RE-ENGINEERING COMPUTING WITH SPIKE-BASED LEARNING: ALGORITHMS & HARDWARE

Priya Panda, Assistant Professor

Electrical engineering

Yale University

priya.panda@yale.edu

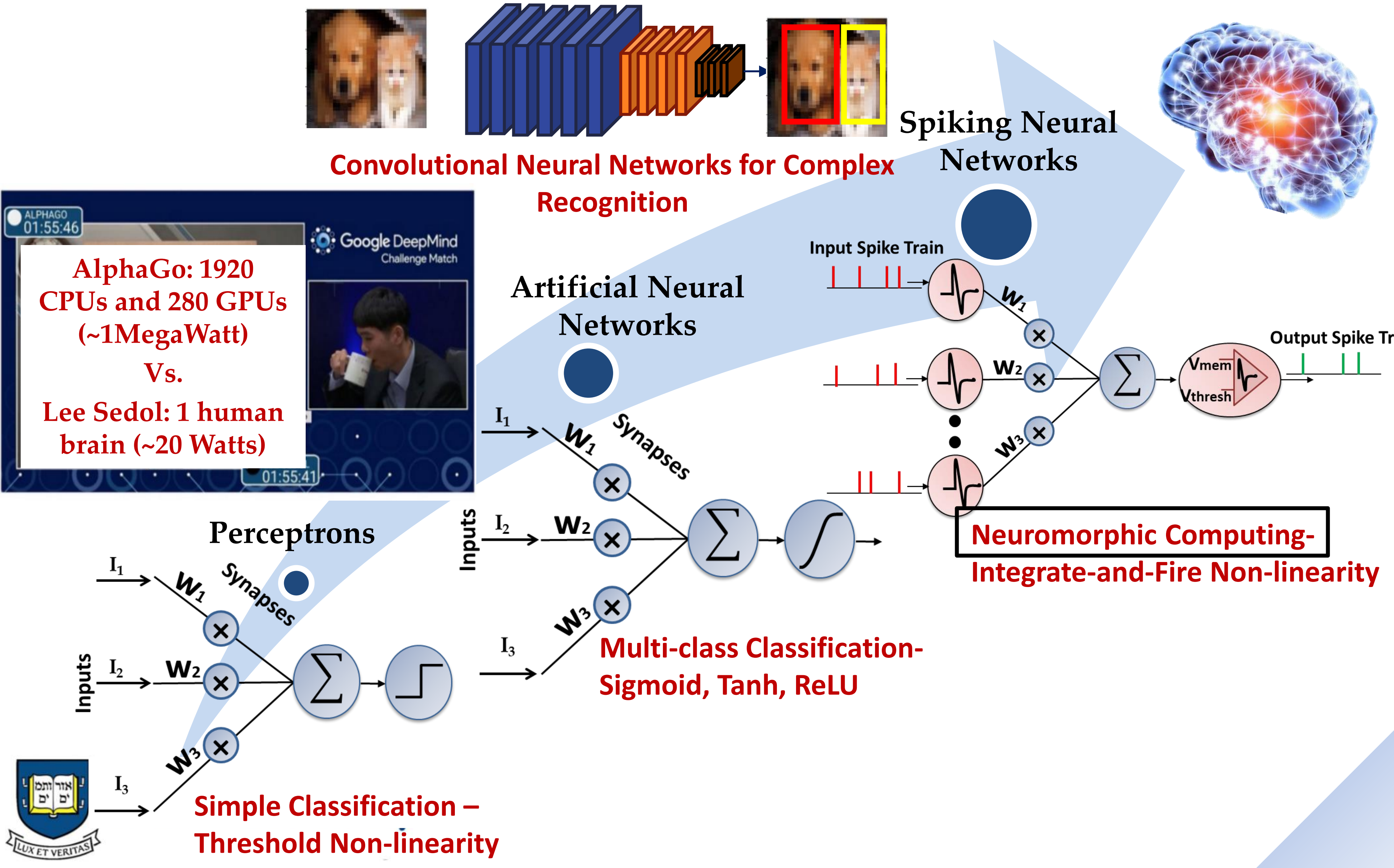
Yale University

INTELLIGENT COMPUTING LAB

https://intelligentcomputinglab.yale.edu/

Acknowledgement:
Prof. Kaushik Roy
CBRIC, Purdue University

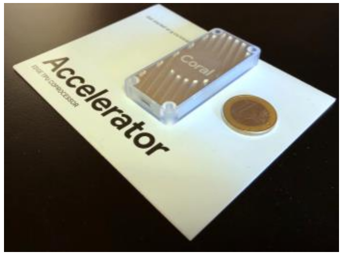
Neural Networks: Different Levels of Bio-fidelity



Efficiency Gap in Edge Devices

- Case study: Object recognition in a smart glass with a state-of-the-art accelerator

+



Google Edge TPU

Retinanet DNN* on a smart glass

Performance	
Frames/sec	13.3
Battery Life	
Energy/op	0.5 pJ/op
Energy/frame	0.15 J/frame
Time-to-die (2.1WH)	**64 mins**

*300 GOPs/inference

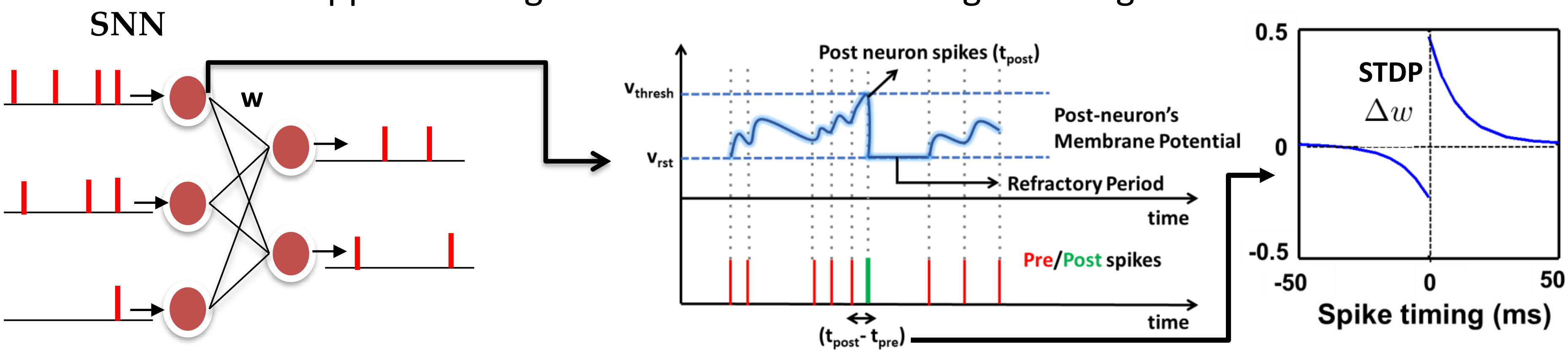
Where do the in-efficiencies come from?

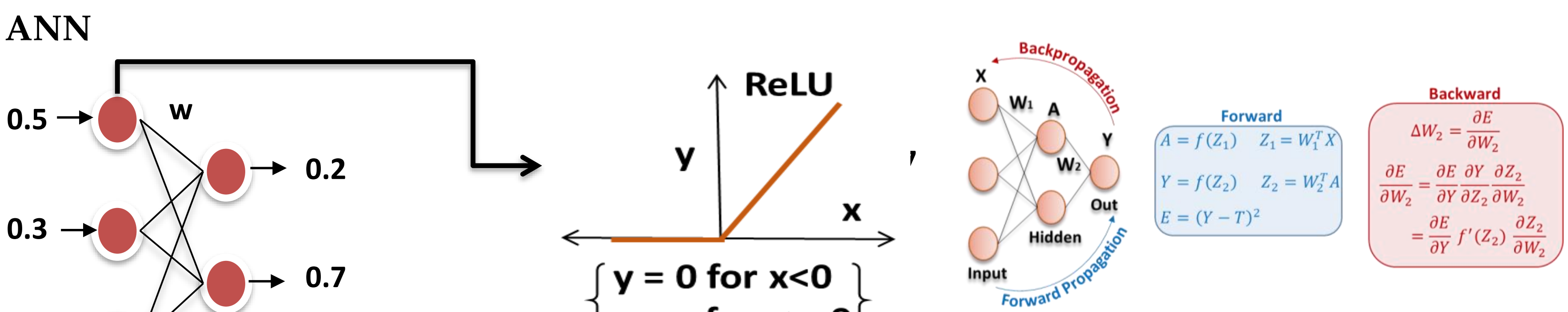
Algorithms — Hardware Architecture — Circuits and Devices

Ref: Venkataramani, S., Roy, K. and Raghunathan, A. “Efficient embedded learning for IoT devices.” In *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 308-311. IEEE.

SNN vs. ANN: Fundamental Differences

- Use spike timing dependent plasticity to perform local layerwise learning
- Use approximate gradient descent with straight through estimation





Backpropagation Training with Gradient Descent

Yale University

Can we efficiently train deep SNNs?

SNN Training

Binary Activation

STDP Learning

Pros: Unsupervised local learning

Cons: Limited accuracy and shallow networks

Reference	MNIST Accuracy
Cook *et al.* Frontiers 2015 (ETH Zurich)	95.00%
Masquelier *et al.* Neural Networks 2017	98.40%
Lee *et al.* TCDS 2018	91.10%

ANN-SNN Conv

Pros: Takes advantage of standard ANN training

Cons: Conversion limited by constraints

Reference	MNIST Accuracy
Pfeiffer *et al.* IJCNN 2015 (ETH Zurich)	99.10%
Eliasmith *et al.* arXiv 2016 (U Waterloo)	99.12%
Liu *et al.* Frontiers 2017 (ETH Zurich)	99.44%

Backprop in SNN

Pros: Higher accuracy

Cons: Limited scalability, Discontinuous spike activities

Reference	MNIST Accuracy
Pfeiffer *et al.* Frontiers 2016 (ETH Zurich)	99.31%
Shi *et al.* Frontiers 2018 (Tsinghua)	99.42%
Zhang *et al.* arXiv 2018 (TAMU)	99.49%

Stochastic STDP

Pros: Unsupervised local learning with binary synaptic weights

Cons: Limited accuracy

Reference	MNIST Accuracy
Gamrat *et al.* Proceedings of the IEEE '15	60.00%
Yousefzadeh *et al.* Frontiers 2018	95.70%
Roy *et al.* (Frontiers, 2019)	98.54%

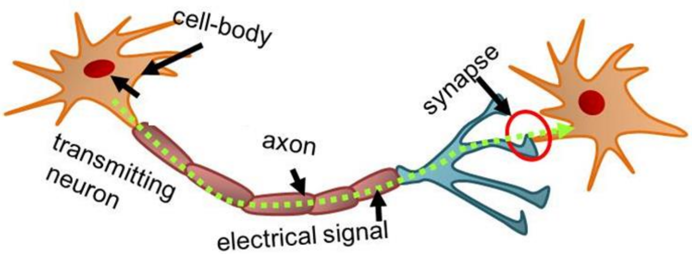
Roy *et al*. Frontiers 2019

Panda, Roy *et al*. Frontiers 2018

Panda, Roy *et al*. Frontiers 2019

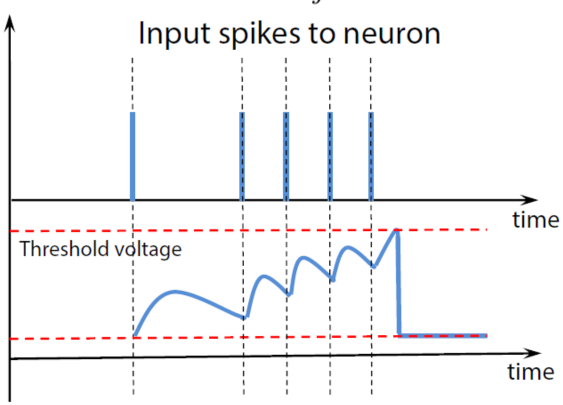
Neuron and Synaptic Models: STDP (local) Learning

Spiking Neuron



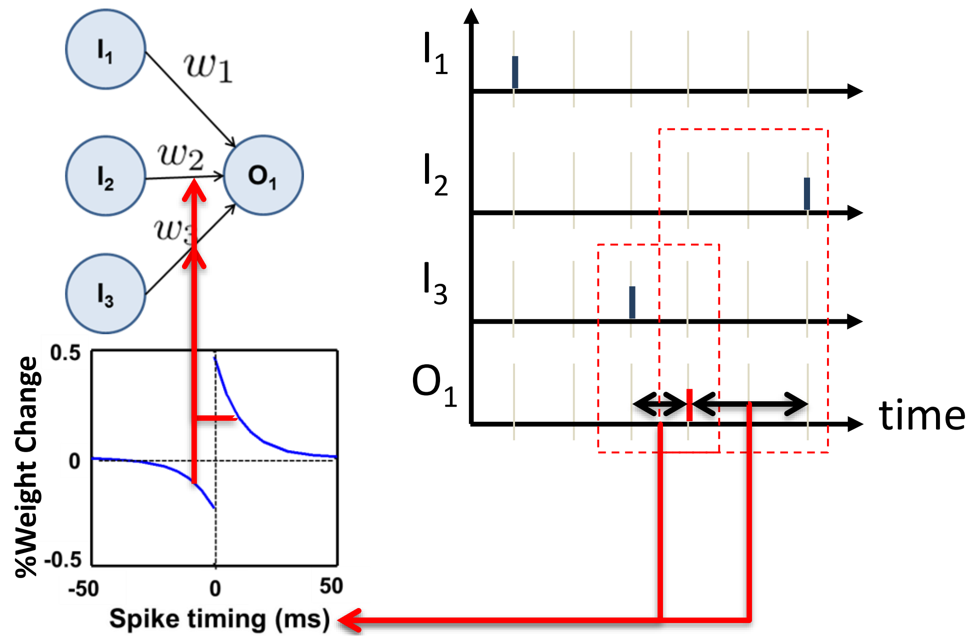
LIF Equation:

$$C\frac{dV}{dt} = -\frac{V}{R} + \sum_j w_j I_{post,j}$$



Neuron Membrane Potential

Synaptic Learning



Weight Update Equations

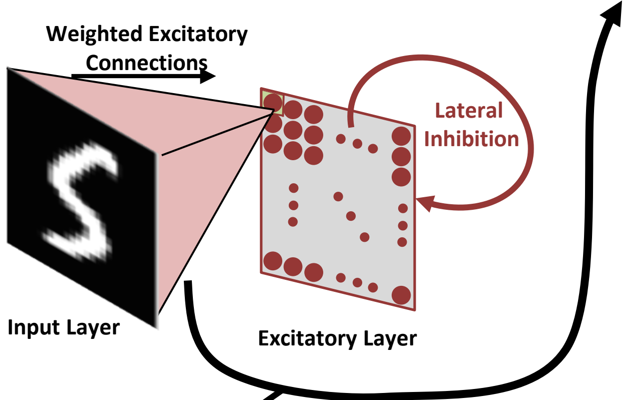
$$w_i^{new} = w_i^{old} + \Delta w(t_i) \times w_{max} \qquad i = 1, 2, 3$$

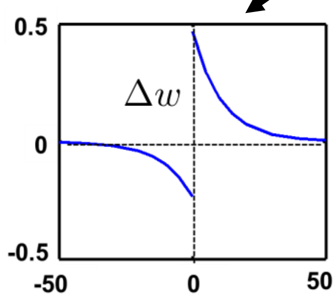
Strength of the synapse should increase (decrease) as post and pre neurons appear to be temporally correlated (uncorrelated)

STDP based Unsupervised Recognition: Overview

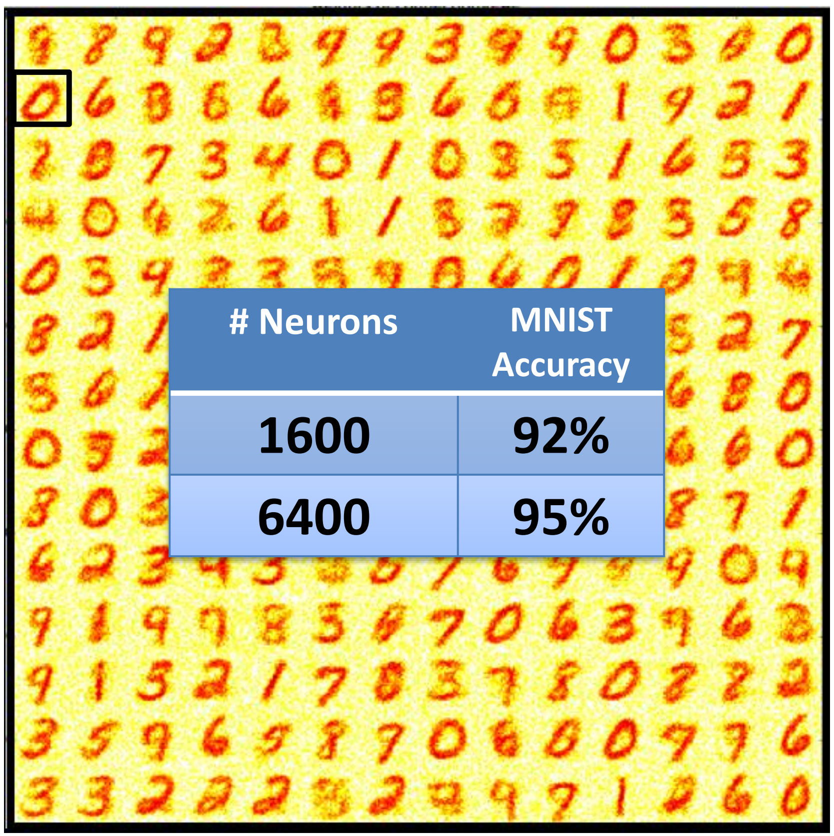
- Excitatory neurons learn general input representations in the synaptic weights

Two-Layer Topology





Spike Timing Dependent Plasticity



# Neurons	MNIST Accuracy
1600	92%
6400	95%

Ref: Diehl, P.U. and Cook, M., 2015. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in computational neuroscience*, *9*, p.99.

Yale University

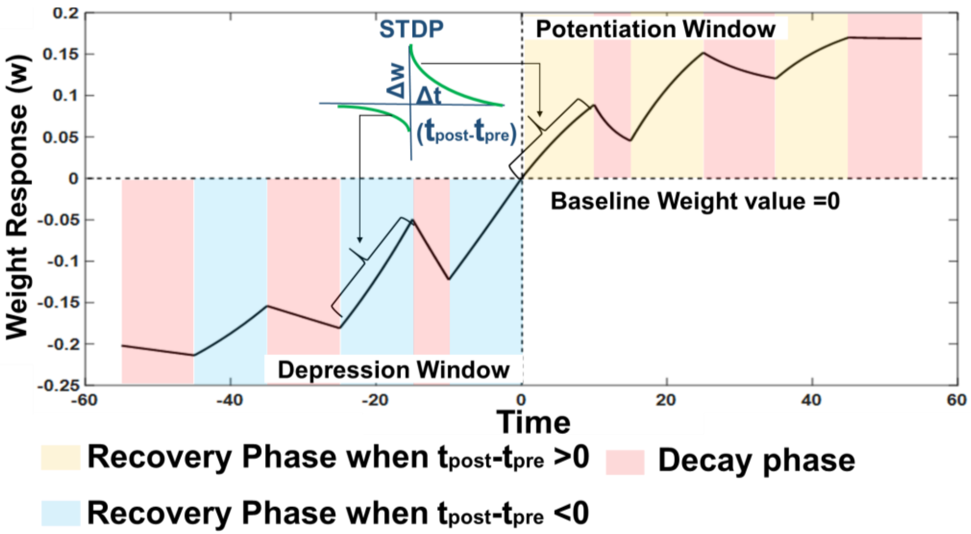
Can we utilize temporal property to do something more?

Lifelong Learning

- How do we learn continually with resource constraints?
  - Can we reorganize or 'forget' something to accommodate new data coming in?

Yale University

Learning to Forget: Adaptive Synaptic Plasticity (ASP)



Recovery Phase

$$\Delta w = \eta(t)[(Pre_{rec} - offset) - k_{const}/2^{Pre_{acc}}]$$

$$\eta(t) \in 1/Post(t)$$

Decay Phase

$$\tau_{leak}\frac{dw}{dt} = -\alpha_{exp}w \quad \tau_{leak}\frac{dw}{dt} = -\alpha_{lin}$$

$$\tau_{leak} \in Post(t), (v_{thresh} + \theta)$$

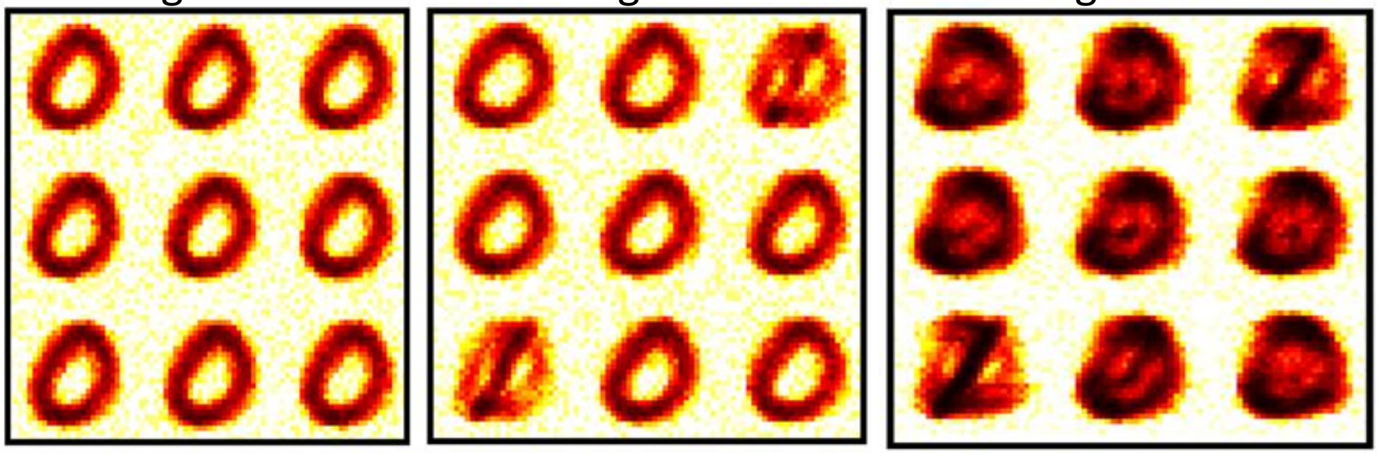
- Neuron that has learnt patterns well will have lower learning rate in recovery phase
- Stable learning, prevents neuron from quick adaptation
- Input significance proportional to the number of training patterns shown
- Neurons learning significant patterns decay or forget less

Ref: Panda, P., Allred, J.M., Ramanathan, S. and Roy, K., 2017. Asp: Learning to forget with adaptive synaptic plasticity in spiking neural networks. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, *8*(1), pp.51-64.
Ref: Zuo, F., Panda, P., …, Roy, K., and Ramanathan, S., 2017. Habituation based synaptic plasticity and organismic learning in a quantum perovskite. *Nature communications*, *8*(1), p.240.

Standard STDP vs ASP

SNN learnt with STDP

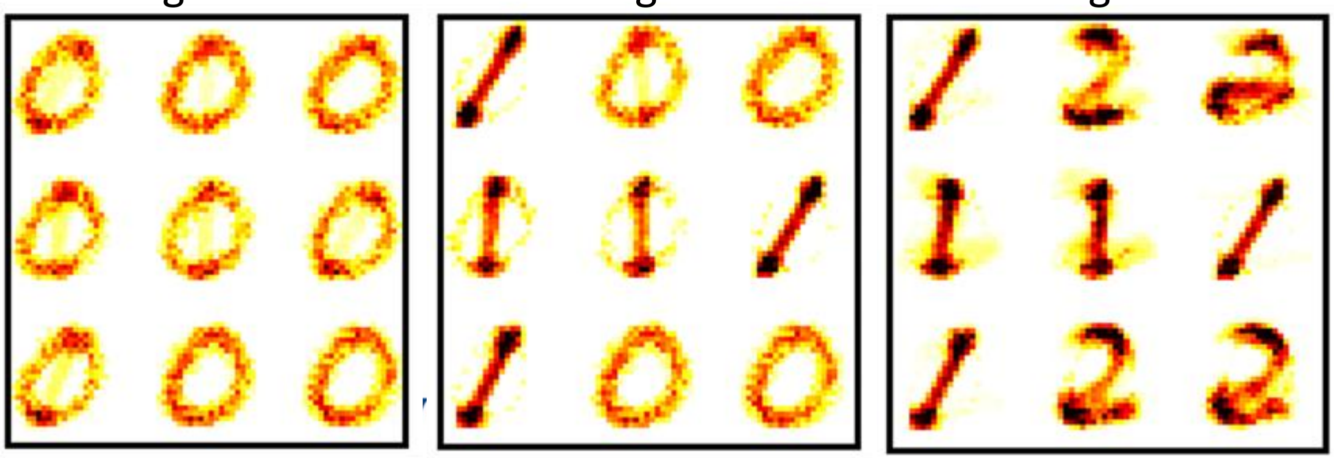
Digit 0 Digit 1 Digit 2



Catastrophic Forgetting

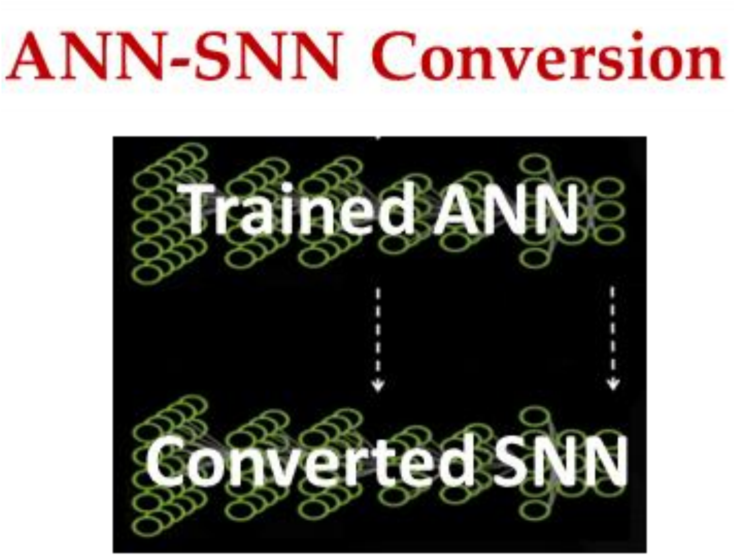
SNN learnt with ASP

Digit 0 Digit 1 Digit 2



Controlled Forgetting

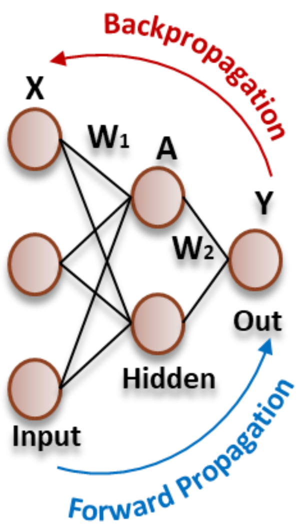
Approaching large-scale SNNs for complex tasks



- Scaling up to ImageNet like tasks while being energy-efficient
- Burden of training is gone!
  – Use existing ML frameworks, PyTorch, TensorFlow etc for training
  – GOAL: Given learnt weights, need to transfer ReLU activations to Integrate-and-Fire activations

Learning with Spike-based Backpropagation

Global Supervised Gradient Descent Learning



Forward

$A = f(Z_1) \qquad Z_1 = W_1^T X$

$Y = f(Z_2) \qquad Z_2 = W_2^T A$

$E = (Y - T)^2$

Backward

$$\Delta W_2 = \frac{\partial E}{\partial W_2}$$

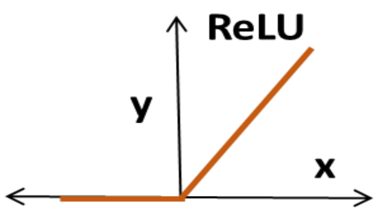
$$\frac{\partial E}{\partial W_2} = \frac{\partial E}{\partial Y}\frac{\partial Y}{\partial Z_2}\frac{\partial Z_2}{\partial W_2}$$

$$= \frac{\partial E}{\partial Y} f'(Z_2) \frac{\partial Z_2}{\partial W_2}$$

$f'(Z_2)$ exists if f is **continuous**

f = **LIF** (discontinous) model for SNN

$\boldsymbol{f_{approx}} \approx$ **Approximate LIF for conducting spike based backpropagation**



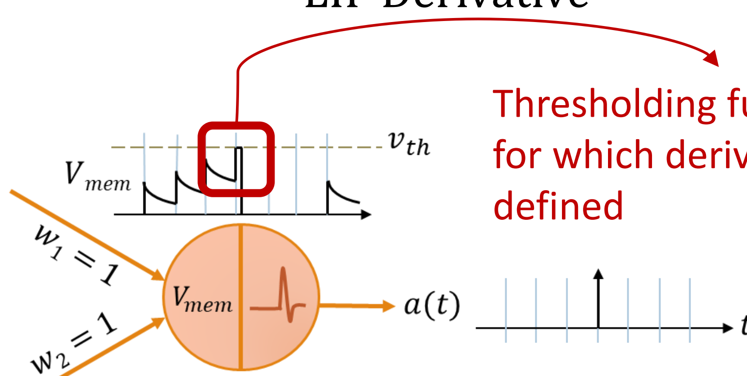
$$\begin{cases} y = 0 \text{ for } x<0 \\ y = x \text{ for } x>=0 \end{cases}$$

$\mathbf{y = x}$

Relu Derivative

$$\frac{\partial y}{\partial x} = 1$$

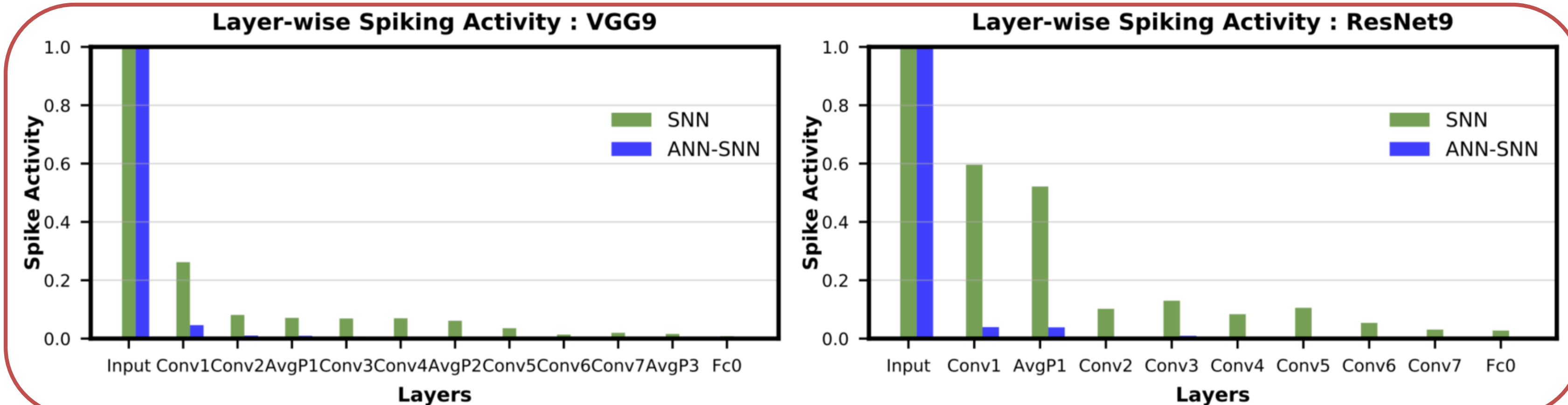
LIF Derivative



Thresholding functionality for which derivative is not defined

Surrogate Backpropagation CIFAR-10 Results

- Spiking activity reduces exponentially with network depth



Network Model	Comparison Criterion	SNN*	Converted SNN*	ANN
VGG9	#Operation	3.61x	28.18x	1x
	Efficiency	8.86x	1.13x	1x
ResNet9	#Operation	5.06x	11.94x	1x
	Efficiency	6.32x	2.68x	1x
ResNet11	#Operation	2.09x	7.26x	1x
	Efficiency	15.31x	4.4x	1x

Deep SNNs can offer 6-15× compute energy efficiency

- Estimated using the #synaptic operations per layer (ACC for SNNs and MAC for ANNs)

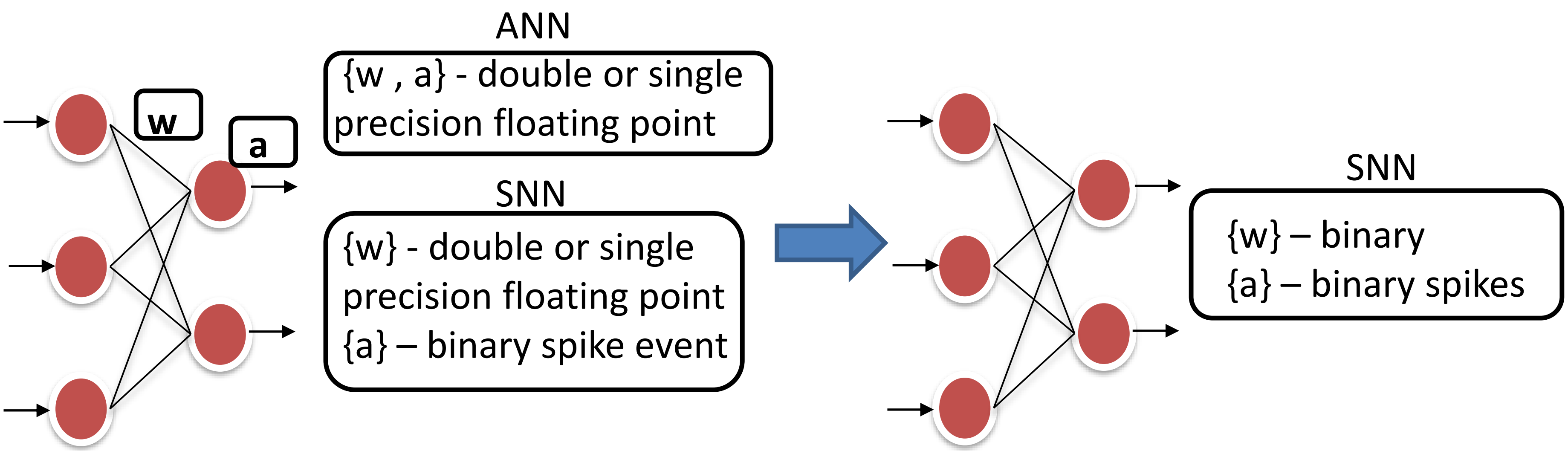
Ref: Lee, C., Sarwar, S. S., G.Srinivasan, P. Panda, & Roy, K. (2019). Enabling Spike-based Backpropagation in State-of-the-art Deep Neural Network Architectures. *arXiv preprint arXiv:1903.06379*.

Hybrid Training (Conversion + Backpropagation)

Architecture	ANN	ANN-SNN	ANN-SNN	SNN
CIFAR10				
VGG5	87.88%	87.64% (T=2500)	84.56% (T=75)	**86.91% (T=75)**
VGG9	91.45%	90.98% (T=2500)	87.31% (T=100)	**90.54% (T=100)**
VGG16	92.81%	92.48% (T=2500)	90.2% (T=100)	**91.13% (T=100)**
ResNet8	91.35%	91.12% (T=2500)	89.5% (T=200)	**91.35% (T=200)**
ResNet20	93.15%	92.94% (T=2500)	91.12% (T=250)	**92.22% (T=250)**
CIFAR100				
VGG11	71.21%	70.94% (T=2500)	65.52% (T=125)	**67.87% (T=125)**
ImageNet				
ResNet34	70.2%	65.1% (T=2500)	56.87% (T=250)	**61.48% (T=250)**
VGG16	69.35%	68.12% (T=2500)	62.73% (T=250)	**65.19% (T=250)**

N. Rathi et al., Enabling Deep Spiking Neural Networks with Hybrid Conversion and Spike Timing Dependent Backpropagation, To Appear in ICLR 2020. https://openreview.net/forum?id=B1xSperKvH

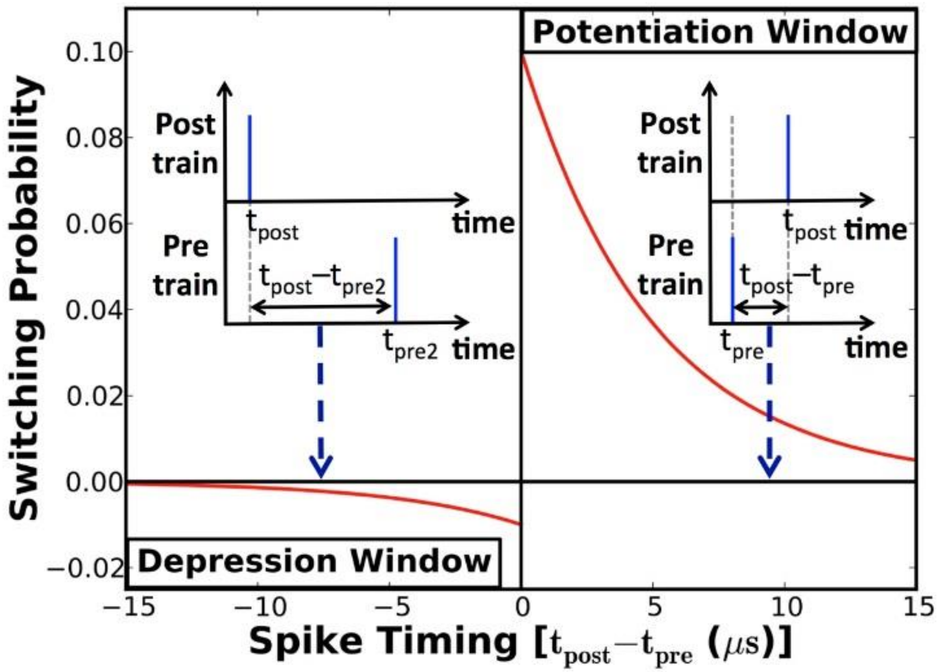
Stochastic-Binary SNNs



- Binarized SNNs with low-precision weights
  - Stochastic STDP training for binary weight updates
  - Backpropagation too!
- Advantage
  - Compatible with low-energy learning and inference on edge devices
  - Even suitable for emerging-technologies

Yale University

Stochastic STDP



STDP Learning

$$\Delta w \approx \exp(t_{post} - t_{pre})$$

Stochastic STDP Learning

$$p(\Delta w) \approx \exp(t_{post} - t_{pre})$$

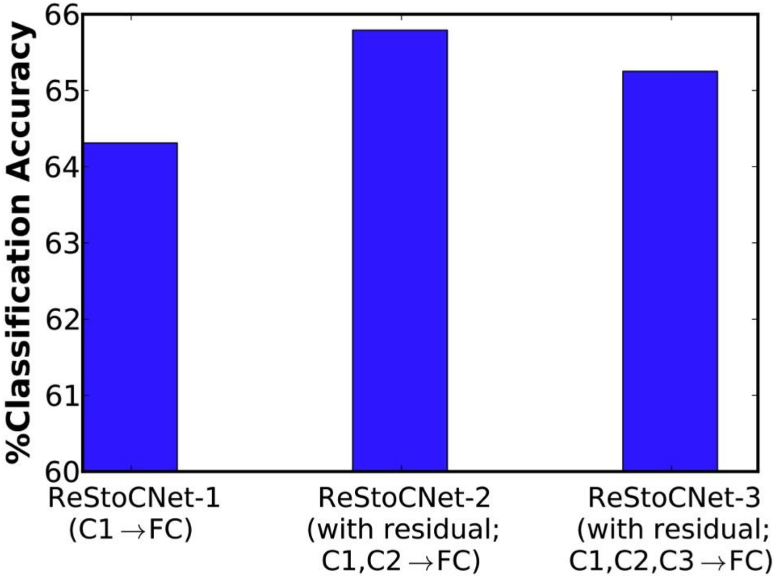
$$\underbrace{\begin{array}{l} if\, t_{post} - t_{pre} \text{ is small, p } \rightarrow 1 \;\Rightarrow \mathrm{w} = \mathrm{w}_{\max} \\ else, p \rightarrow 0 \quad \Rightarrow \text{w does not change} \end{array}}_{\text{Potentiation Rules}}$$

- Switching probability $p(\Delta w)$ of a binary synapse depends on the degree of timing correlation between pre- and post-spikes.

Yale

Ref: Srinivasan, G., Sengupta, A. and Roy, K., 2016. Magnetic tunnel junction based long-term short-term stochastic synapse for a spiking neural network with on-chip STDP learning. *Scientific reports*, *6*, p.29545.

Results on the CIFAR-10 dataset



- Trained up to 5-layer deep ReStoCNet (36C3-36C3-36C3-1024FC-10FC) on CIFAR-10
  - Classification accuracy saturates for depth greater than 4 layers
  - STDP has limitations! Cannot extract finer features in deeper layers.

Model	Training Method	Accuracy (%)
Panda and Roy, 2016	Regenerative learning with backpropagation	70.16
Ferré et al., 2018	STDP + backpropagation	71.20
Srinivasan and Roy, 2019	Stoch-STDP + backpropagation	66.23

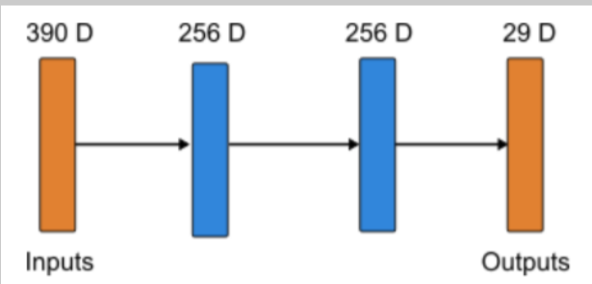
21× kernel memory compression (lower for deeper networks)

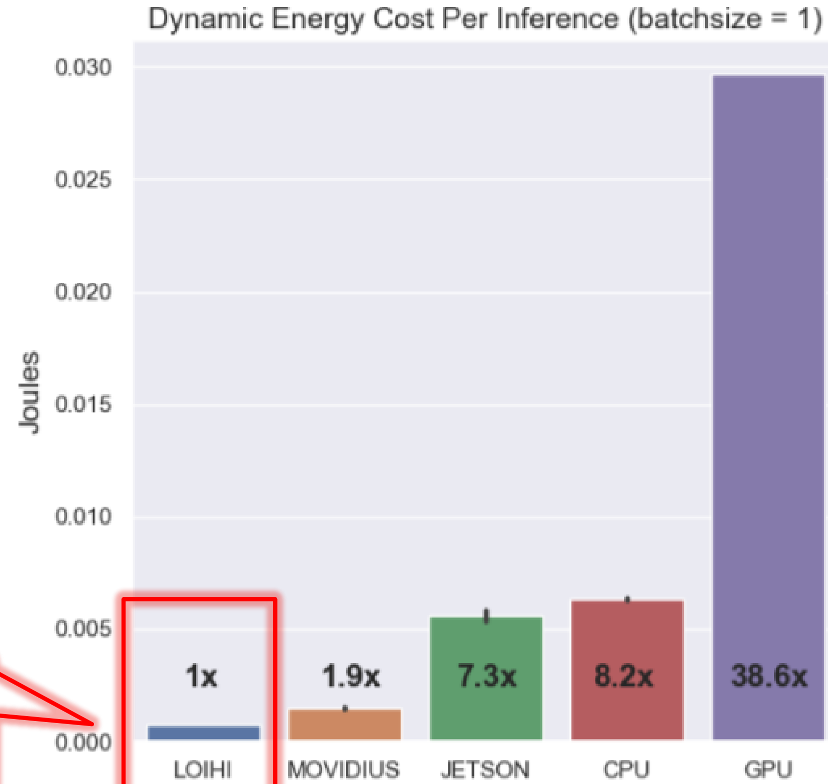
Yale

Ref: Srinivasan, G. and Roy, K., 2019. ReStoCNet: Residual Stochastic Binary Convolutional Spiking Neural Network for Memory-Efficient Neuromorphic Computing. *Frontiers in Neuroscience*, *13*, p.189.

SNN Hardware Efficiency: A Benchmarking Study

- Sparse spike-based event-driven computing capability can enable energy-efficient on-chip neuromorphic computing
- **Case study:** Keyword spotting task using SNN on Intel's Loihi

Specifications for benchmarking study	
Network size	
Comparison Hardware	CPU (Xeon E5-2630), GPU (Quadro K4000), Jetson TX1, Movidius NCS

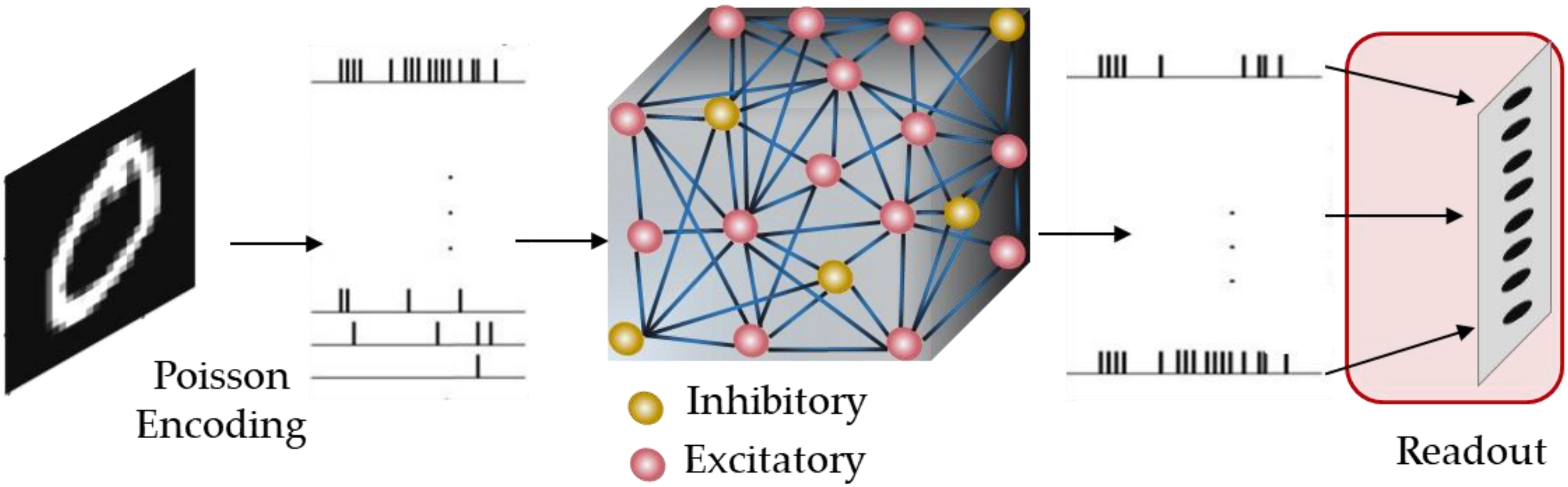


SNN on Loihi outperforms ANN on GPU/CPU on energy per inference while maintaining nearly-equivalent inference accuracy

Yale University

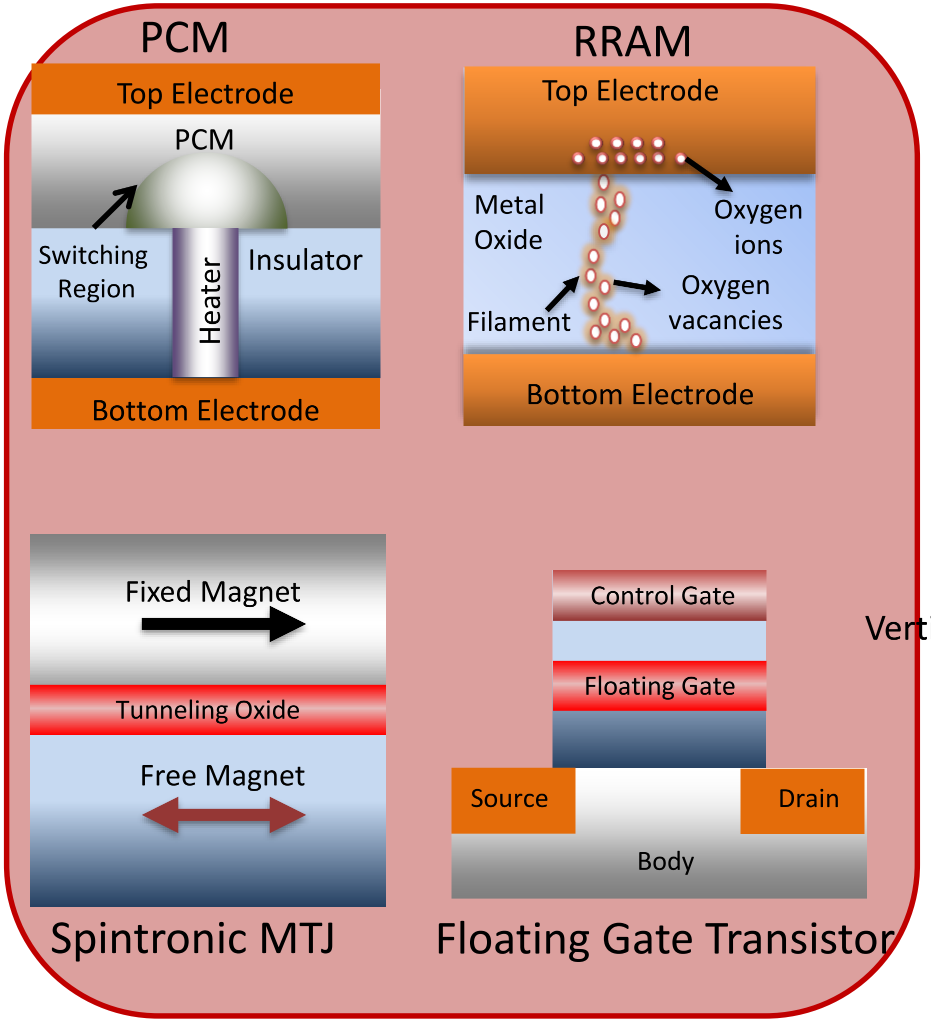
Ref: Peter Blouw et al. "Benchmarking Keyword Spotting Efficiency on Neuromorphic Hardware", arXiv preprint arXiv:1812.01739 (2018)

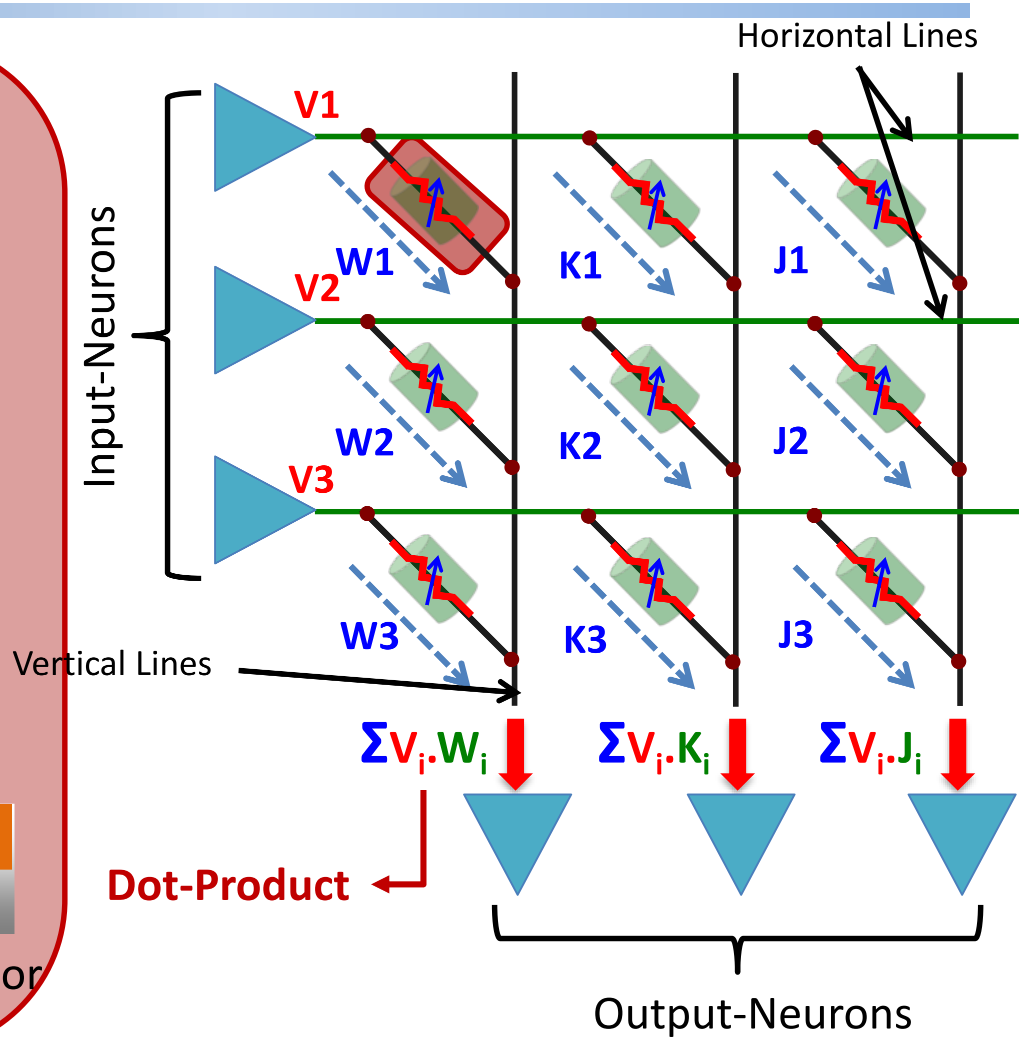
LIQUID STATE MACHINES



1) Panda, P., & Roy, K. (2017). Learning to generate sequences with combination of hebbian and non-hebbian plasticity in recurrent spiking neural networks. *Frontiers in neuroscience*, *11*, 693.
2) Panda, P., & Srinivasa, N. (2018). Learning to recognize actions from limited training examples using a recurrent spiking neural model. *Frontiers in neuroscience*, *12*, 126.
3) Wijesinghe, P., Srinivasan, G., Panda, P., & Roy, K. (2019). Analysis of Liquid Ensembles for Enhancing the Performance and Accuracy of Liquid State Machines. *Frontiers in neuroscience*, *13*, 504.
4) Srinivasan, G., Panda, P., & Roy, K. (2018). Spilinc: spiking liquid-ensemble computing for unsupervised speech and image recognition. *Frontiers in neuroscience*, *12*, 524.

Post-CMOS Devices as Synaptic Memory Elements





Yale University

Where do SNNs fit?

- Energy-efficiency is the primary advantage at this point!
- Potential for Edge Computing (training/inference)
- Emerging devices can effectively harness spiking synapse/neuron functionality

Algorithm-Hardware co-design guided by biological principles can be the pathway for next-generation neural computing.

Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda.
"Towards spike-based machine intelligence with neuromorphic computing"
Nature 575.7784 (2019): 607-617.

THANK YOU!**

**A majority of slides comprise of the work done by Priya Panda while she was a PhD student at Purdue University working with Prof. Kaushik Roy.

Yale University

Prof. Priya Panda

INTELLIGENT COMPUTING LAB

https://intelligentcomputinglab.yale.edu/