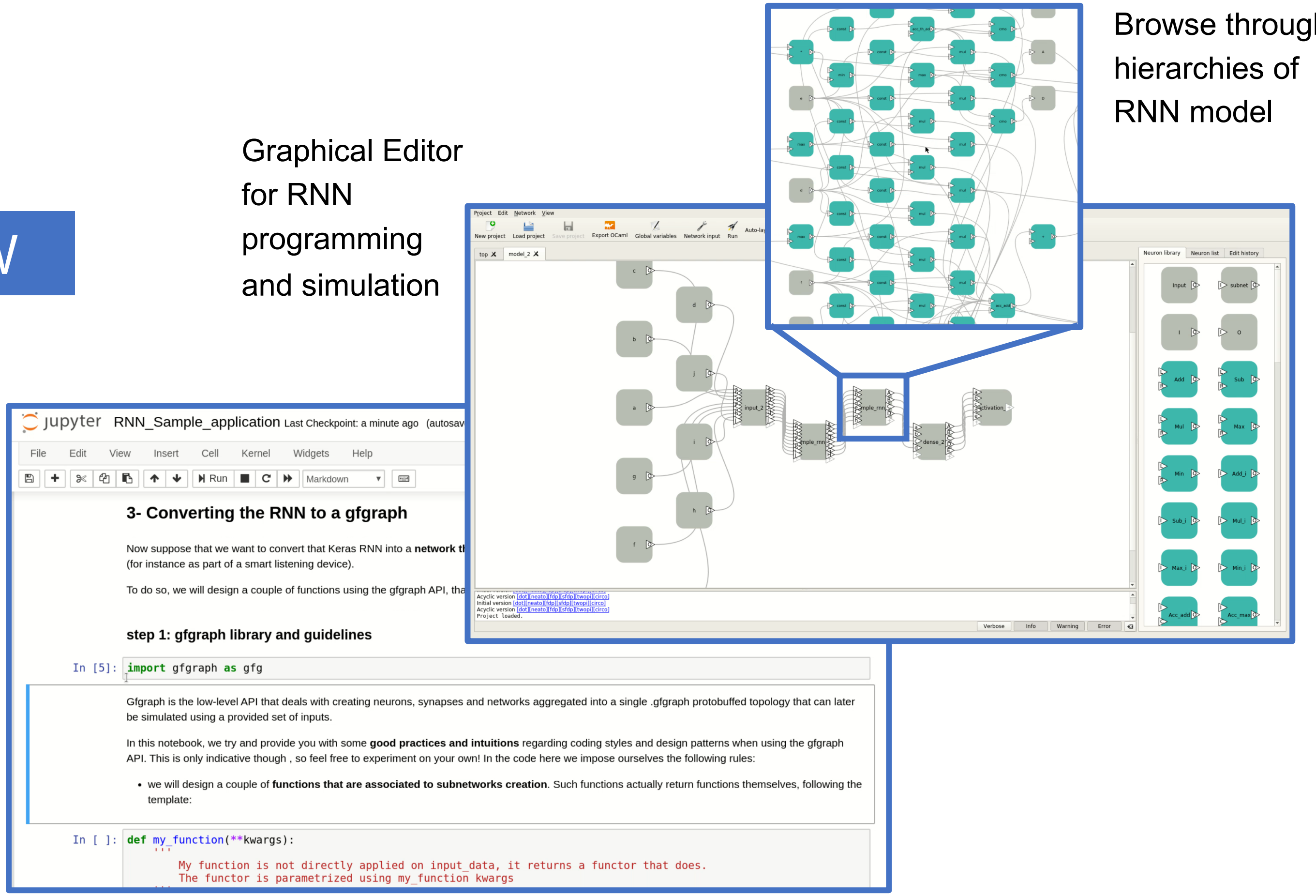
RNN

IN GRAIFLOW

Graphical Editor for RNN programming and simulation

Browse through hierarchies of RNN model

Jupyter Notebook with RNN template



Conclusions

- Current computational technology has plateaued; but a “Cambrian explosion” in computing architectures requires us to recognize that architectures should be domain-specific
- Neuromorphic architectures may be particularly well-suited to edge workloads in which data are real-time, highly correlated and sparse, and we can use Batch << 1 processing
- GrAI Matter LABS’ NeuronFlow is an architecture which has been optimized for these workloads, using a selection of neuromorphic principles

Thanks!