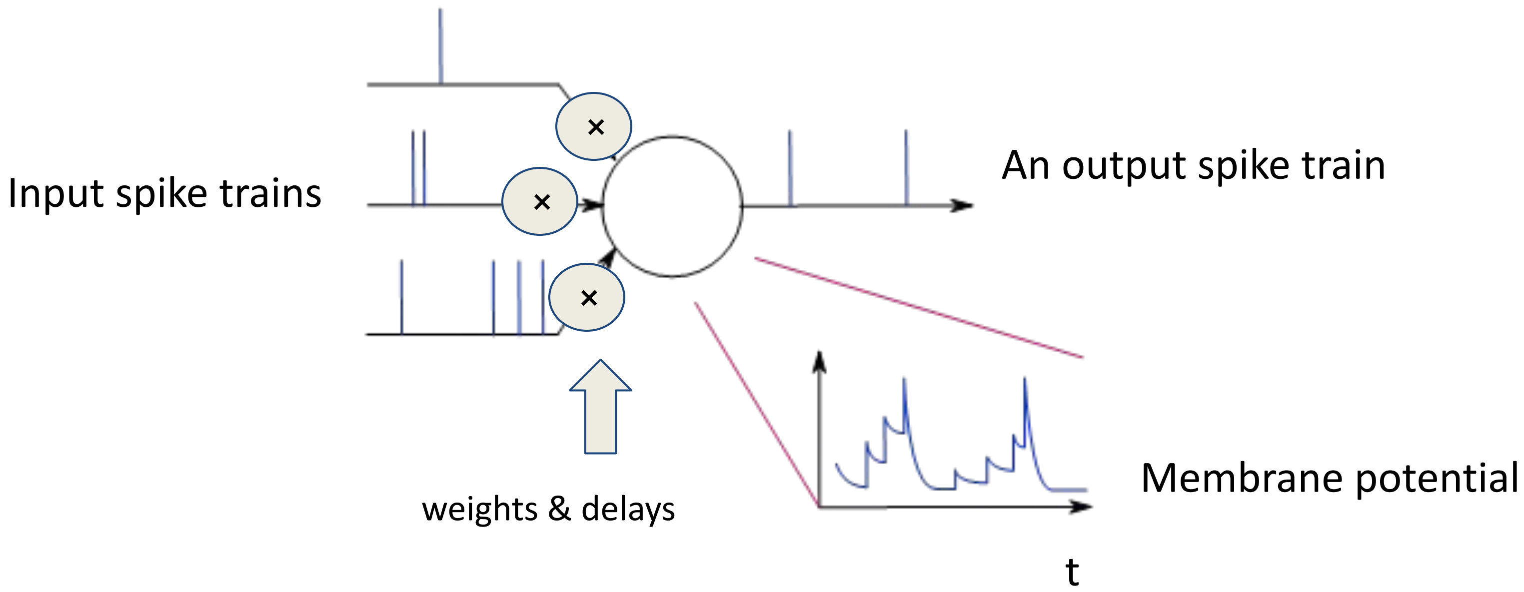
Mini-Batch Training along Convolution Windows for Representation Learning Based on Spike Time Dependent Plasticity Rule

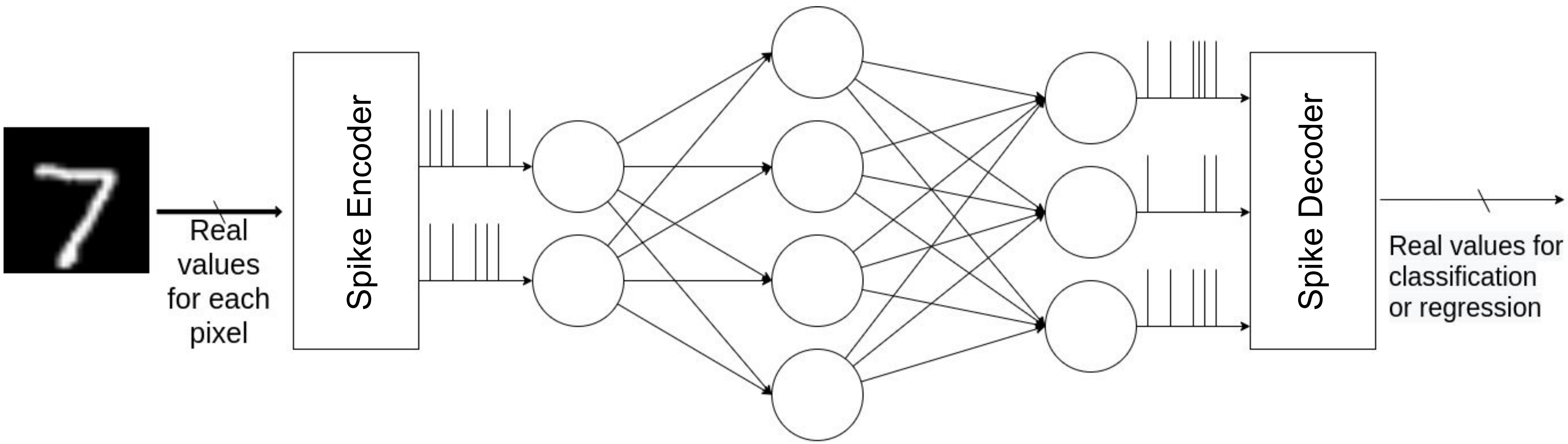
Yohei Shimmyo, Yuichi Okuyama
Graduate School of Computer Science and Engineering, University of Aizu, Japan

Spiking Neural Network (SNN)



- An accurate neuron model
  - Voltage impulses (called ‘spikes’) represents analog values
  - An internal parameter (membrane potential) excited by input spikes
- In contrast to ANN, SNN uses temporal processing

How SNN Applied



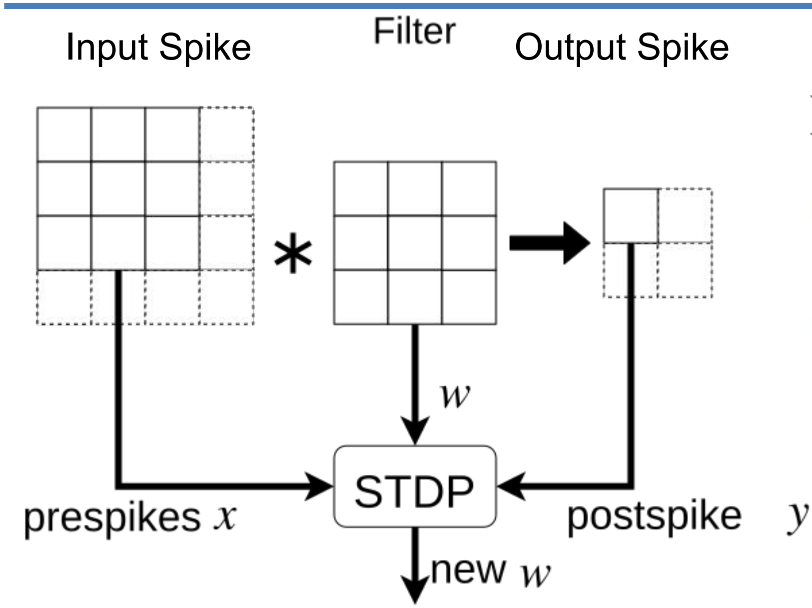
- Neurons work independently
- The neuron input/output are spikes
  - The function between neuron input and output is indifferentiable
  - Conventional backpropagation is difficult

SNN Features Compared to ANN

- Asynchrony neurons
- Effective inference when in an external circuit (Painkras *et al*., 2013)
  - Power consumption
  - Circuit area
- Complex dynamics
- Difficulty of training a network due to differentiability
  - ANN-SNN conversion
  - Backpropagation approximation
  - **Spike time dependent plasticity (STDP)**

Painkras, Eustace, et al. "SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation." IEEE Journal of Solid-State Circuits 48.8 (2013): 1943-1953.

Convolutional SNN and STDP



If postspike happened,

$$w \leftarrow w + \Delta w$$

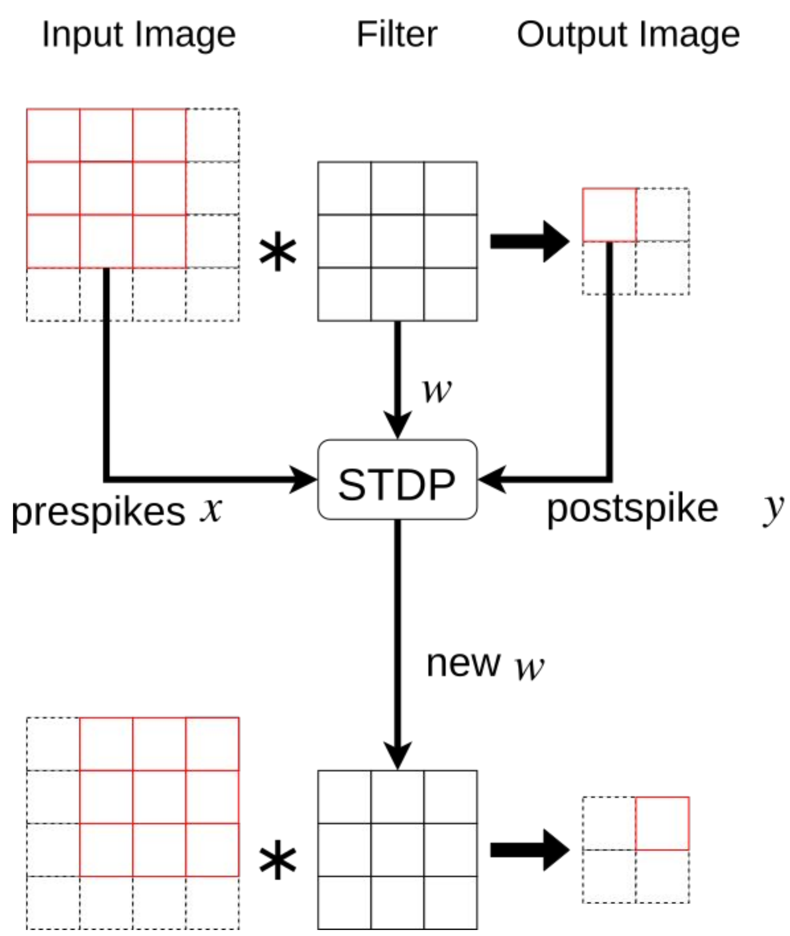
$$\Delta w = \begin{cases} a \cdot (1 - w) & \text{if corresponding prespike happened} \\ a \cdot (-w) & \text{otherwise} \end{cases}$$

w : a pixel of a filter, a : learning rate

- STDP rule trains a sparse code for natural images by spike timings in unsupervised manner (Zylberberg *et al.*, 2011)
- Combination with backpropagation approximation achieved 98.60% MNIST classification accuracy (Tavanaei *et al.*, 2018a)

Zylberberg, Joel, Jason Timothy Murphy, and Michael Robert DeWeese. "A sparse coding model with synaptically local plasticity and spiking neurons can account for the diverse shapes of V1 simple cell receptive fields." PLoS computational biology 7.10 (2011): e1002250.
Tavanaei, Amirhossein, Zachary Kirby, and Anthony S. Maida. "Training spiking convnets by stdp and gradient descent." 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, 2018a.

Computation Dependency of STDP



Conventional STDP implementation (Zylberberg et al., 2011, Tavanaei 2018b) assumes computation dependency.

Computation dependency prevents effective parallel computation by such as GPUs & biologically implausible

Conv. STDP was bottleneck in the work by (Tavanaei *et al.*, 2018a) rather than dense layers' BPA

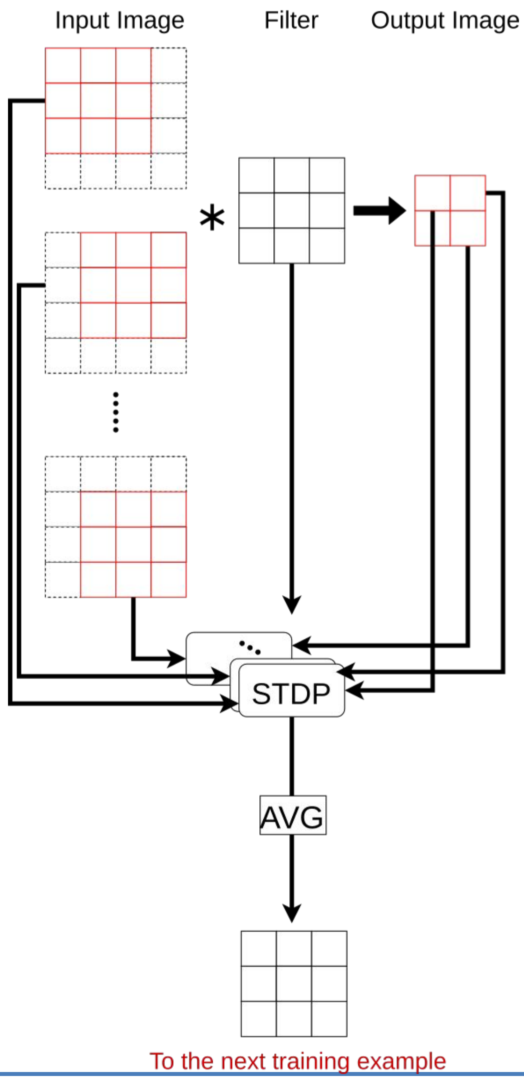
Tavanaei, A., Masquelier, T., & Maida, A. (2018b). Representation learning using event-based STDP. *Neural Networks*, *105*, 294-303.

Goal

- We developed a mini-batch processing for STDP to train effectively on GPUs.

- Experiments showed larger batch sizes reduced execution time and accuracy degradation, due to inequivalent calculation order, was little

Reduction of Training Time by Mini-Batch Processing



Calculate weight updates for several convolution windows at once (mini-batch along convolution windows)

Average of their weight is applied to the next conv. filter

(Calculation result differs)

Implementation

Experiment measured execution time.

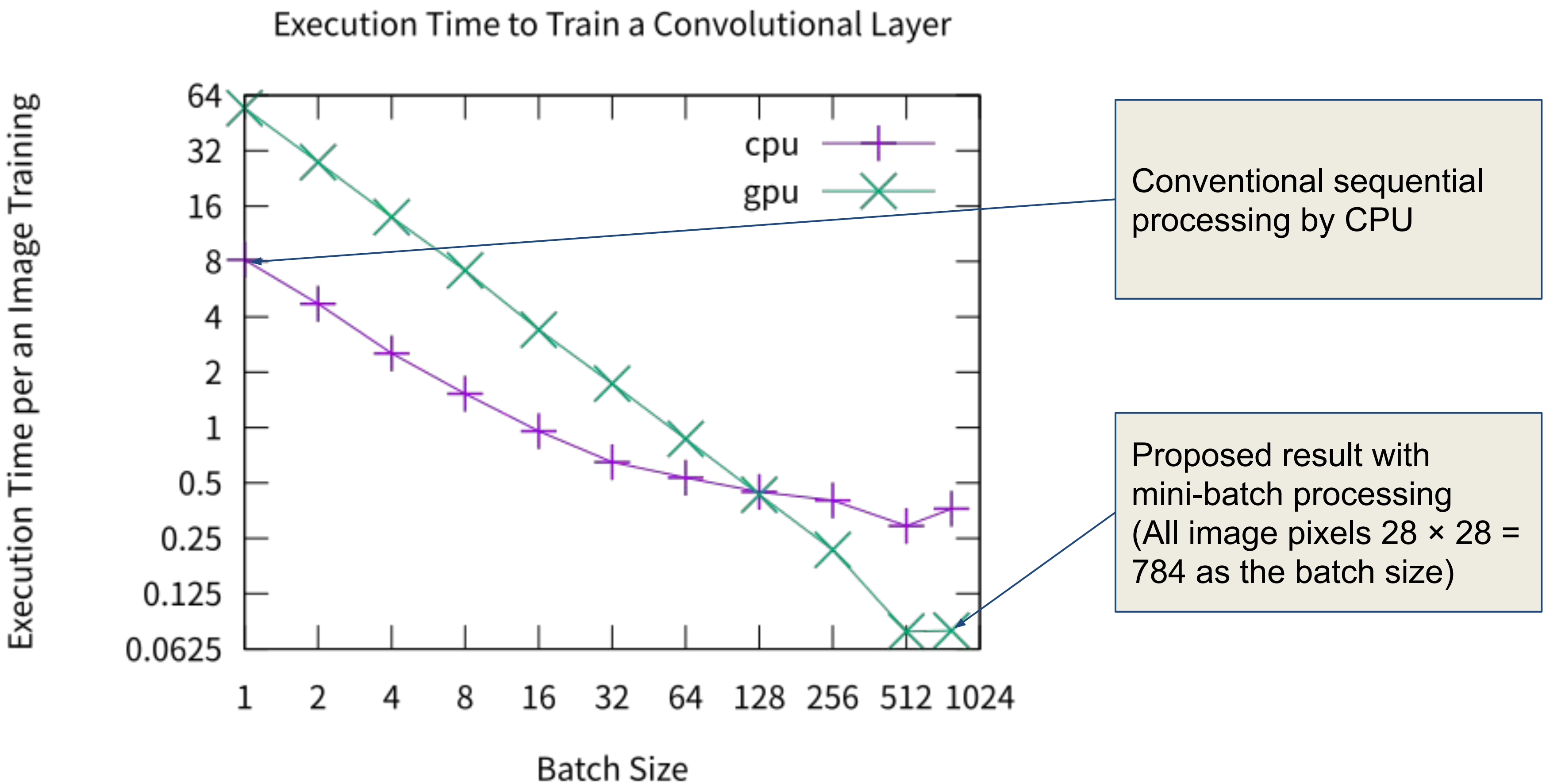
Implementation:

- Python 3.8.8
- Numpy 1.20.2
- Cupy 8.6.0
- CUDA 10.1

On a computer with

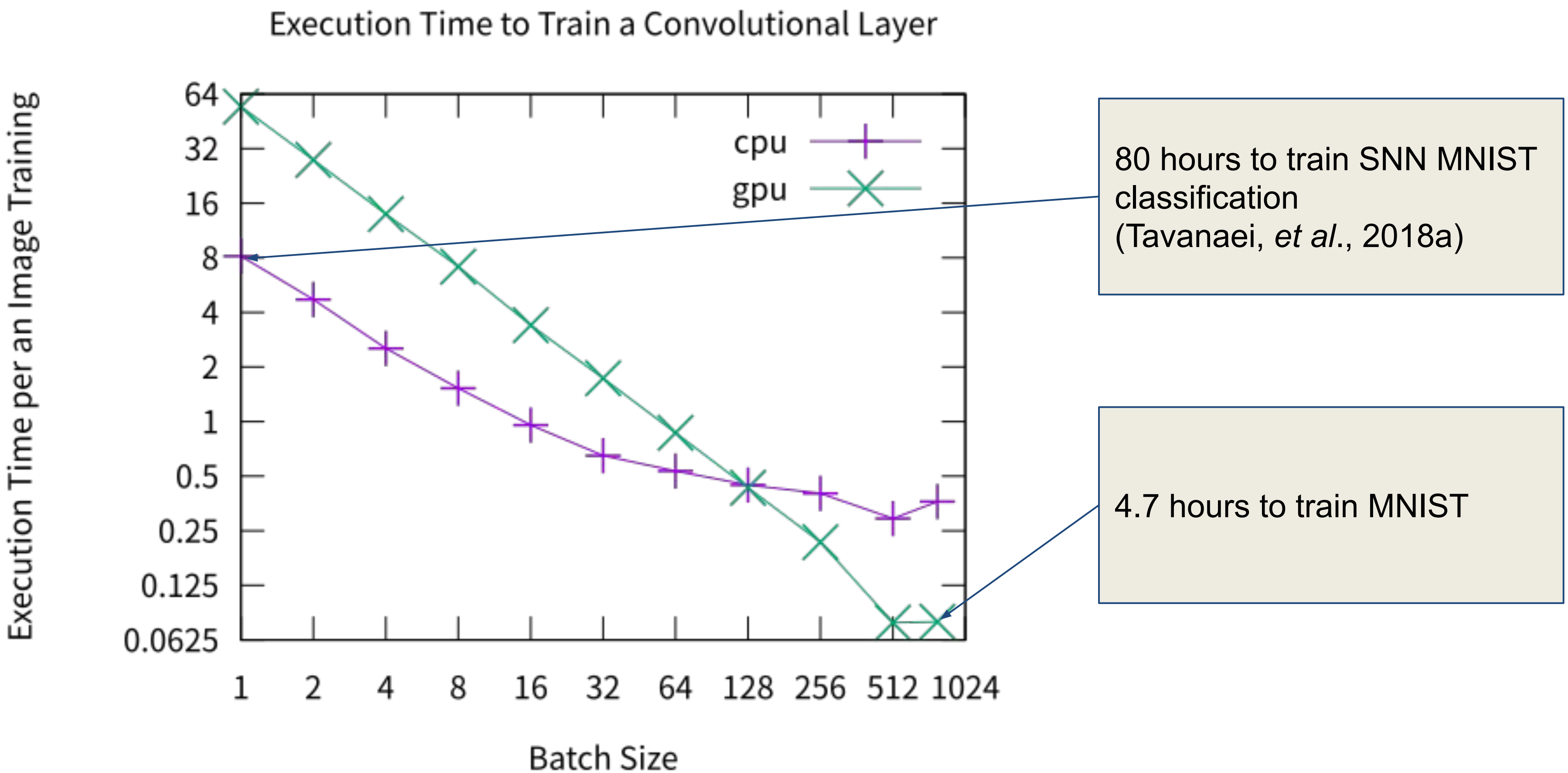
- Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz
- GeForce GTX 1060 6GB VRAM

Execution Speed to Train MNIST



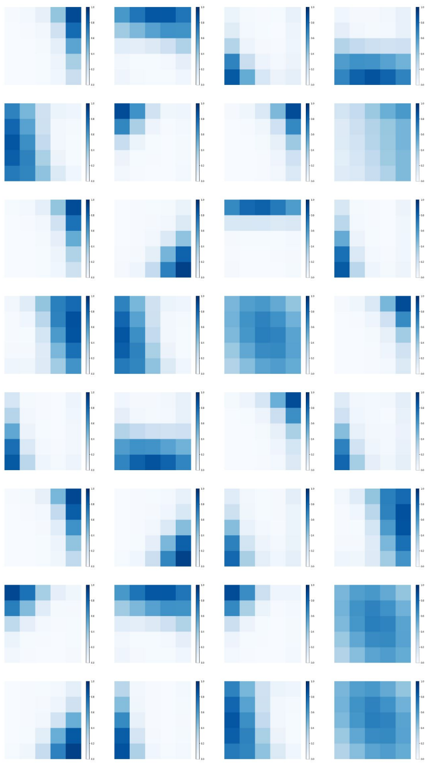
- Larger batch sizes contribute to speed especially with GPU (×120 faster)
- GPU reaches a peak around 512 batch size (no need to mini-batch over images)

Execution Speed to Train MNSIT

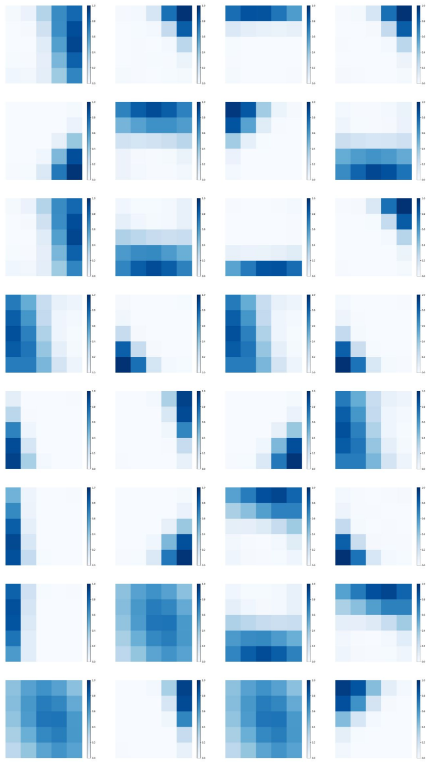


- Larger batch sizes contribute to speed especially with GPU (×200 than conventional 1 batch processing)
- GPU reaches a peak around 512 batch size (no need to mini-batch over images)

Qualitative Evaluation of Convolution Filters



Batch size
1

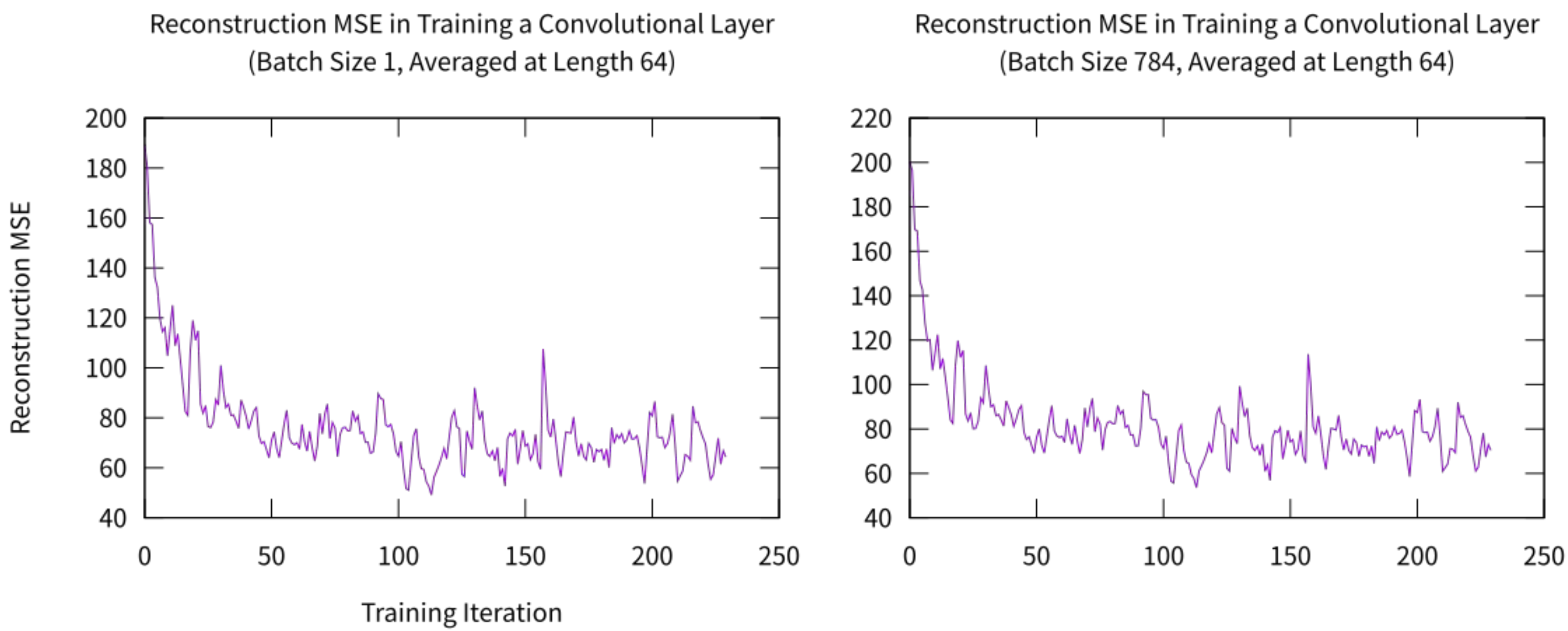


Batch size
784

- Convolution filters after training by MNIST images for batch size 1 and 784.
- Both consist of similar components: horizontal line, vertical line, chunk on the edge and flat filters.
- Similar filter components suggests little effects due to mini-batch training

Reconstruction Degradation for MNIST

- Reconstruction MSE of MNIST during training



- Similar transition suggests no significant effects
- Less than 10% Reconstruction MSE degradation around the final training epochs

Conclusion

- STDP trains indifferentiable spiking neural network for sparse coding in unsupervised manner.
- We proposed mini-batch processing of convolutional STDP over convolution windows
- Evaluation by MNIST showed mini-batch processing enabled effective GPU computation with slight degradation of reconstruction MSE.
  - Processing several convolution windows at once