ETH zürich

# Optimal ANN-SNN Conversion Applied to DVS Gesture Data

Ravi Srinivasan, Andrea Pinto, Robin Chan
Supervisor: Dr. Martino Sorbaro

## 1 Introduction

**Spiking Neural Networks** (SNNs) are inspired by the dynamics of biological neurons in human brain. Neurons in SNNs do not transmit information at each cycle, as it happens in classical Artificial Neural Networks (ANNs), but rather fires and transmits a spike whenever neuron's membrane action potential reaches a certain threshold *θ*.

We aim (i) to reproduce the paper introducing optimal ANN-SNN conversion [1] and (ii) to apply it to the IBM DVSGestures data.

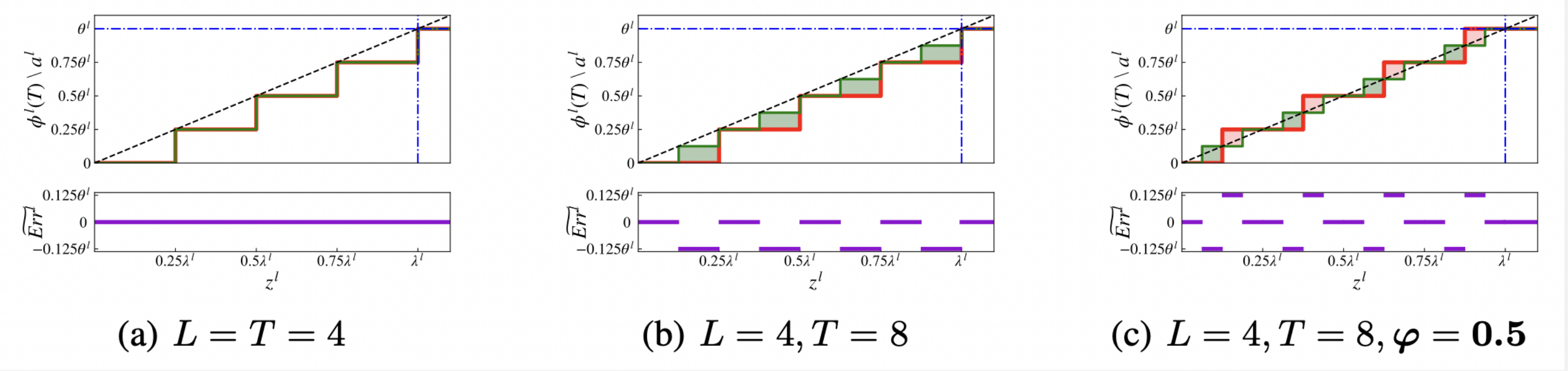
## 2 Method Overview

**SNN IF-Neuron Model.** Integrate-and-fire (IF-)neurons potential at some time $t$ is given as the sum of (i) its potential before firing and (ii) the weighted input of the previous layer. Whenever the neuron's potential $\boldsymbol{m}^l(t)$ passes the firing threshold $\theta^l$, the neuron fires a new spike (*H* function) as input into the next layer.

$$\boldsymbol{m}^l(t) = \boldsymbol{v}^l(t-1) + \boldsymbol{W}^l x^{l-1} \qquad \boldsymbol{s}^l(t) = H(\boldsymbol{m}^l(t) - \boldsymbol{\theta}^l)$$

This activation function is non-differentiable and can therefore not be backpropagated through. **ANN-SNN conversion** addresses this by mapping the SNN average postsynaptic potential $\phi^l(t)$ to the ANN activation value $a^l$ by training the ANN with a clip-floor-shift activation function to simultaneously learn the ANN and SNN weights:

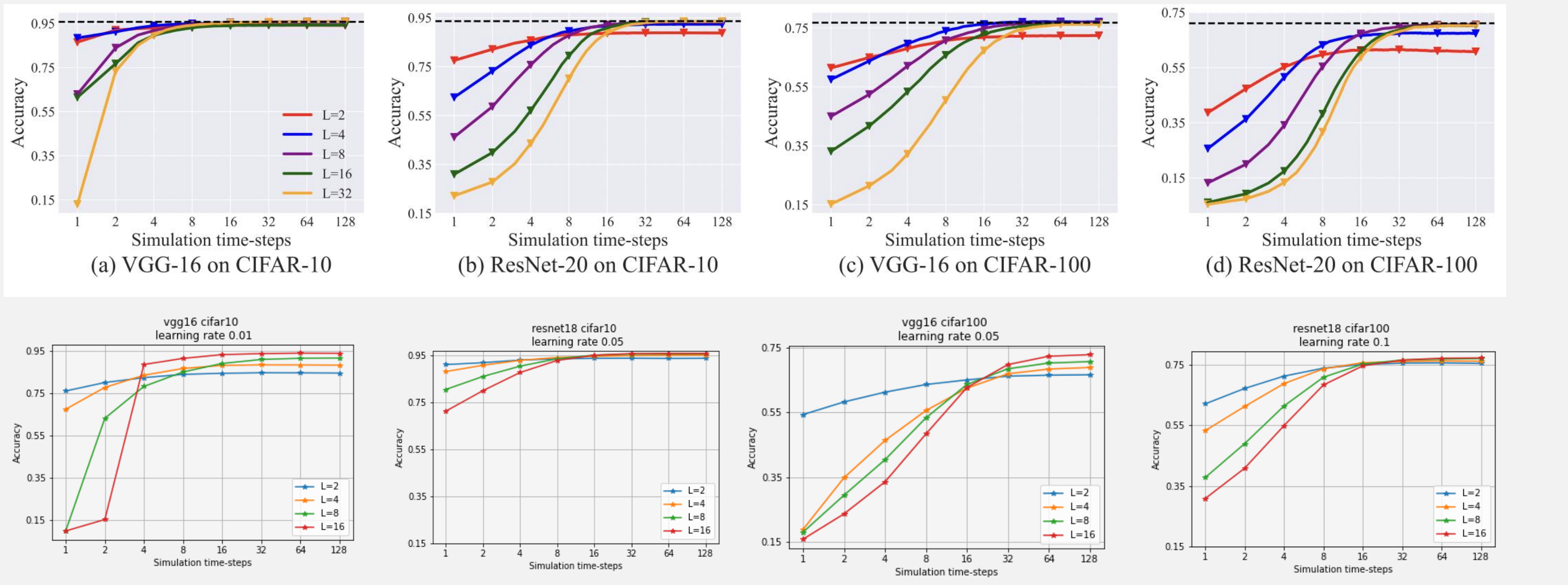
$$\boldsymbol{a}^l = \hat{h}\left(\boldsymbol{z}^l\right) = \lambda^l \mathrm{clip}\left(\frac{1}{L}\lfloor \boldsymbol{z}^l L/\lambda^l + \boldsymbol{\varphi} \rfloor, 0, 1\right)$$


(a) $L = T = 4$ (b) $L = 4, T = 8$ (c) $L = 4, T = 8, \boldsymbol{\varphi} = \mathbf{0.5}$

Comparison of SNN output and ANN output with the same inputs

## 3. Reproducibility Results

Results from the paper were able to be reproduced for both VGG and ResNet models on both CIFAR10 and CIFAR100 datasets as well as the presented values of the quantization step L hyperparameter.

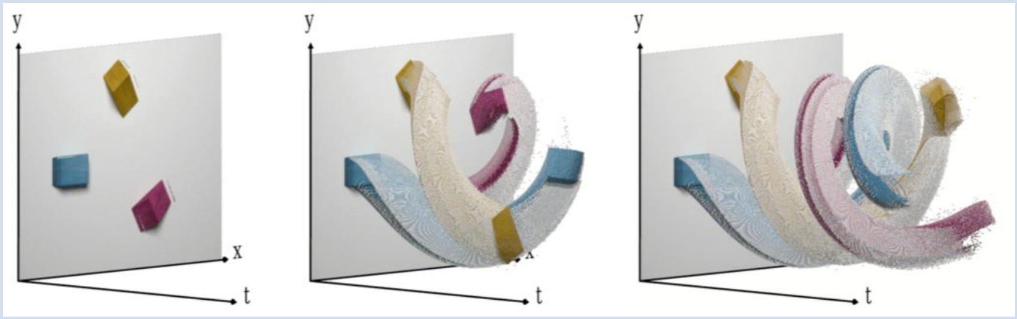

(a) VGG-16 on CIFAR-10 (b) ResNet-20 on CIFAR-10 (c) VGG-16 on CIFAR-100 (d) ResNet-20 on CIFAR-100

Results from the paper are in the first row whereas ours in the second row

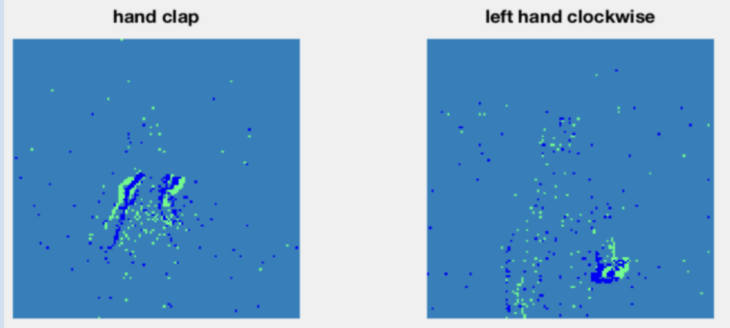
## 4. Event-Based Data (DVSGestures)

**Event cameras** such as Dynamic Vision Sensor (DVS) output pixel-level brightness changes instead of intensity frames like conventional digital cameras.

**IBM DVSGestures** is a dataset containing two-channel 128x128 series of images capturing positive (+) and negative (-) polarity. One frame is the aggregated sum of a temporal slice of the DVS polarities from the series of images.



https://tonic.readthedocs.io/en/latest/getting_started/intro-event-cameras.html



https://research.ibm.com/interactive/dvsgesture/

## 5. SNNs & Event-based Data

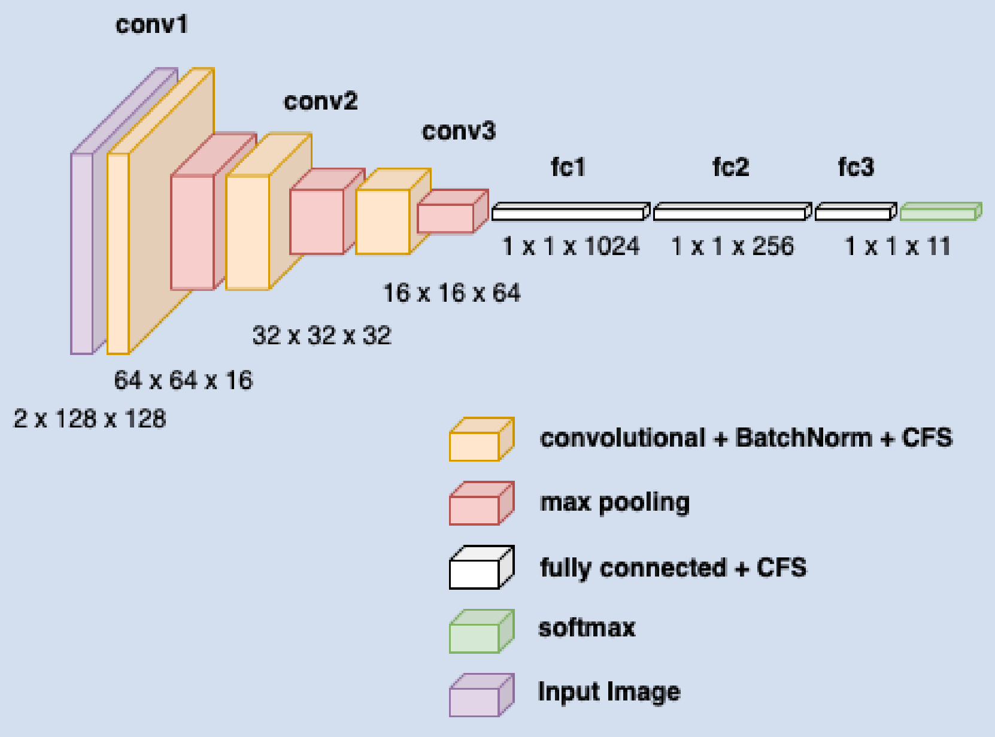
Inference on event-based data is fast (pseudo-simultaneous) and energy-efficient on SNNs running on neuromorphic hardware, since spikes start to propagate to output layers immediately.

## 6. SNN Applied to DVSGestures

We tested the proposed SNN framework on actual DVS Data. We accumulated 1000ms of data for each (normalized) frames, which we passed to a common CNN architecture [4] we built for training. We trained for 25 epochs, with **L=500** and learning rate 0.001 using Adam. On the testing set we obtain an ANN accuracy of **0.86**.

We then swapped the layer units with IF-Neurons and tested the obtained SNN by passing 1ms frames from the DVS data.

The best SNN accuracy we got was **0.76** and it was obtained by the model after 500 time steps of the simulation (matching with L).



Proposed CNN architecture

## 7. Live Gesture Classification

**Live gesture classification.** When trying to do live gesture recognition using a DVS camera, the model struggled to recognize human gestures. This might be because we need more sophisticated filtering of the polarity images sent by the DVS camera.

One issue we observed was the saturation of the neurons when using a long buffer time (long accumulation of spikes) and low accuracy when using a short buffer time. This may be an issue related to the architecture of the network we used.

## References

1. Tong Bu, Wei Fang, Jianhao Ding, Penglin Dai, Zhaofei Yu, Tiejun Huang (2022). Optimal ANN-SNN Conversion for High-accuracy and Ultra-low-latency Spiking Neural Networks
2. Tonic Library https://tonic.readthedocs.io/en/latest
3. IBM DVS Gesture Dataset https://research.ibm.com/interactive/dvsgesture/
4. Learning from Event Cameras with Sparse Spiking Convolutional Neural https://arxiv.org/pdf/2104.12579.pdf