Exploiting activation sparsity (a'la spiking) for accelerating CNNs and RNNs

Tobi Delbruck
Sensors Group, Inst. of Neuroinformatics, UZH-ETH Zurich

Our finding:
Synchronous digital accelerators benefit from being "neuromorphic"

Delbruck, T., and S. Liu. 2019. "**Data-Driven Neuromorphic DRAM-Based CNN and RNN Accelerators**." In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, 500–506. https://doi.org/10.1109/IEEECONF44664.2019.9048865 .

Sparsity

Estimate energy use and spike rate in the human brain

$$10^{11} \times 10^{4} \times 10^{-1} \times 10^{-9} \times 10^{-3} \times X = 10^{1}$$

Neurons* Syn/neuron V A sec Avg. spike rate W

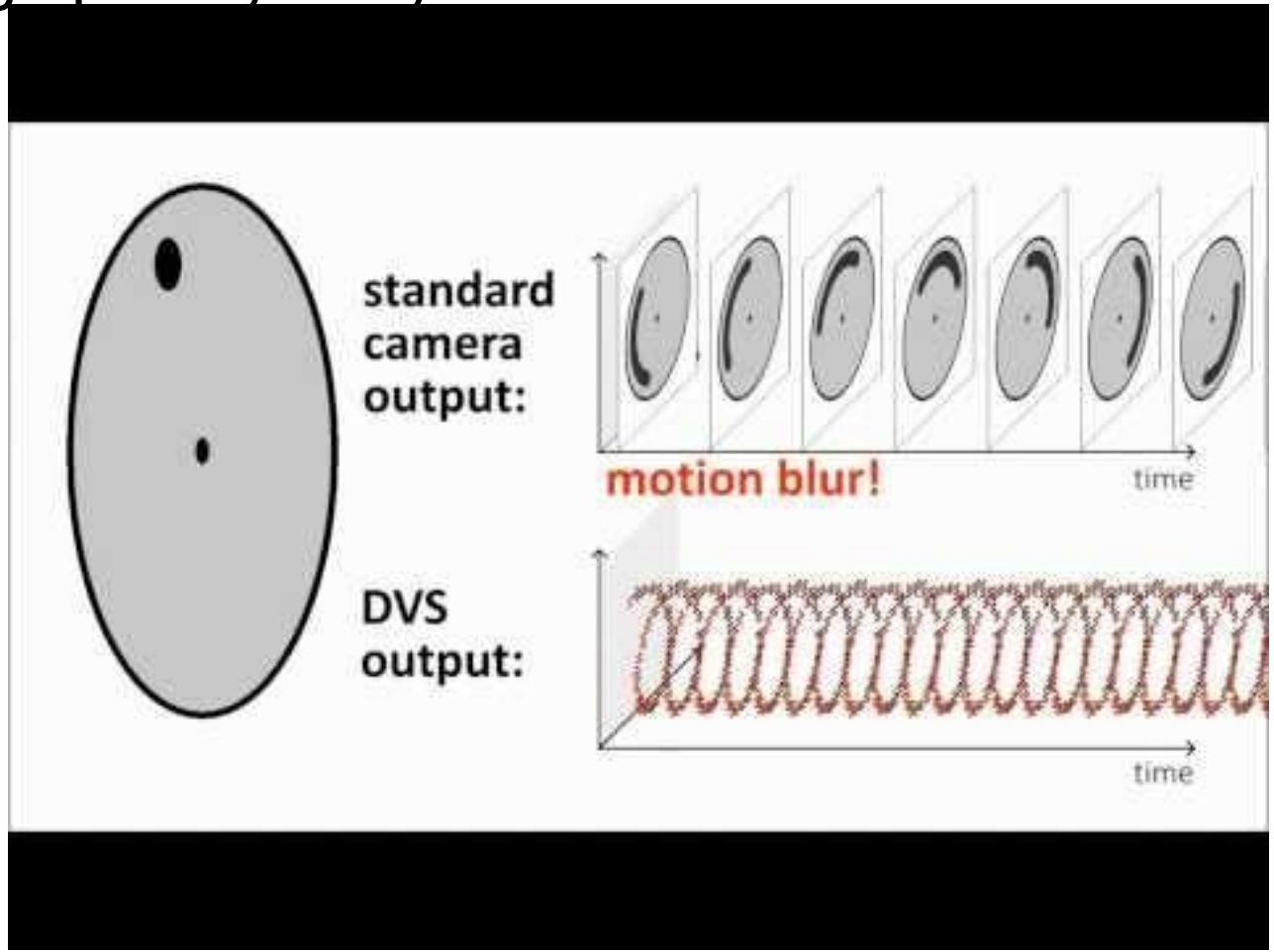
J/syn. Act.=10^{-13}J=**0.1pJ**

Avg. **output** spike rate = 1 Hz

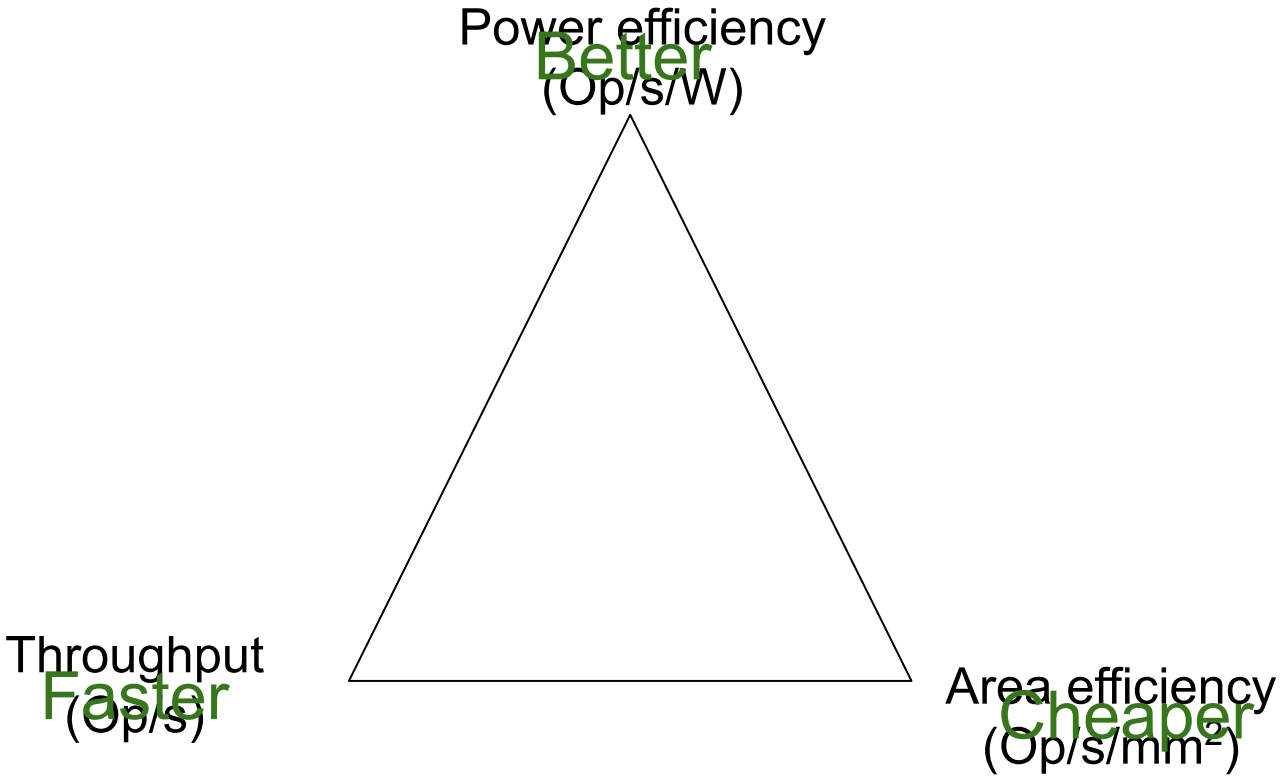
10^{4} fan-out means avg. synaptic **input** rate per neuron = 10 kHz

It is very different than conventional DNNs, where every neuron sends its messages to all recipients at the sample rate, e.g. 100Hz
There is a big opportunity to exploit sparsity in weights and states to avoid useless memory access and computations

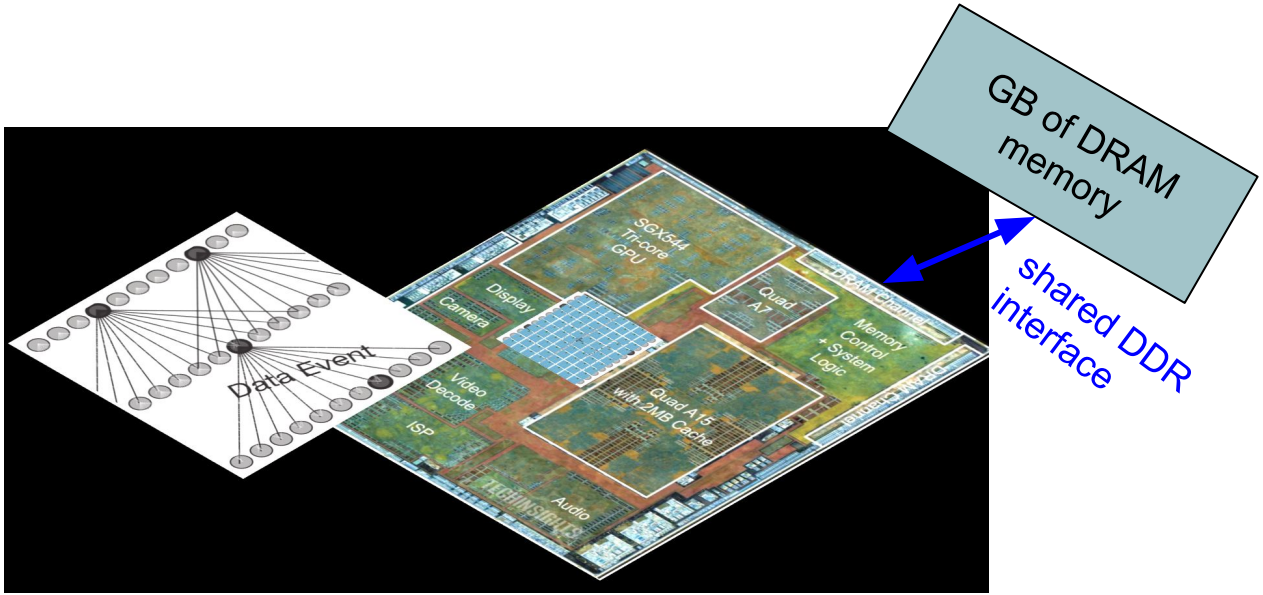
Exploiting sparsity in dynamic vision sensor event cameras



FOMs


Power efficiency
Better
(Op/s/W)
Throughput
Faster
(Op/s)
Area efficiency
Cheaper
(Op/s/mm²)

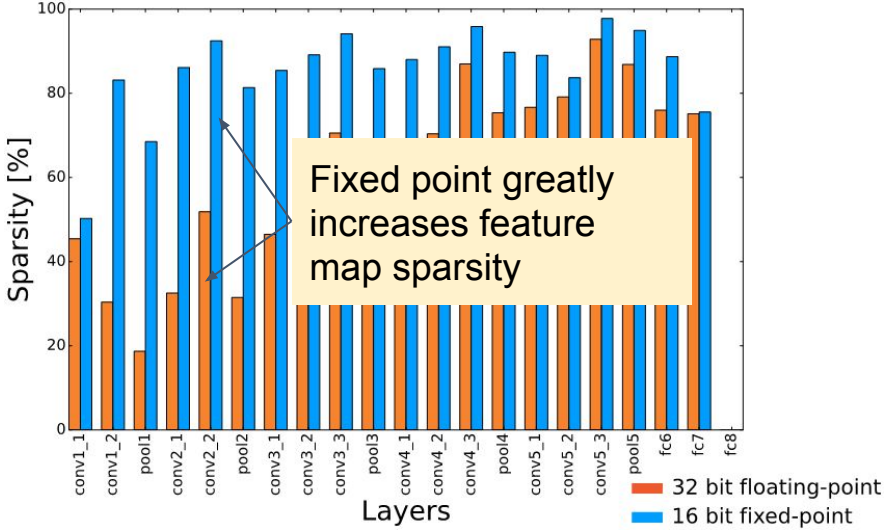
Our context:
Shared workload AI hardware that is competitive for all 3 FOMs



The networks are too big and there are too many workloads to fit on any single core.
Only DRAM offers affordable, scalable memory.
Minimize its accesses by making them sparse and reusing them to amortize costs.

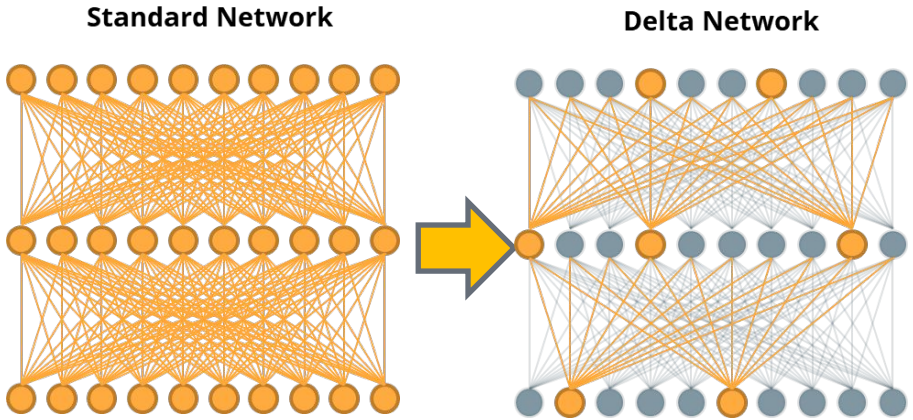
Exploit SNN sparsity concepts for higher efficiency and throughput with DRAM storage

Aimar's *Nullhop* CNN accelerator exploits spatial activation sparsity by compressing feature map zeros and skipping resulting "non spiking" MACs.



- Provides ~4X speedup
- In 28nm, gave SOA 3 TOp/s/W core and 470 GOp/s in 6.3mm^2

Gao's *DeltaRNN* RNN accelerator exploits temporal activation sparsity by propagating only "spiking" units with changes in activation >threshold



- Provides ~5-10X speedup (via less DRAM weight access)
- Enables $90 2W *EdgeDRNN* to run large multilayer RNNs as quick as a 100W GPU

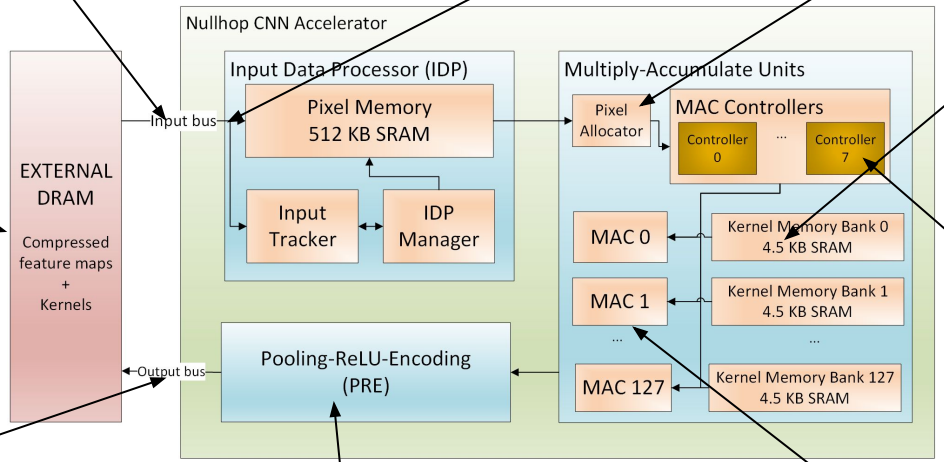
Nullhop CNN accelerator

1. Compressed layers are stored using sparsity map that uses 1 bit for zero activations and 16 bits for nonzero activations

2. Pixels are loaded only once per 128 output feature maps

3. Pixel & channel ordering scales to arbitrary image size

4. Zero pixel MACs are completely skipped



5. Kernels for layer are loaded to MAC SRAM banks

6. Controllers cluster MAC units for 16-128 output maps/pass

7. Output maps are computed in parallel (up to 128)

8. 2x2 max pooling “on the fly” cuts external DRAM writes by 4X

9. Compressed layer is written out ready for being streamed back

Delta RNN accelerator

Motivation

Accelerate Gated RNNs at the Very Edge

Gated RNNs

- Long Short-Term Memory (**LSTM**) [1], Gated Recurrent Unit (**GRU**) [2]
- Useful in temporal sequence processing
- Widely used in speech recognition and machine translation.
- Computation is usually on cloud

amazon alexa

Edge Inference

- Immune to network failure
- Less privacy issue
- Latency critical (batch-1)

Examples

- On-device speech recognition
- Human-in-the-loop Robot Control

State-of-the-Art RNN FPGA Accelerator

BBS [3]

- Exploit weight sparsity in LSTM
- SOA Batch-1 throughput: 2.4 TOp/s
- High-end Arria 10 FPGA (8.2 MB on-chip mem.)
- Power > 19 W
- Cost > $4000

1L-1024H-LSTM
6.4 MB dense weights

8x comp.

0.8 MB weights on-chip

Still expensive for most edge platforms

DRAM needed for big networks on the edge

Performance limited by DRAM Bandwidth

Our Target: EdgeDRNN

- To use **spiking Delta RNN** to reduce DRAM access
- To exploit **temporal sparsity** in GRU
- To achieve **low latency** of running **large networks** on the **edge**
- **Scalable** for any available memory bandwidth
- For real-world problem, e.g. classification and control

[1] S. Hochreiter, et al., "Long Short-Term Memory", Neural Computation, 1997
[2] K. Cho, et al., "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation", EMNLP 2014
[3] S. Cao, et al., "Efficient and Effective Sparse LSTM on FPGA with Bank-Balanced Sparsity", FPGA 2019


Trixsy
The card finding magic robot

Aimar, Alessandro, Hesham Mostafa, Enrico Calabrese, Antonio Rios-Navarro, Ricardo Tapiador-Morales, Iulia-Alexandra Lungu, Moritz B. Milde, et al. 2019. "NullHop: A Flexible Convolutional Neural Network Accelerator Based on Sparse Representations of Feature Maps." *IEEE Transactions on Neural Networks and Learning Systems* 30 (3): 644–56. https://doi.org/10.1109/TNNLS.2018.2852335.

Neil, Daniel, Jun Haeng Lee, Tobi Delbruck, and Shih-Chii Liu. 2017. "Delta Networks for Optimized Recurrent Network Computation." In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, 2584–93. ICML'17. Sydney, NSW, Australia: JMLR.org. http://dl.acm.org/citation.cfm?id=3305890.3305948.

Gao, C., A. Rios-Navarro, X. Chen, S-C Liu, and T. Delbruck. 2020. "EdgeDRNN: Recurrent Neural Network Accelerator for Edge Inference." *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 1–1. https://doi.org/10.1109/JETCAS.2020.3040300.

Delbruck, T., and S. Liu. 2019. "Data-Driven Neuromorphic DRAM-Based CNN and RNN Accelerators." In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, 500–506. https://doi.org/10.1109/IEEECONF44664.2019.9048865 .