Electrical and Computer Engineering/Smart Circuits and Systems Lab (SCSL)

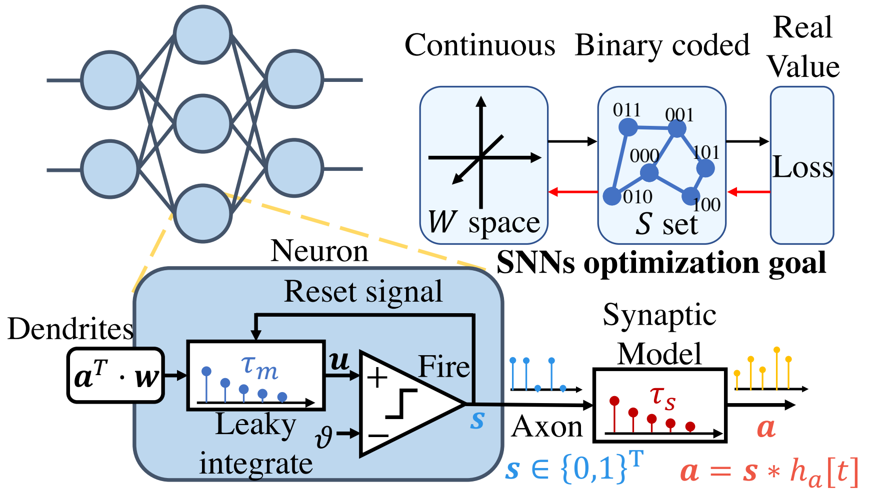
Backpropagated Neighborhood Aggregation for Accurate Training of Spiking Neural Networks

Yukun Yang, Wenrui Zhang, Peng Li

University of California, Santa Barbara

Take home info:
A New Algorithm to Train SNN with Higher Accuracy!

Forward



(a) Forward path of Leaky Integrate-and-Fire SNNs and the generic optimization framework

$$u_i^{(l)}[t+1] = \left(1 - \frac{1}{\tau_m}\right) u_i^{(l)}[t]\left(1 - s_i^{(l)}[t]\right) + \sum_{j=1}^{N^{(l-1)}} w_{ij}^{(l)} a_j^{(l-1)}[t+1]$$

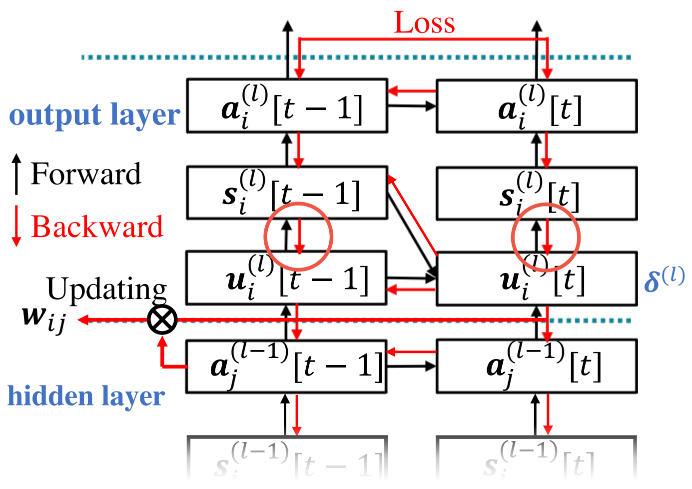
$$a_i^{(l)} = s_i^{(l)} * \sigma, \quad \sigma[t] = \frac{1}{\tau_s}\left(1 - \frac{1}{\tau_s}\right)^t$$

$$a_i^{(l)}[t+1] = \left(1 - \frac{1}{\tau_s}\right) a_i^{(l)}[t] + \left(\frac{1}{\tau_s}\right) s_i^{(l)}[t+1]$$

$$s_i^{(l)}[t] = H\left(u_i^{(l)}[t] - \vartheta\right)$$

$$L = \sum_{t=0}^{N_t} E[t] = \sum_{t=0}^{N_t} \frac{1}{2}\left((\sigma * d)[t] - (\sigma * s)[t]\right)^2 = \sum_{t=0}^{N_t} \frac{1}{2}\left((\sigma * d)[t] - a[t]\right)^2$$

BPTT's Difficulty



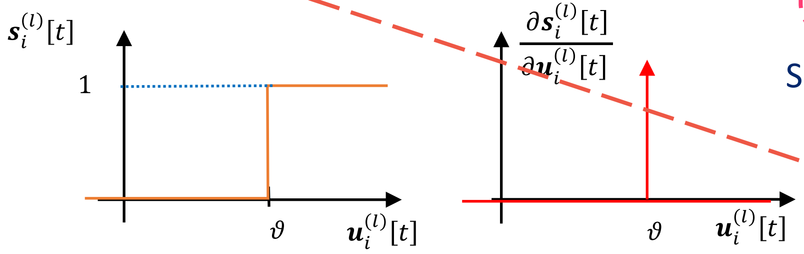
(b) Typical backpropagation through time dataflow

$$g_i^{(l)} = \begin{cases} a_i^{(l)} - (\sigma * d_i), & \text{output layer} \\ \sum_{p=1}^{N^{(l+1)}} w_{ji}^{(l+1)} \delta_p^{(l+1)}, & \text{hidden layer} \end{cases}$$

$$e_i^{(l)}[t] = \begin{cases} g_i^{(l)}[t], & t = N_t \\ g_i^{(l)}[t] + \frac{\partial a_i^{(l)}[t+1]}{\partial a_i^{(l)}[t]} e_i^{(l)}[t+1], & t < N_t \end{cases}$$

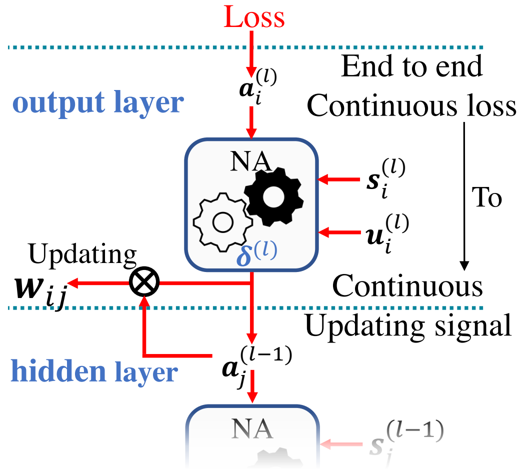
$$\delta_i^{(l)}[t] = \begin{cases} e_i^{(l)}[t] \frac{\partial a_i^{(l)}[t]}{\partial u_i^{(l)}[t]}, & t = N_t \\ e_i^{(l)}[t] \frac{\partial a_i^{(l)}[t]}{\partial u_i^{(l)}[t]} + \delta_i^{(l)}[t+1] \frac{\partial u_i^{(l)}[t+1]}{\partial u_i^{(l)}[t]}, & t < N_t \end{cases}$$

Non-differentiable



$$\frac{\partial L}{\partial w_{ij}^{(l)}} = \sum_{t=0}^{N_t} \frac{\partial u_i^{(l)}[t]}{\partial w_{ij}^{(l)}} \delta_i^{(l)}[t] = \sum_{t=0}^{N_t} a_j^{(l-1)}[t] \delta_i^{(l)}[t]$$

Our NA Algorithm



(c) Neighborhood aggregation (NA) algorithm

1. $$f_{d(u,u_p)} L = \frac{L(u_p) - L(u)}{d_{MP}(u, u_p)} \approx \frac{e \cdot (a_p - a)}{d_{MP}(u, u_p)}$$

$$L(u_p) - L(u) = L(a_p) - L(a) \approx \nabla_a L (a_p - a) = e \cdot (a_p - a)$$

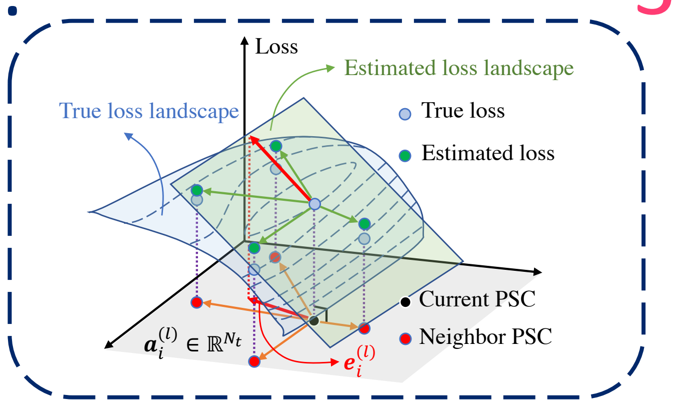
2. $$d_{MP}(u, u') = \| u' \boxminus u \|_2$$

3. $$\begin{bmatrix} o_1^T \\ o_2^T \\ \vdots \\ o_M^T \end{bmatrix} \cdot \tilde{\nabla}_u L = \begin{bmatrix} f_{d(u,u_1)} L \\ f_{d(u,u_2)} L \\ \vdots \\ f_{d(u,u_M)} L \end{bmatrix}, \quad \tilde{\nabla}_u L = \begin{bmatrix} o_1^T \\ o_2^T \\ \vdots \\ o_M^T \end{bmatrix}^+ \begin{bmatrix} f_{d(u,u_1)} L \\ f_{d(u,u_2)} L \\ \vdots \\ f_{d(u,u_M)} L \end{bmatrix}$$

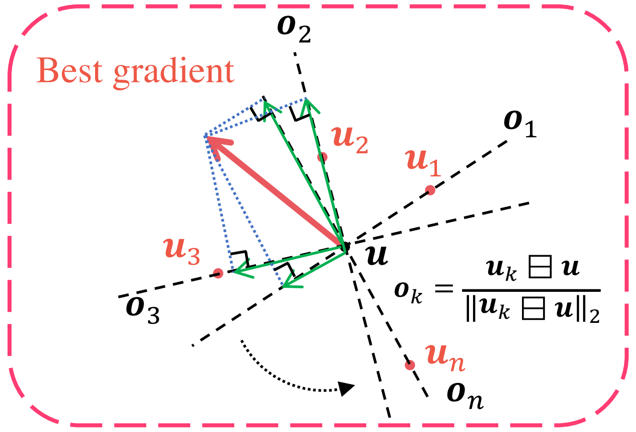
Simple neighborhood selection

4. $$\delta \approx \tilde{\nabla}_u L = \begin{bmatrix} f_{d(u,u_1)} L \\ f_{d(u,u_2)} L \\ \vdots \\ f_{d(u,u_M)} L \end{bmatrix}$$

1.



3.



2.

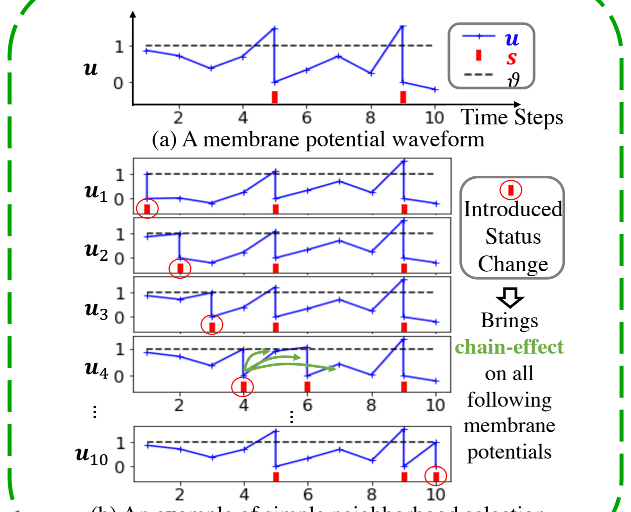
Algorithm 1 Membrane potential addition: $u' = u \boxplus \epsilon$

Inputs: u, ϵ; **Output:** u'; ϑ: firing threshold;
Initialization: $u = 0$;
$c = f^{-1}(u)$;
for t in $range(1, N_t)$**:**
  $u'[t] = u\left(1 - \frac{1}{\tau_m}\right) + \epsilon[t] + c[t]$;
  $u = u'[t]\left(1 - H(u'[t] - \vartheta)\right)$;
return u'

Algorithm 2 Membrane potential subtraction: $\epsilon = u' \boxminus u$

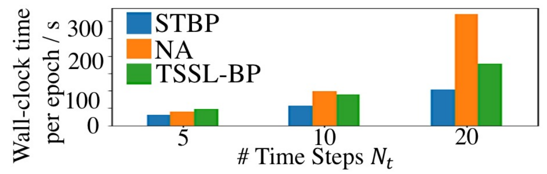
Inputs: u, u'; Output: ϵ; ϑ: firing threshold;
initialization: $u = 0$;
$c = f^{-1}(u)$;
for t in $range(1, N_t)$**:**
  $u[t] = u(1 - \frac{1}{\tau_m}) + c[t]$;
  $\epsilon[t] = u'[t] - u[t]$;
  $u = u'[t]\left(1 - H(u'[t] - \vartheta)\right)$;
return ϵ

4.



(b) An example of simple neighborhood selection

Experimental Results



Wall-clock time comparison (MNIST dataset, single TI-TAN XP GPU, Batch Size=64)

MNIST

Method	#Steps	BestAcc
HM2BP (Jin et al., 2018)	400	99.49%
ST-RSBP (Zhang & Li, 2019)	400	99.62%
SLAYER (Shrestha & Orchard, 2018)	300	99.41%
STBP (Wu et al., 2018)	30	99.42%
TSSL-BP (Zhang & Li, 2020)	5	99.53%
This work	**5**	**99.69%**

Spiking CNN structure: 15C5-P2-40C5-P2-300

CIFAR10

Methods	Structure	#Time steps	Best accuracy
STBP	AlexNet	12	85.24%
STBP	CifarNet	12	90.53%
TSSL-BP	AlexNet	5	89.22%
This work	AlexNet	5	**91.76%**

AlexNet structure: 96C3-256C3-P2-384C3-P2-384C3-256C3-1024-1024
CifarNet structure: 128C3-256C3-P2-512C3-P2-1024C3- 512C3-1024-512

Contact

https://github.com/superrrpotato/Neighbors

yukunyang@ucsb.edu
wenruizhang@ucsb.edu
lip@ucsb.edu

Wu, Y., Deng, L., Li, G., Zhu, J., Xie, Y., and Shi, L. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1311–1318, 2019.

Zhang, W. and Li, P. Temporal spike sequence learning via backpropagation for deep spiking neural networks. *Advances in Neural Information Processing Systems*, 33, 2020.

Jin, Y., Zhang, W., and Li, P. Hybrid macro/micro level backpropagation for training deep spiking neural networks. *Advances in neural information processing systems*, 31: 7005–7015, 2018.

Zhang, W. and Li, P. Spike-train level backpropagation for training deep recurrent spiking neural networks. *Advances in neural information processing systems*, 2019.

Shrestha, S. B. and Orchard, G. Slayer: Spike layer error reassignment in time. *Advances in Neural Information Processing Systems*, 2018.