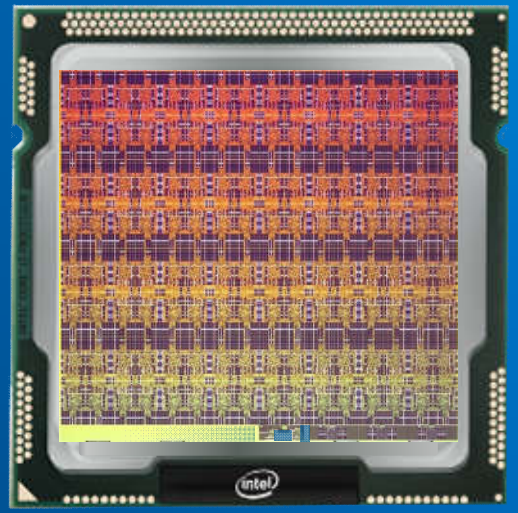
Spiking Neural Networks for Event-based Vision

Garrick Orchard

CVPR Event-based Vision Workshop

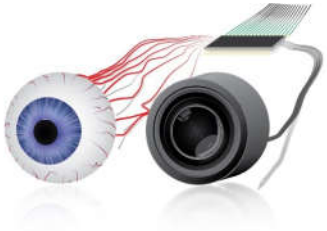
17th June 2019, Long Beach, CA



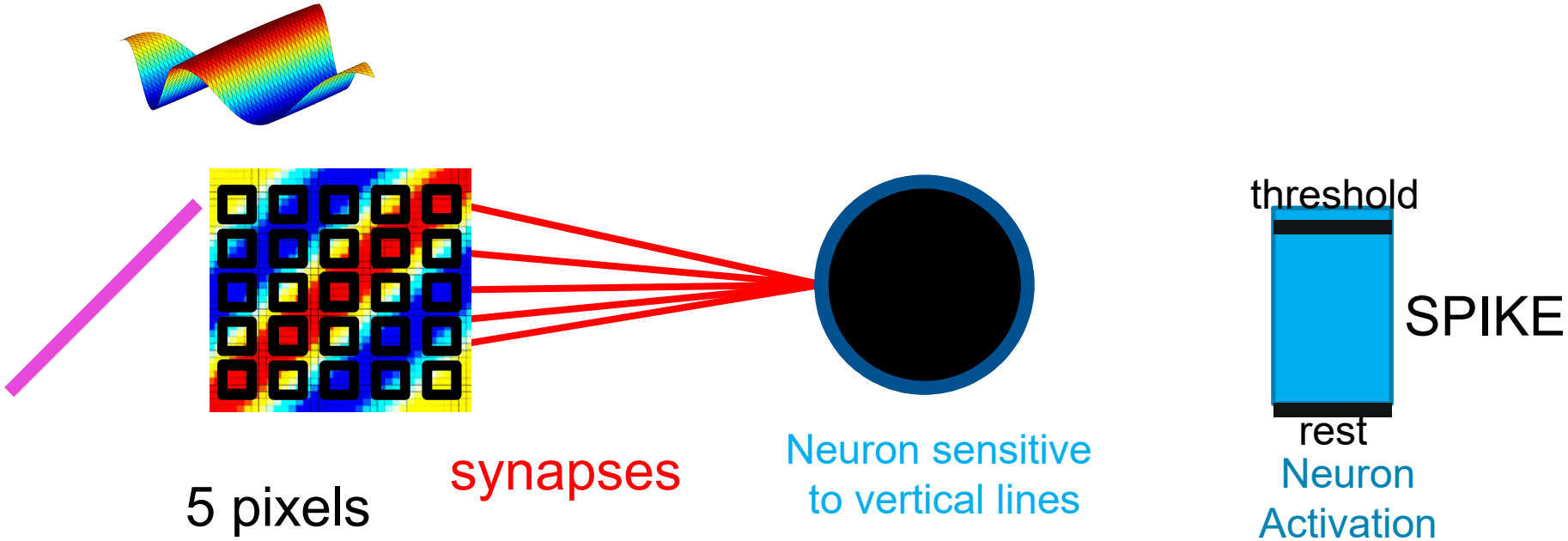
Intel's Loihi Neuromorphic Research Chip

Legal Information

Visual Recognition

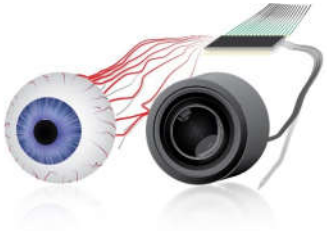


Spatial filters using neurons as coincidence detectors

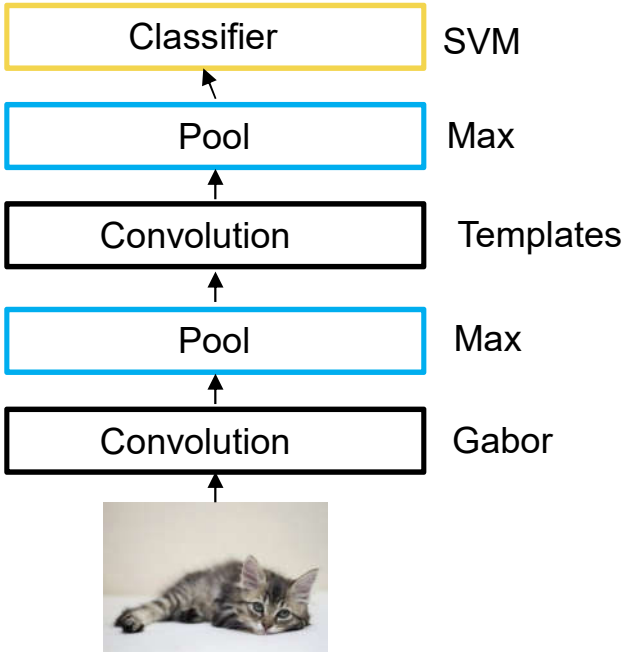


Orchard, G.; Meyer, C.; Etienne-Cummings, R.; Posch, C.; Thakor, N.; and Benosman, R., "HFIRST: A Temporal Approach to Object Recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* vol. 37, no. 10, pp. 2028-2040, Oct. 2015

Visual Recognition

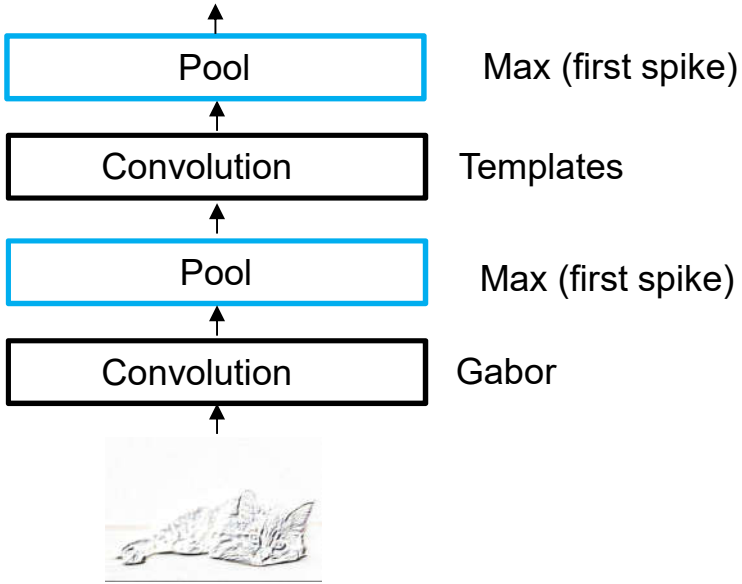


Basic HMAX (frames)



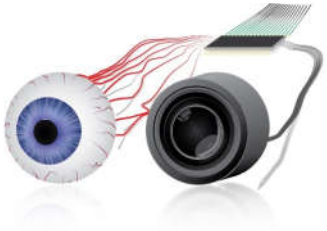
Serre, Thomas, Lior Wolf, and Tomaso Poggio. *Object recognition with features inspired by visual cortex*. Massachusetts Inst Of Tech Cambridge Dept Of Brain And Cognitive Sciences, 2006.

HFirst (spikes)

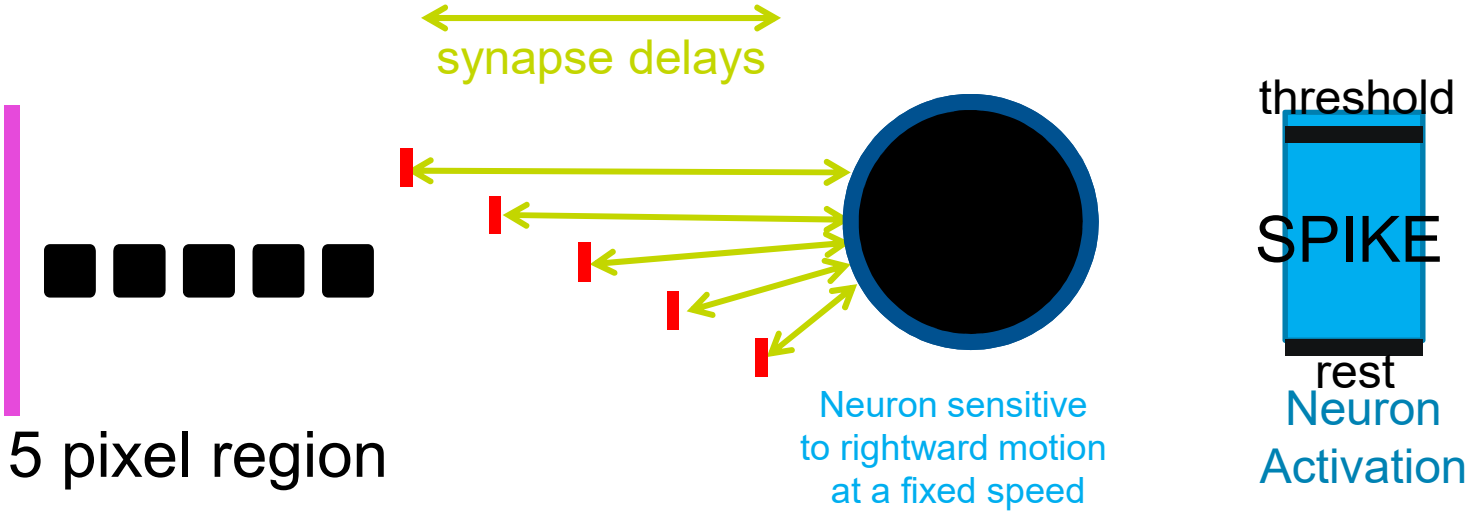


Orchard, G.; Meyer, C.; Etienne-Cummings, R.; Posch, C.; Thakor, N.; and Benosman, R., "HFIRST: A Temporal Approach to Object Recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* vol. 37, no. 10, pp. 2028-2040, Oct. 2015

Visual Motion Estimation

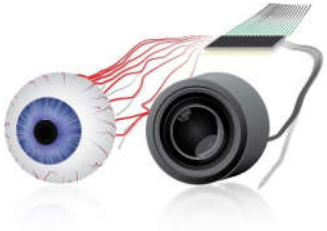


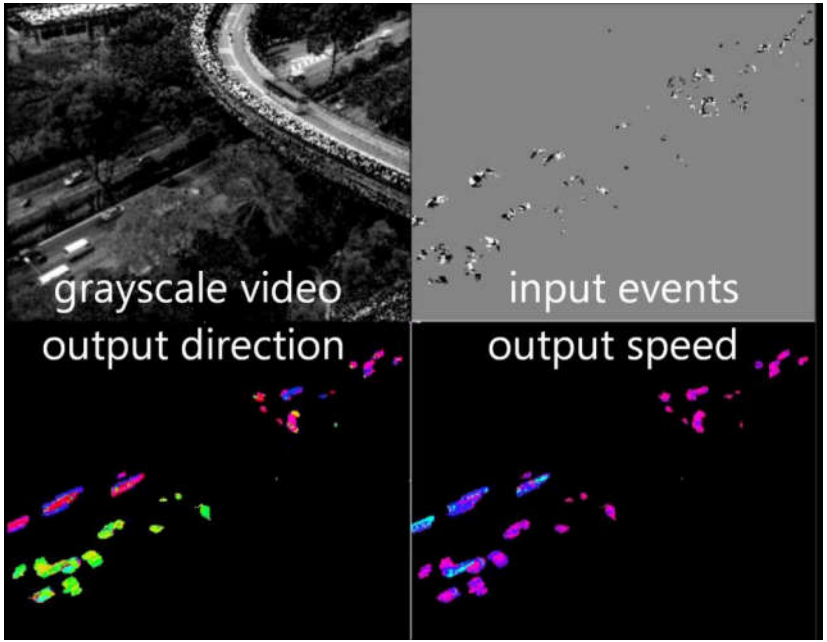
Creating coincidence using delays

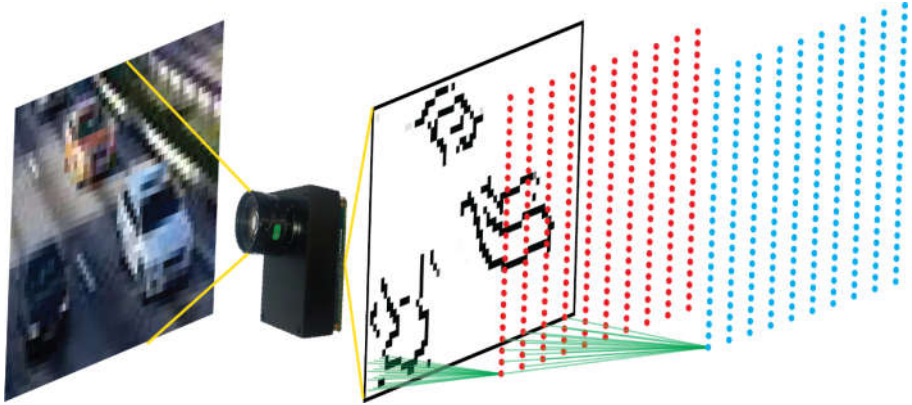


Orchard, G.; Benosman, R.; Etienne-Cummings, R.; and Thakor, N. "A Spiking Neural Network Architecture for Visual Motion Estimation," *IEEE Biomedical Circuits and Systems, Rotterdam, Holland, Nov 2013.*

Visual Motion Estimation



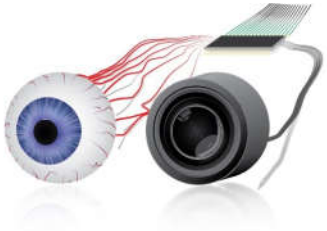




pix/s

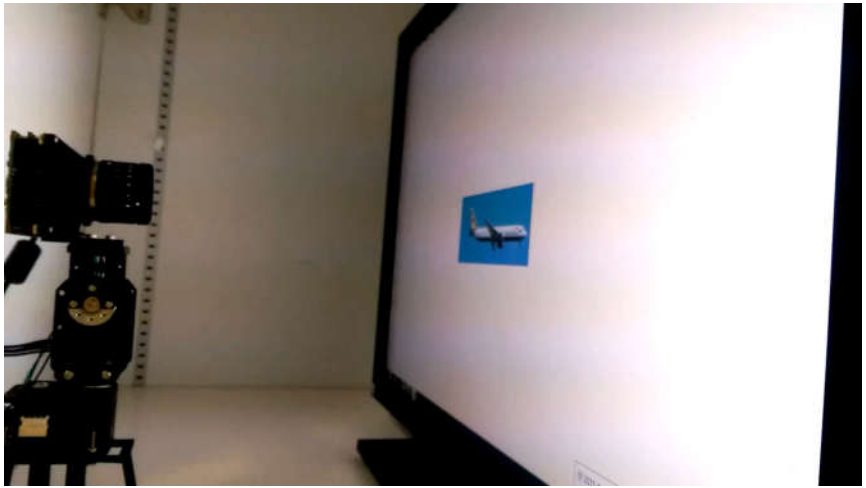
Orchard, G.; Benosman, R.; Etienne-Cummings, R.; and Thakor, N. "A Spiking Neural Network Architecture for Visual Motion Estimation," *IEEE Biomedical Circuits and Systems, Rotterdam, Holland, Nov 2013.*

Turning attention to Datasets



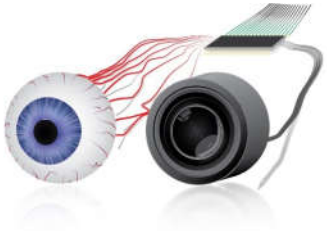
HFirst and SAVME models were both hand-tuned on very small self-captured datasets

There was a clear need for larger better datasets



Orchard, G.; Cohen, G.; Jayawant, A.; and Thakor, N. “Converting Static Image Datasets to Spiking Neuromorphic Datasets Using Saccades", *Frontiers in Neuroscience,* vol. 9, pp. 437, Oct. 2015

Applying Deep Learning to SNNs

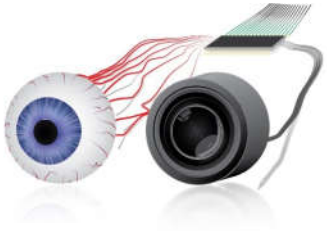


- Two main obstacles to using backpropagation for SNNs
  1) Non-differentiability of spike generation
  2) Error assignment

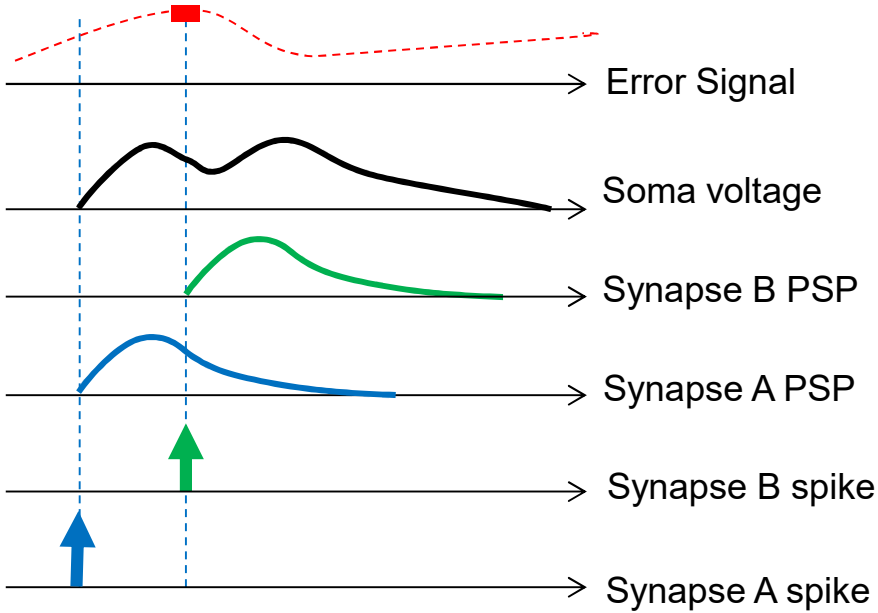
1) Non-differentiability of spike generation
   - A proxy function can be used in place of the derivative
   - Easily and successfully implemented by many works

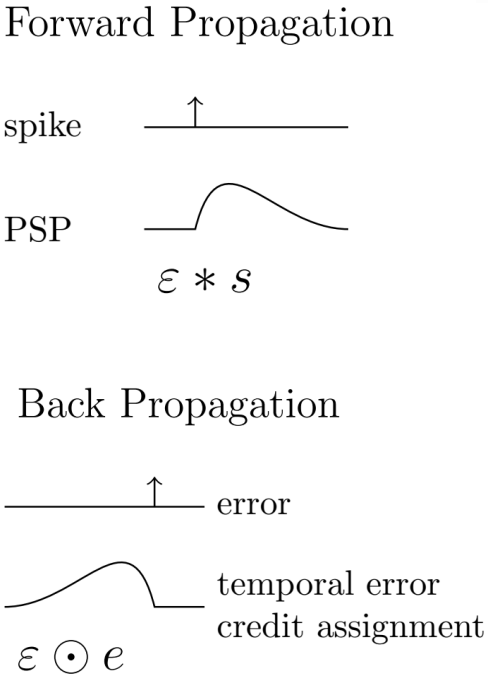
2) Error assignment
   - A neuron's state depends on its previous inputs
   - Error at the current time should be distributed to these previous inputs
   - Largely ignored by most works

SLAYER

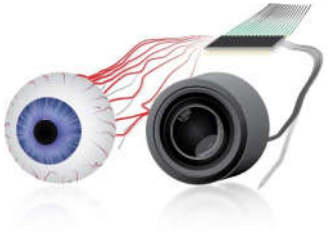


Spike LAYer Error Reassignment





SLAYER

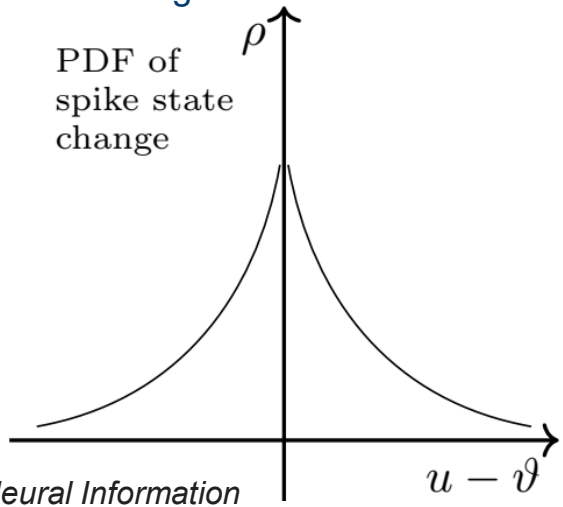


A proxy function for the derivative of the spike function (s)

- If at time τ a small perturbation $\pm\zeta$ of the Soma voltage u causes the spike s to change
- The "derivative" can be approximated as $\frac{\Delta s}{\Delta u} = \frac{\delta(t-\tau)}{\Delta\zeta}$

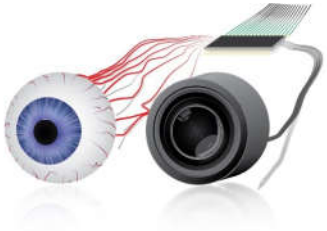
$$E[f_s'(\tau)] = E[\tfrac{\Delta s}{\Delta u}] = \rho(u(\tau) - \vartheta)$$

$$\rho(t) = \alpha \exp(-\beta\,|u(t) - \vartheta|)$$



Shrestha, S. B.; and Orchard, G.; "SLAYER: Spike Layer Error Reassignment in Time", *Neural Information Processing Systems, Montreal, Canada, Dec 2018*

SLAYER: Forward Model



SNN

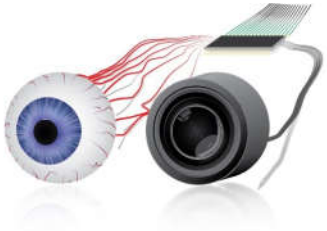
ANN

$$z^{(l+1)} = W^{(l)} a^{(l)} + b^{(l)}$$

$$a^{(l)} = f(z^{(l)})$$

Shrestha, S. B.; and Orchard, G.; "SLAYER: Spike Layer Error Reassignment in Time", *Neural Information Processing Systems, Montreal, Canada, Dec 2018*

SLAYER: Backpropagation



ANN

$$e^{(l)} = \begin{cases} \frac{\partial E}{\partial \boldsymbol{a}^{(n_l)}} & \text{if } l = n_l \\ \left(\boldsymbol{W}^{(l)}\right)^\top \boldsymbol{\delta}^{(l+1)} & \text{otherwise} \end{cases}$$

$$\boldsymbol{\delta}^{(l)} = f'(\boldsymbol{z}^{(l)}) \cdot \boldsymbol{e}^{(l)}$$

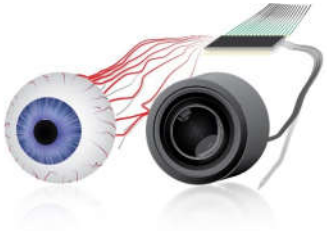
$$\nabla_{\boldsymbol{W}^{(l)}} E = \boldsymbol{\delta}^{(l+1)} \left(\boldsymbol{a}^{(l)}\right)^\top$$

SNN

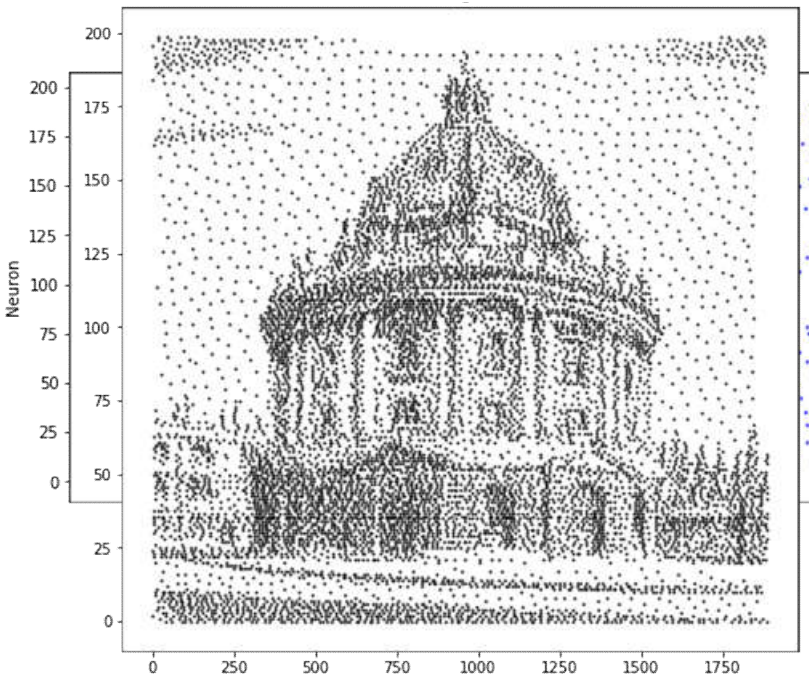
$$\boldsymbol{e}^{(l)}(t) = \begin{cases} \frac{\partial L(t)}{\partial \boldsymbol{a}^{(n_l)}} & \text{if } l = n_l \\ \left(\boldsymbol{W}^{(l)}\right)^\top \boldsymbol{\delta}^{(l+1)}(t) & \text{otherwise} \end{cases}$$

Shrestha, S. B.; and Orchard, G.; "SLAYER: Spike Layer Error Reassignment in Time", *Neural Information Processing Systems, Montreal, Canada, Dec 2018*

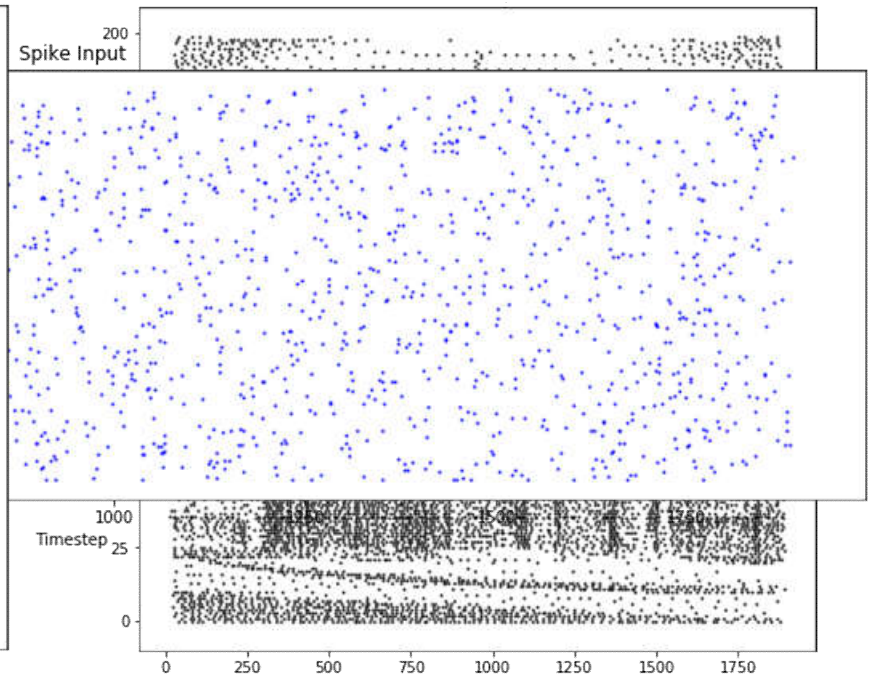
SLAYER: Radcliffe Camera



Target



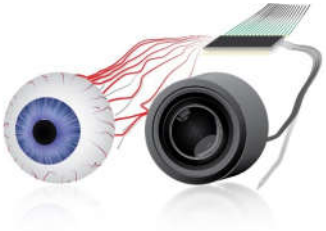
SLAYER

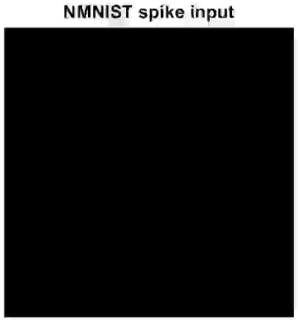


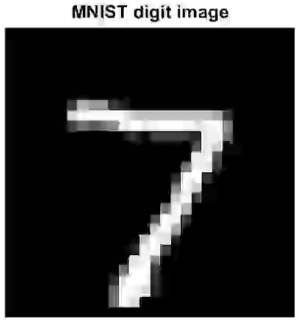
Zenke, Friedemann, and Surya Ganguli. "SuperSpike: supervised learning in multilayer spiking neural networks. "*Neural computation* 30, no. 6 (2018): 1514-1541.

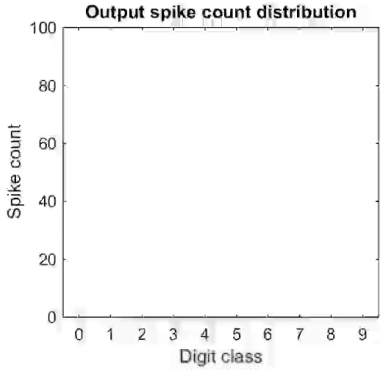
Data licensed under GNU GPL v3

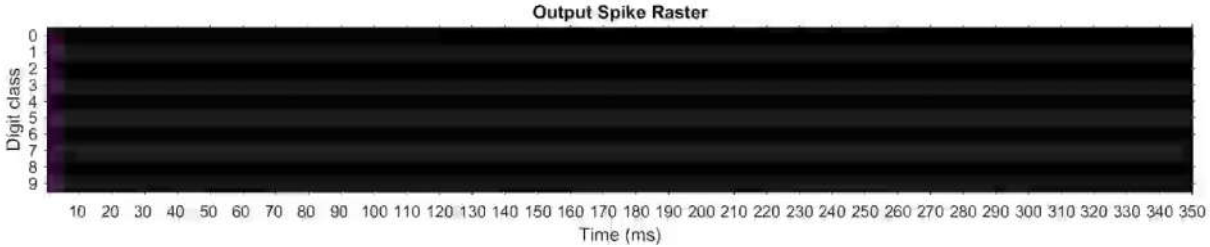
SLAYER: NMNIST





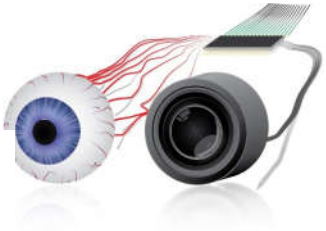


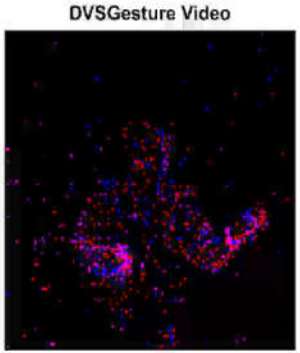




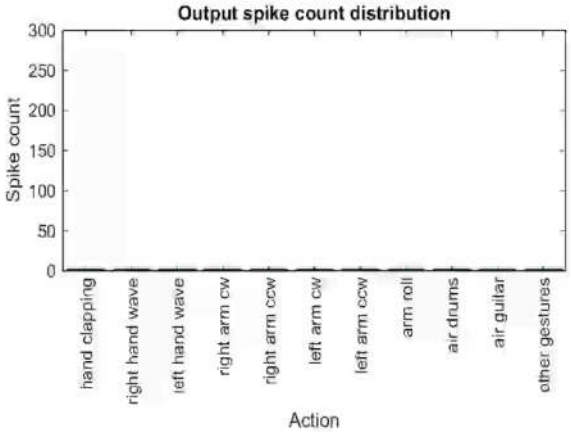
Shrestha, S. B.; and Orchard, G.; "SLAYER: Spike Layer Error Reassignment in Time", *Neural Information Processing Systems, Montreal, Canada, Dec 2018*
Orchard, G.; Cohen, G.; Jayawant, A.; and Thakor, N. “Converting Static Image Datasets to Spiking Neuromorphic Datasets Using Saccades", *Frontiers in Neuroscience,* vol. 9, pp. 437, Oct. 2015

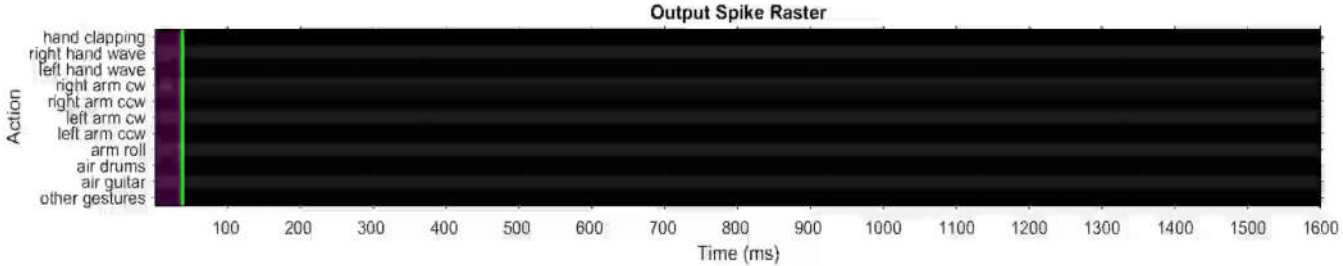
SLAYER: DVS Gesture





air guitar

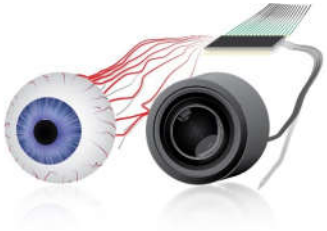




Amir, Arnon, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak et al. "A low power, fully event-based gesture recognition system." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7243-7252. 2017.

SLAYER: Results

First results on Loihi!



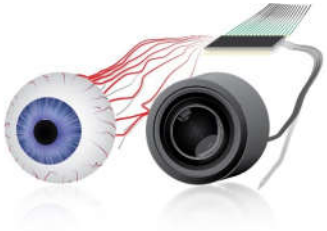
Dataset	Method	Architecture	Accuracy
MNIST	Lee et al.	28x28-800-10	99.31%
	Rueckauer et al.	SNN converted from standard ANN	**99.44%**
	SLAYER	28x28-12c5-2a-64c5-2a-10o	$99.35 \pm 0.07\%$

98.5%

92.23%

Shrestha, S. B.; and Orchard, G.; "SLAYER: Spike Layer Error Reassignment in Time", *Neural Information Processing Systems, Montreal, Canada, Dec 2018*

Loihi DVS Gesture Implementation

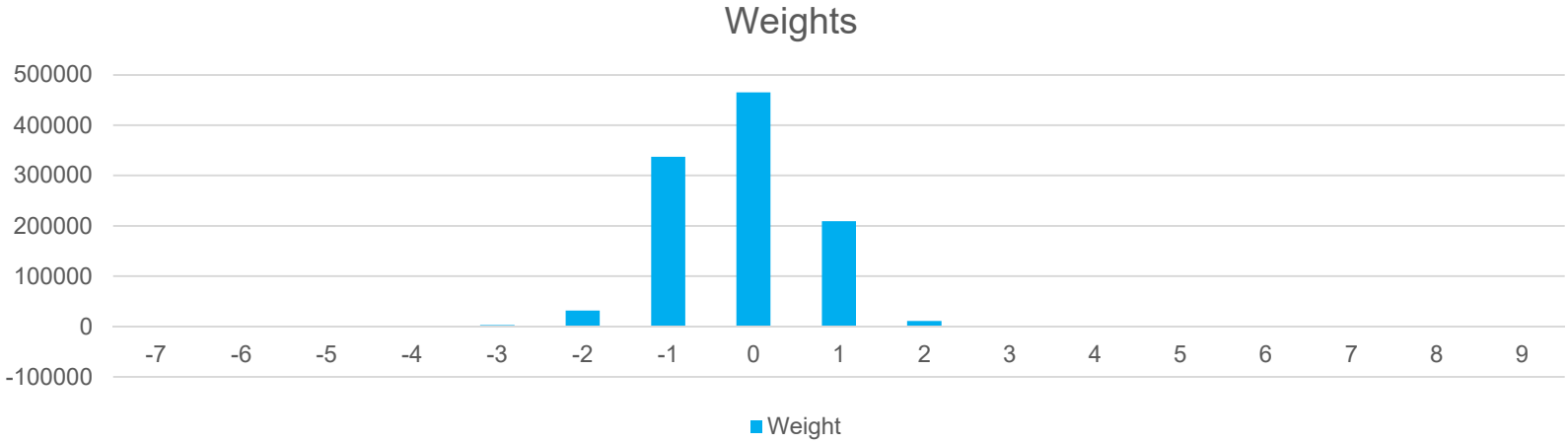


128x128x2 - 4a - 16c5z - 2a - 32c3z - 2a - 512 – 11

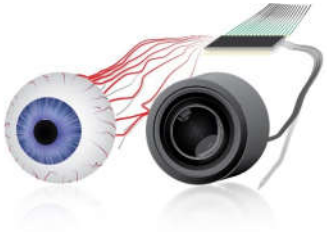
1.05M weights to learn

0.47M weights are zero

2 weights (8,9) are outside range of a 4 bit signed number



Summary

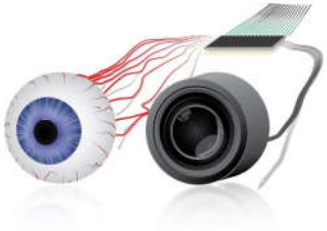


We have come a long way from hand tuning weights and delays in tiny models on tiny datasets!

SLAYER is a thoroughly derived powerful framework for configuring SNNs

- Enables learning in deep spiking networks (10+ layers)
- Can learn both weights and delays
- Allows learning of precise spike times and/or rates (depending on loss function)
- Preliminary results indicate good translation to limited precision hardware

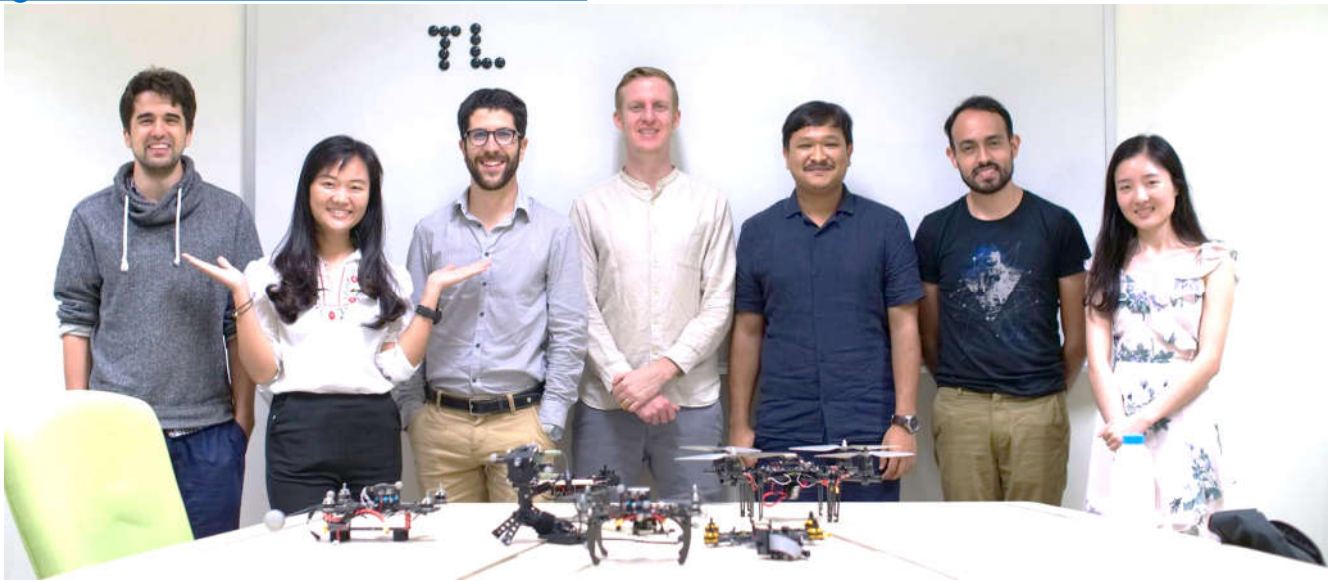
Thanks!



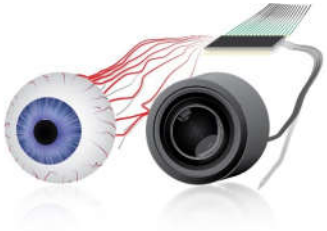
https://nusneuromorphic.github.io/labmembers/index.html

Sumit Shrestha

Luca Della Vedova



Resources



PyTorch implementation of SLAYER (use this one!)

- https://github.com/bamsumit/slayerPytorch

Singapore lab website

- https://nusneuromorphic.github.io/

Links to code (SLAYER/Hfirst/SAVME), datasets (NMNIST/NCaltech101), and papers

- www.garrickorchard.com

intel®