NEURAL INFORMATION PROCESSING SYSTEMS

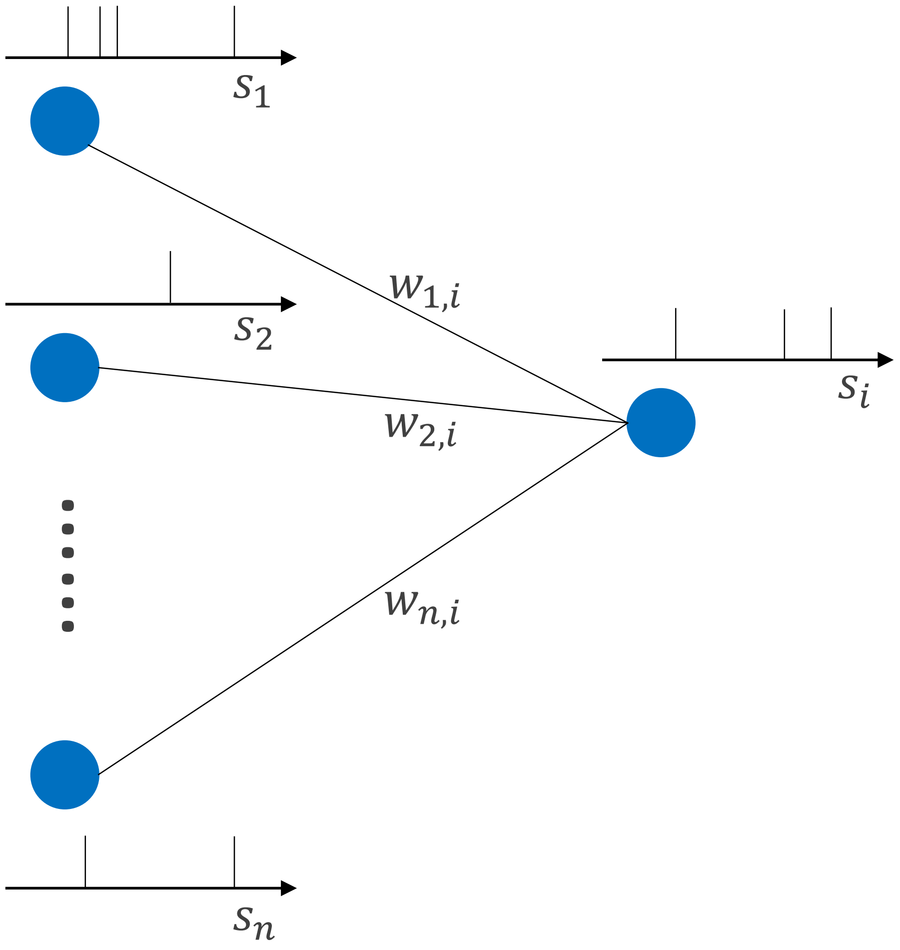
Training Feedback Spiking Neural Networks by Implicit Differentiation on the Equilibrium State

Mingqing Xiao[1], Qingyan Meng[2,3], Zongpeng Zhang[1], Yisen Wang[1], Zhouchen Lin[1,4]

[1]Peking University [2]The Chinese University of Hong Kong, Shenzhen

[3]Shenzhen Research Institute of Big Data [4]Pazhou Lab, Guangzhou

SNN Background



Spike -> Current -> Membrane Potential

If membrane potential reaches threshold, generate spike and reset

Event-driven energy-efficient computation on neuromorphic hardware

SNN Background

Membrane potential:

$$\textbf{IF:} \quad \frac{du}{dt} = R \cdot I(t), \qquad u < V_{th}$$

$$\textbf{LIF:} \quad \tau_m \frac{du}{dt} = -(u - u_{rest}) + R \cdot I(t), \qquad u < V_{th}$$

Current: $I_i(t) = \sum_j w_{ij} s_j(t) + b$

Discrete formulation:

$$\begin{cases} u_i\,[t+0.5] = \lambda u_i[t] + \sum_j w_{ij} s_j[t] + b, \\ s_i[t+1] = H(u_i\,[t+0.5] - V_{th}), \\ u_i[t+1] = u_i\,[t+0.5] - V_{th} s_i[t+1], \end{cases}$$

$$H\,(x) = \begin{cases} 1, x > 0 \\ 0, x < 0 \end{cases}.$$

SNN Background

Membrane potential:

$$\textbf{IF:} \quad \frac{du}{dt} = R \cdot I(t), \qquad u < V_{th}$$

$$\textbf{LIF:} \quad \tau_m \frac{du}{dt} = -(u - u_{rest}) + R \cdot I(t), \qquad u < V_{th}$$

Current: $I_i(t) = \sum_j w_{ij} s_j(t) + b$

Complex neuron model

Non-differentiable

Hard to train

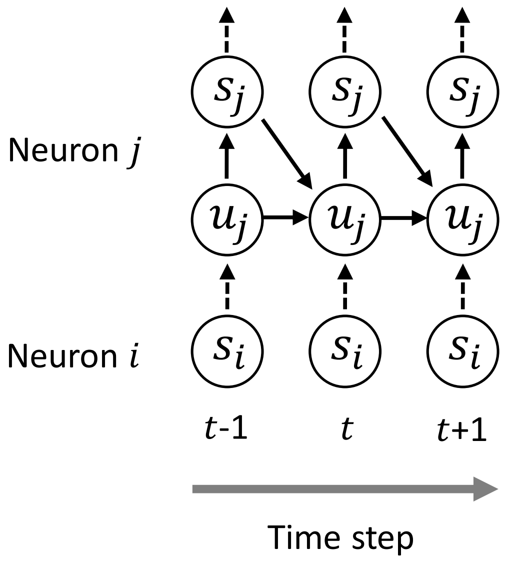
Discrete formulation:

$$\begin{cases} u_i\left[t+0.5\right] = \lambda u_i[t] + \sum_j w_{ij} s_j[t] + b, \\ s_i[t+1] = H(u_i\left[t+0.5\right] - V_{th}), \\ u_i[t+1] = u_i\left[t+0.5\right] - V_{th} s_i[t+1], \end{cases}$$

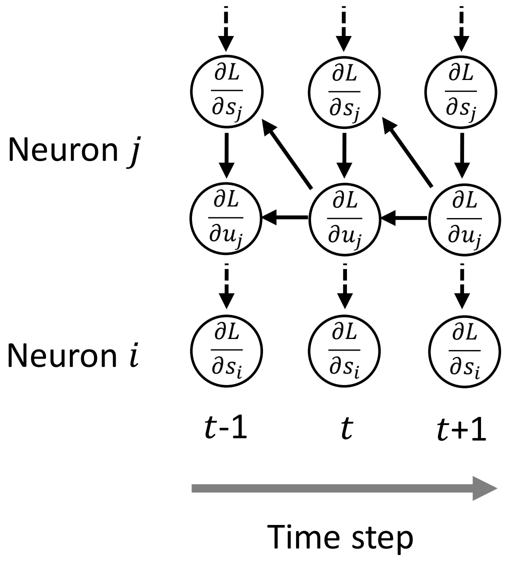
$$H(x) = \begin{cases} 1, x > 0 \\ 0, x < 0 \end{cases}.$$

Previous Training Framework

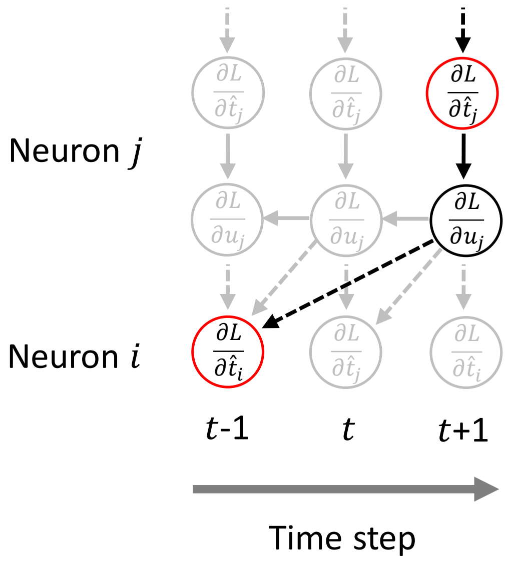
BPTT-like; Activation-based / Timing-based



Forward computational graph



Activation-based



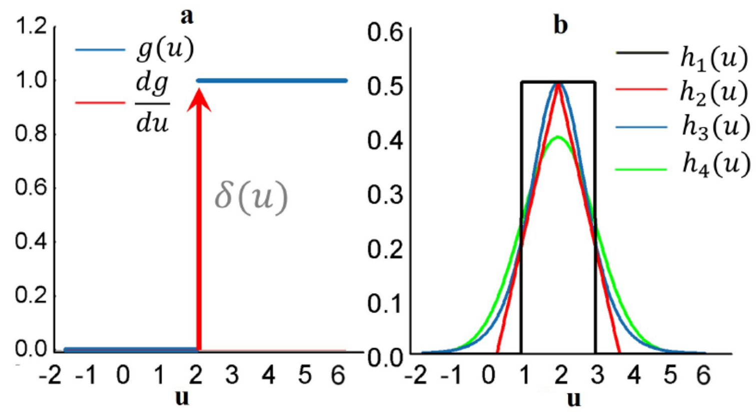
Timing-based

Backpropagation along computational graphs

Kim J., Kim K., and Kim J. Unifying activation-and timing-based learning rules for spiking neural networks. NeurIPS, 2020.

Previous Training Framework

BPTT-like; Activation-based / Timing-based



Activation-based with surrogate derivative

$$\frac{\partial u_j[t_a]}{\partial u_i[\hat{t}_i]} = \frac{\partial u_j[t_a]}{\partial \hat{t}_i} \frac{\partial \hat{t}_i}{\partial u_i[\hat{t}_i]}$$

Timing-based with assumption of the existence of spikes

Wu Y., Deng L., Li G., et al. Spatio-temporal backpropagation for training high-performance spiking neural networks. Frontiers in Neuroscience, 2018.
Zhang W. and Li P. Temporal spike sequence learning via backpropagation for deep spiking neural networks. NeurIPS, 2020.

Previous Training Framework

BPTT-like; Activation-based / Timing-based

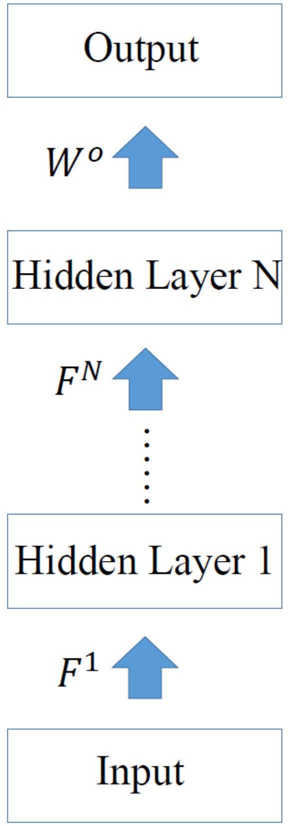
Drawback for BPTT-like methods:

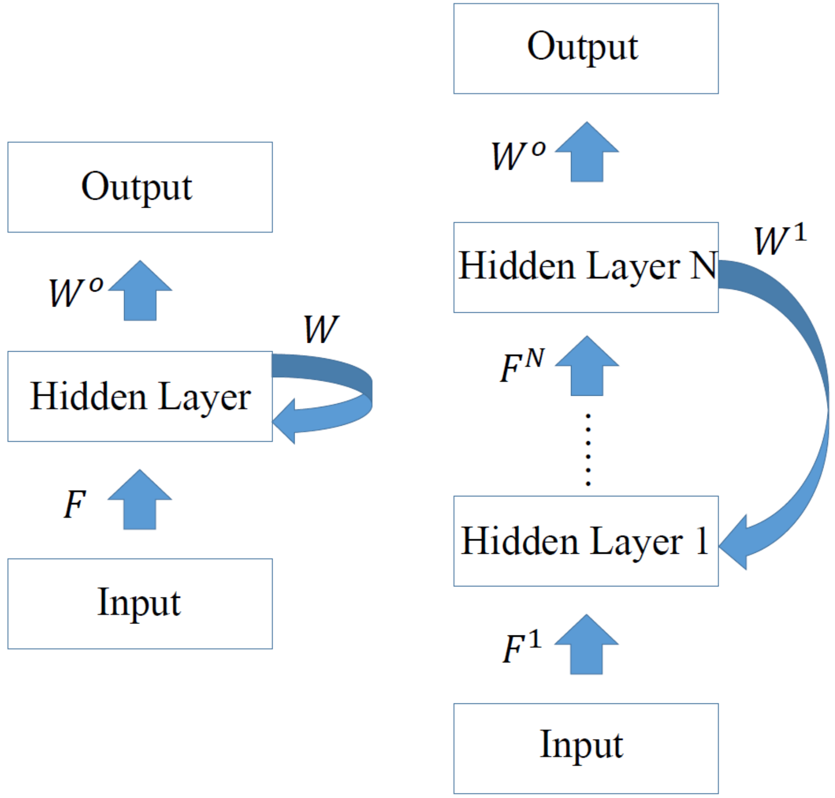
- Large memory costs;
- Biologically implausible;
- Accumulated approximation error (activation-based);
- “Dead neuron” problem – no spike, no learning (timing-based).

New training methods other than backpropagation along the computational graph?

Network Structure

Feedforward vs. Feedback





(a) Single-layer (b) Multi-layer

Feedback (recurrent) circuits are critical to *human's vision system*

ANN with recurrence: *Shallow , Higher functional fidelity* of human brains

SNN naturally compute for multiple time steps – *avoid uneconomical costs*

We consider feedback spiking neural networks (FSNN)

Kar K., Kubilius J., Schmidt K., et al. Evidence that recurrent circuits are critical to the ventral stream's execution of core object recognition behavior. Nature Neuroscience, 2019.
Kubilius J., Schrimpf M., Kar K., et al. Brain-like object recognition with high-performing shallow recurrent ANNs. NeurIPS, 2019.

Equilibrium of Neural Networks

Early energy-based models

The dynamics of feedback neural networks minimize an *energy function*

Converge to a *minimum of the energy*

Training: recurrent backpropagation, equilibrium propagation

Deep equilibrium models

Express the entire deep network as solving an *implicit fixed-point equilibrium point*

Forward: root-finding methods

Backward: implicit differentiation on the equation

Almeida LB. A learning rule for asynchronous perceptrons with feedback in a combinatorial environment. ICNN, 1987.
Scellier B., and Bengio Y. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. Frontiers in Computational Neuroscience, 2017
Bai S., Kolter J. Z., and Koltun V. Deep equilibrium models. NeurIPS, 2019.
Bai S., Koltun V., and Kolter J. Z. Multiscale deep equilibrium models. NeurIPS, 2020.

Method Overview

1. Derive the **equilibrium states** with a fixed-point equation for FSNN computation

2. View the forward computation of FSNN as solving an implicit equation for the equilibrium state; apply **implicit differentiation** to calculate gradients

- *Backward computation is **decoupled from** the forward computational graph*
- *Avoid common SNN training problems including non-differentiability and large memory costs*
- *More biologically plausible than BPTT-like methods*

Equilibrium States of FSNN

Continuous View – IF model

First consider a group of spiking neurons with feedback connections.

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{W}\mathbf{s}(t - \Delta t_d) + \mathbf{F}\mathbf{x}(t) + \mathbf{b} - V_{th}\mathbf{s}(t)$$

Define *average firing rates*: $\mathbf{a}(t) = \frac{1}{t}\int_0^t \mathbf{s}(\tau)\mathrm{d}\tau$ *average inputs*: $\overline{\mathbf{x}}(t) = \frac{1}{t}\int_0^t \mathbf{x}(\tau)\mathrm{d}\tau$

Through integration, we have:

$$\mathbf{a}(t) = \frac{1}{V_{th}}\left(\frac{t - \Delta t_d}{t}\mathbf{W}\mathbf{a}(t - \Delta t_d) + \mathbf{F}\overline{\mathbf{x}}(t) + \mathbf{b} - \underline{\frac{\mathbf{u}(t)}{t}}\right)$$

Equilibrium States of FSNN

Continuous View – IF model

Neurons will not spike when accumulated $\boldsymbol{u}_i(t)$ is negative

↓

$\boldsymbol{u}_i(t)$ can be divided as $\boldsymbol{u}_i(t) = \boldsymbol{u}_i^-(t) + \boldsymbol{u}_i^+(t)$

$\frac{\mathbf{1}}{\boldsymbol{t}}\boldsymbol{u}_i^-(t) = \min\left(0, \left(\frac{t-\Delta t_d}{t}\boldsymbol{W}\boldsymbol{a}(t-\Delta t_d) + \boldsymbol{F}\overline{\boldsymbol{x}}(t) + \boldsymbol{b}\right)_i\right)$ is the remaining negative term

$\boldsymbol{u}_i^+(t)$ is a bounded remaining positive term

$$\mathbf{a}(t) = \mathrm{ReLU}\left(\frac{1}{V_{th}}\left(\frac{t-\Delta t_d}{t}\mathbf{W}\mathbf{a}(t-\Delta t_d) + \mathbf{F}\overline{\mathbf{x}}(t) + \mathbf{b} - \frac{\mathbf{u}^+(t)}{t}\right)\right)$$

Theorem 1. *If the average inputs converge to an equilibrium point* $\overline{\mathbf{x}}(t) \rightarrow \mathbf{x}^*$*, and there exists constant* c *and* $\gamma < 1$ *such that* $|\mathbf{u}_i^+(t)| \leq c, \forall i, t$ *and* $\|\mathbf{W}\|_2 \leq \gamma V_{th}$*, then the average firing rates of FSNN with continuous IF model in Eq.(6) will converge to an equilibrium point* $\mathbf{a}(t) \rightarrow \mathbf{a}^*$*, which satisfies the fixed-point equation* $\mathbf{a}^* = ReLU\left(\frac{1}{V_{th}}(\mathbf{W}\mathbf{a}^* + \mathbf{F}\mathbf{x}^* + \mathbf{b})\right)$. Equilibrium State

Equilibrium States of FSNN

Discrete View – IF model

$$\mathbf{u}[t+1] = \mathbf{u}[t] + \mathbf{W}\mathbf{s}[t] + \mathbf{F}\mathbf{x}[t] + \mathbf{b} - V_{th}\mathbf{s}[t+1]$$

Define *average firing rates*: $\mathbf{a}[t] = \frac{1}{t}\sum_{\tau=1}^{t}\mathbf{s}[\tau]$ *average inputs*: $\overline{\mathbf{x}}(t) = \frac{1}{t+1}\sum_{\tau=0}^{t}\mathbf{x}[\tau]$

By summation, we have:

$$\mathbf{a}[t+1] = \frac{1}{V_{th}}\left(\frac{t}{t+1}\mathbf{W}\mathbf{a}[t] + \mathbf{F}\overline{\mathbf{x}}[t] + \mathbf{b} - \frac{\mathbf{u}[t+1]}{t+1}\right)$$

Equilibrium States of FSNN

Discrete View – IF model

Differently, there could be at most t spikes during t discrete time steps

$$\boldsymbol{u}_i[t] = \boldsymbol{u}_i^-[t] + \boldsymbol{u}_i^+[t]$$

$\boldsymbol{u}_i^-[t]$ is the remaining negative term or the exceeded positive term

$\boldsymbol{u}_i^+[t]$ is a bounded remaining positive term

$$\mathbf{a}[t+1] = \sigma\left(\frac{1}{V_{th}}\left(\frac{t}{t+1}\mathbf{W}\mathbf{a}[t] + \mathbf{F}\overline{\mathbf{x}}[t] + \mathbf{b} - \frac{\mathbf{u}^+[t+1]}{t+1}\right)\right), \text{where } \sigma(x) = \begin{cases} 1, & x > 1 \\ x, & 0 \le x \le 1 \\ 0, & x < 0 \end{cases}$$

Theorem 2. *If the average inputs converge to an equilibrium point* $\overline{\mathbf{x}}[t] \to \mathbf{x}^*$, *and there exists constant* c *and* $\gamma < 1$ *such that* $|\mathbf{u}_i^+[t]| \le c, \forall i, t$ *and* $\|\mathbf{W}\|_2 \le \gamma V_{th}$, *then the average firing rates of FSNN with discrete IF model in Eq.(9) will converge to an equilibrium point* $\mathbf{a}[t] \to \mathbf{a}^*$, *which satisfies the fixed-point equation* $\mathbf{a}^* = \sigma\left(\frac{1}{V_{th}}(\mathbf{W}\mathbf{a}^* + \mathbf{F}\mathbf{x}^* + \mathbf{b})\right)$.

Equilibrium State

Equilibrium States of FSNN

Continuous View – LIF model

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = -\frac{1}{\tau_m}\mathbf{u} + \mathbf{W}\mathbf{s}(t-\Delta t_d) + \mathbf{F}\mathbf{x}(t) + \mathbf{b} - V_{th}\mathbf{s}(t)$$

$$\mathbf{u}(t) = \mathbf{W}\int_0^{t-\Delta t_d} \kappa(t-\Delta t_d-\tau)\mathbf{s}(\tau)\mathrm{d}\tau + \mathbf{F}\int_0^t \kappa(t-\tau)\mathbf{x}(\tau)\mathrm{d}\tau + t\mathbf{b} - V_{th}\int_0^t \kappa(t-\tau)\mathbf{s}(\tau)\mathrm{d}\tau$$

$\kappa(\tau) = \exp(-\frac{\tau}{\tau_m})$ is the response kernel of the LIF model

Define *weighted average firing rates*: $\hat{\mathbf{a}}(t) = \frac{\int_0^t \kappa(t-\tau)\mathbf{s}(\tau)\mathrm{d}\tau}{\int_0^t \kappa(t-\tau)\mathrm{d}\tau}$

weighted average inputs: $\hat{\mathbf{x}}(t) = \frac{\int_0^t \kappa(t-\tau)\mathbf{x}(\tau)\mathrm{d}\tau}{\int_0^t \kappa(t-\tau)\mathrm{d}\tau}$

$$\hat{\mathbf{a}}(t) = \frac{1}{V_{th}}\left(\frac{\int_0^{t-\Delta t_d}\kappa(\tau)\mathrm{d}\tau}{\int_0^t \kappa(\tau)\mathrm{d}\tau}\mathbf{W}\hat{\mathbf{a}}(t-\Delta t_d) + \mathbf{F}\hat{\mathbf{x}}(t) + \mathbf{b} - \underline{\frac{\mathbf{u}(t)}{\int_0^t \kappa(\tau)\mathrm{d}\tau}}\right)$$

Equilibrium States of FSNN

Continuous View – LIF model

$\boldsymbol{u}_i(t)$ can be similarly divided as $\boldsymbol{u}_i(t) = \boldsymbol{u}_i^-(t) + \boldsymbol{u}_i^+(t)$

Random error caused by $\frac{\mathbf{u}^+(t)}{\int_0^t \kappa(\tau)\mathrm{d}\tau}$ is not eliminated with time

An approximate solver for the equilibrium with bounded random errors

Proposition 1. *If the weighted average inputs converge to an equilibrium point* $\hat{\mathbf{x}}(t) \rightarrow \mathbf{x}^*$, *and there exists constant* c *and* $\gamma < 1$ *such that* $|\mathbf{u}_i^+(t)| \leq c, \forall i, t$ *and* $\|\mathbf{W}\|_2 \leq \gamma V_{th}$, *then the weighted average firing rates* $\hat{\mathbf{a}}(t)$ *of FSNN with continuous LIF model gradually approximate an equilibrium point* $mathbf{a}^*$ *with bounded random errors, which satisfies* $\mathbf{a}^* = ReLU\left(\frac{1}{V_{th}}(\mathbf{W}\mathbf{a}^* + \mathbf{F}\mathbf{x}^* + \mathbf{b})\right)$.

The same equation as the IF model

Equilibrium States of FSNN

Discrete View – LIF model

$$\mathbf{u}[t+1] = \lambda\mathbf{u}[t] + \mathbf{W}\mathbf{s}[t] + \mathbf{F}\mathbf{x}[t] + b - V_{th}\mathbf{s}[t+1]$$

Define *weighted average firing rates*: $\hat{\mathbf{a}}[t] = \frac{\sum_{\tau=1}^{t} \lambda^{t-\tau}\mathbf{s}[\tau]}{\sum_{\tau=1}^{t} \lambda^{t-\tau}}$

weighted average inputs: $\hat{\mathbf{x}}[t] = \frac{\sum_{\tau=0}^{t} \lambda^{t-\tau}\mathbf{x}[\tau]}{\sum_{\tau=0}^{t} \lambda^{t-\tau}}$

Proposition 2. *If the weighted average inputs converge to an equilibrium point* $\hat{\mathbf{x}}[t] \rightarrow \mathbf{x}^*$*, and there exists constant* c *and* $\gamma < 1$ *such that* $|\mathbf{u}_i^+[t]| \leq c, \forall i, t$ *and* $\|\mathbf{W}\|_2 \leq \gamma V_{th}$*, then the weighted average firing rates* $\hat{\mathbf{a}}[t]$ *of FSNN with discrete LIF model gradually approximate an equilibrium point* $\mathbf{a}^*$ *with bounded random errors, which satisfies* $\mathbf{a}^* = \sigma\left(\frac{1}{V_{th}}\left(\mathbf{W}\mathbf{a}^* + \mathbf{F}\mathbf{x}^* + \mathbf{b}\right)\right)$.

The same equation as the IF model

Implicit Differentiation

Consider the fixed-point equation $\mathbf{a} = f_\theta(\mathbf{a})$ $g_\theta(\mathbf{a}) = f_\theta(\mathbf{a}) - \mathbf{a}$

Consider the objective function on the equilibrium state $\mathcal{L}(\mathbf{a}^*)$

The implicit differentiation satisfies $\left(I - \frac{\partial f_\theta(\mathbf{a}^*)}{\partial \mathbf{a}^*}\right)\frac{\mathrm{d}\mathbf{a}^*}{\mathrm{d}\theta} = \frac{\partial f_\theta(\mathbf{a}^*)}{\partial \theta}$

Gradients can be calculated by $$\frac{\partial \mathcal{L}(\mathbf{a}^*)}{\partial \theta} = -\frac{\partial \mathcal{L}(\mathbf{a}^*)}{\partial \mathbf{a}^*}\left(J_{g_\theta}^{-1}|_{\mathbf{a}^*}\right)\frac{\partial f_\theta(\mathbf{a}^*)}{\partial \theta}$$

The inverse Jacobian can be solved by solving $\left(J_{g_\theta}^T|_{\mathbf{a}^*}\right)\mathbf{x} + \left(\frac{\partial \mathcal{L}(\mathbf{a}^*)}{\partial \mathbf{a}^*}\right)^T = 0$

Bai S., Kolter J. Z., and Koltun V. Deep equilibrium models. NeurIPS, 2019.
Bai S., Koltun V., and Kolter J. Z. Multiscale deep equilibrium models. NeurIPS, 2020.

Loss and Training Pipeline

1. **Simulate SNN by T time steps**. Treat the (weighted) average firing rates $\mathbf{a}[T]$ as approximately following the fixed-point equation of the equilibrium state

2. Configure a readout layer after these spiking neurons with neurons that do not spike or reset, the output is equivalent as $\mathbf{o} = \mathbf{W}^o\mathbf{a}[T]$. Loss $L(a[T]) = \mathcal{L}(o, y)$.

3. **Calculate gradients based on implicit differentiation on the equilibrium state**

3.1 Solve $\left(J_{g_\theta}^T|_{\mathbf{a}^*}\right)\mathbf{x} + \left(\frac{\partial \mathcal{L}(\mathbf{a}^*)}{\partial \mathbf{a}^*}\right)^T = 0$ by root-finding methods

3.2 Calculate gradients for parameters $\frac{\partial \mathcal{L}(\mathbf{a}^*)}{\partial \theta} = -\frac{\partial \mathcal{L}(\mathbf{a}^*)}{\partial \mathbf{a}^*}\left(J_{g_\theta}^{-1}|_{\mathbf{a}^*}\right)\frac{\partial f_\theta(\mathbf{a}^*)}{\partial \theta}$

4. Optimize parameters by SGD or its variants

Biological Plausibility

Consider $\mathbf{a}^* = f_\theta(\mathbf{a}^*, \mathbf{x}^*) = \mathrm{ReLU}\left(\frac{1}{V_{th}}\left(\mathbf{W}\mathbf{a}^* + \mathbf{F}\mathbf{x}^* + \mathbf{b}\right)\right)$

Let $\mathbf{m} = f'_\theta(\mathbf{a}^*, \mathbf{x}^*) = H\left(\frac{1}{V_{th}}\left(\mathbf{W}\mathbf{a}^* + \mathbf{F}\mathbf{x}^* + \mathbf{b}\right)\right) = H(\mathbf{a}^*)$

$\mathbf{M} = \mathrm{Diag}(\mathbf{m}), \widetilde{\mathbf{W}} = \mathbf{M}\mathbf{W}, \text{ where } H(x) = \begin{cases} 1, & x > 0 \\ 0, & x \le 0 \end{cases}$

The gradient is $\nabla_\theta \mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial \theta}\right)^T = \left(\frac{\partial f_\theta(\mathbf{a}^*, \mathbf{x}^*)}{\partial \theta}\right)^T \left(I - \frac{1}{V_{th}} \widetilde{\mathbf{W}}^T\right)^{-1} \left(\frac{\partial \mathcal{L}}{\partial \mathbf{a}^*}\right)^T$

We can solve $\beta^* = \left(I - \frac{1}{V_{th}} \widetilde{\mathbf{W}}^T\right)^{-1} \left(\frac{\partial \mathcal{L}}{\partial \mathbf{a}^*}\right)^T$ by $\beta^{k+1} = \frac{1}{V_{th}} \widetilde{\mathbf{W}}^T \beta^k + \frac{\partial \mathcal{L}}{\partial \mathbf{a}^*}, \beta^k \to \beta^*$

- computing another equilibrium for these neurons by the inverse directions of connections
- M is a mask matrix based on the firing condition in the forward stage, inhibition mechanisms

Biological Plausibility

Two-stage equilibrium computation

Then $\nabla_{\mathbf{W}}\mathcal{L} = \frac{1}{V_{th}}\mathbf{M}\beta^*\mathbf{a}^{*T}, \nabla_{\mathbf{F}}\mathcal{L} = \frac{1}{V_{th}}\mathbf{M}\beta^*\mathbf{x}^{*T}$

$$\nabla_{\mathbf{W}_{i,j}}\mathcal{L} = \frac{1}{V_{th}}m_i\beta_i^* a_j^*, \nabla_{\mathbf{F}_{i,j}}\mathcal{L} = \frac{1}{V_{th}}m_i\beta_i^* x_j^*$$

Connection to the locally updated Hebbian learning rule $\Delta w_{i,j} \propto x_i x_j$

Difference:

- We consider average firing rates
- We use two-stage and temporal information
- There is a mask possibly based on inhibition mechanisms

More biologically plausible than BPTT-like methods; still other issues

Incorporating Multi-layer Structure

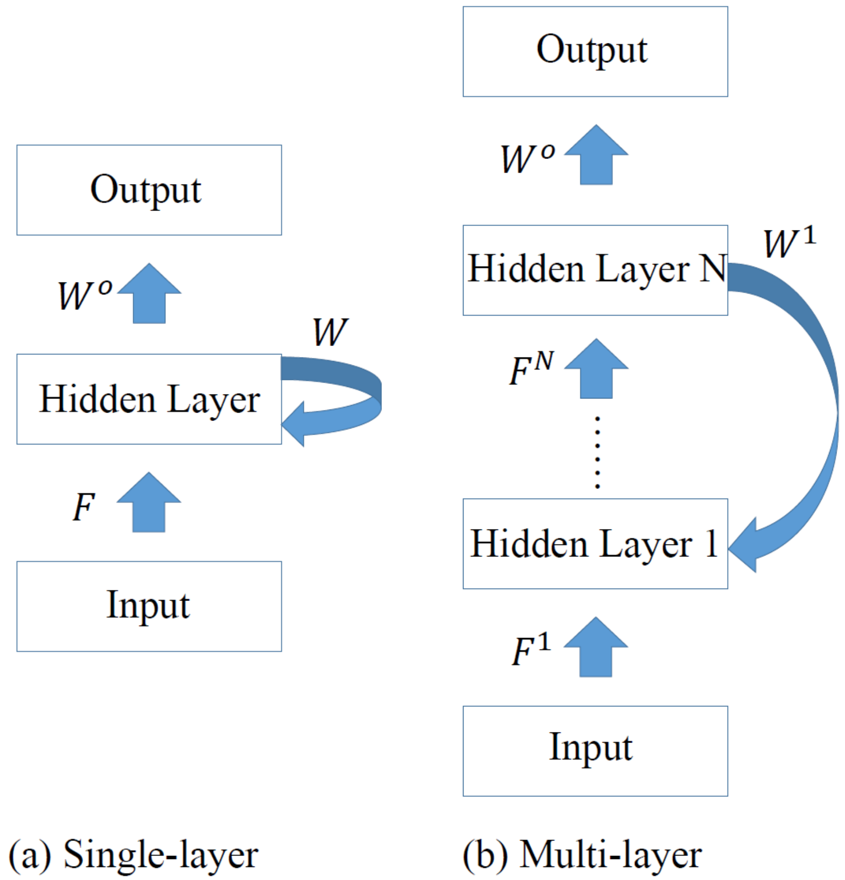
Enhance the non-linearity of the fixed-point equilibrium equation

IF model

Theorem 3. *If the average inputs converge to an equilibrium point* $\overline{\mathbf{x}}[t] \to \mathbf{x}^*$*, and there exists constant* c *and* $\gamma < 1$ *such that* $|\mathbf{u}_i^{l+}[t]| \le c, \forall i, l, t$ *and* $\|\mathbf{W}^1\|_2 \|\mathbf{F}^N\|_2 \cdots \|\mathbf{F}^2\|_2 \le \gamma V_{th}^N$*, then the average firing rates of multi-layer FSNN with discrete IF model will converge to equilibrium points* $\mathbf{a}^l[t] \to \mathbf{a}^{l*}$*, which satisfy the fixed-point equations* $\mathbf{a}^{1*} = f_1\left(f_N \circ \cdots \circ f_2(\mathbf{a}^{1*}), \mathbf{x}^*\right)$ *and* $\mathbf{a}^{l+1*} = f_{l+1}(\mathbf{a}^{l*})$*, where* $f_1(\mathbf{a}, \mathbf{x}) = \sigma\left(\frac{1}{V_{th}}(\mathbf{W}^1\mathbf{a} + \mathbf{F}^1\mathbf{x} + \mathbf{b}^1)\right)$, $f_{l+1}(\mathbf{a}) = \sigma\left(\frac{1}{V_{th}}(\mathbf{F}^{l+1}\mathbf{a} + \mathbf{b}^{l+1})\right)$.

LIF model

Proposition 3. *If the weighted average inputs converge to an equilibrium point* $\hat{\mathbf{x}}[t] \to \mathbf{x}^*$*, and there exists constant* c *and* $\gamma < 1$ *such that* $|\mathbf{u}_i^{l+}[t]| \le c, \forall i, l, t$ *and* $\|\mathbf{W}^1\|_2 \|\mathbf{F}^N\|_2 \cdots \|\mathbf{F}^2\|_2 \le \gamma V_{th}^N$*, then the weighted average firing rates* $\hat{\mathbf{a}}^l[t]$ *of multi-layer FSNN with discrete LIF model gradually approximate equilibrium points* $\mathbf{a}^{l*}$ *with bounded random errors, which satisfy* $\mathbf{a}^{1*} = f_1\left(f_N \circ \cdots \circ f_2(\mathbf{a}^{1*}), \mathbf{x}^*\right)$ *and* $\mathbf{a}^{l+1*} = f_{l+1}(\mathbf{a}^{l*})$*, where* $f_1(\mathbf{a}, \mathbf{x}) = \sigma\left(\frac{1}{V_{th}}(\mathbf{W}^1\mathbf{a} + \mathbf{F}^1\mathbf{x} + \mathbf{b}^1)\right)$, $f_{l+1}(\mathbf{a}) = \sigma\left(\frac{1}{V_{th}}(\mathbf{F}^{l+1}\mathbf{a} + \mathbf{b}^{l+1})\right)$.



Implementation Details

Restriction on Spectral Norm

$$\mathbf{W} = \alpha \frac{\mathbf{W}}{\|\mathbf{W}\|_2} \qquad \alpha \in [-c, c]$$

Modified Batch Normalization

BN with fixed statistics is a simple linear transformation;

Fix the statistics during forward computation;

Update the statistics during backward calculation.

Experiments

Simple Static Inputs

- Relatively small time steps
- Fewer neurons and params
- LIF slightly better than IF
  - LIF leverages temporal information by encoding weighted average firing rates

MNIST

Method	Network structure	Time steps	Mean±Std	Best	Neurons	Params
BP [21]	20C5-P2-50C5-P2-200	>200	/	99.31%	33K	518K
STBP [41]	15C5-P2-40C5-P2-300	30	/	99.42%	26K	607K
SLAYER [38]	12C5-P2-64C5-P2	300	99.36%±0.05%	99.41%	28K	51K
HM2BP [14]	15C5-P2-40C5-P2-300	400	99.42%±0.11%	99.49%	26K	607K
ST-RSBP [45]	15C5-P2-40C5-P2-300	400	**99.57%±0.04%**	**99.62%**	26K	607K
TSSL-BP [46]	15C5-P2-40C5-P2-300	5	99.50%±0.02%	99.53%	26K	607K
IDE-IF (ours)	64C5s (F64C5)	30	99.49%±0.04%	99.55%	13K	229K
IDE-LIF (ours)	64C5s (F64C5)	30	99.53%±0.04%	99.59%	13K	229K

Fashion-MNIST

Method	Network structure	Time steps	Mean±Std	Best	Neurons	Params
ANN [45]	512-512	/	/	89.01%	1.8K	670K
HM2BP [45]	400-400	400	/	88.99%	1.6K	478K
TSSL-BP [46]	400-400	5	89.75%±0.03%	89.80%	1.6K	478K
ST-RSBP [45]	400 (F400)	400	90.00%±0.14%	90.13%	1.2K	478K
IDE-IF (ours)	400 (F400)	**5**	**90.04%±0.09%**	**90.14%**	1.2K	478K
IDE-LIF (ours)	400 (F400)	**5**	**90.07%±0.10%**	**90.25%**	1.2K	478K

Experiments

Neuromorphic Inputs

- Relatively small time steps
- Fewer neurons and params
- LIF slightly better than IF
  - LIF leverages temporal information by encoding weighted average firing rates

Table 2: Performance on N-MNIST. Results are based on 5 runs of experiments.

Method	Network structure	Time steps	Mean±Std	Best	Neurons	Params
HM2BP [14]	400-400	600	98.88%±0.02%	98.88%	3K	1.1M
SLAYER [38]	500-500	300	98.89%±0.06%	98.95%	3K	1.4M
SLAYER [38]	12C5-P2-64C5-P2	300	99.20%±0.02%	99.22%	40K	61K
TSSL-BP [46]	12C5-P2-64C5-P2	30	99.23%±0.05%	99.28%	40K	61K
STBP w/o NeuNorm [42]	CNN[1]	60	/	99.44%	414K	17.3M
IDE-IF (ours)	64C5s (F64C5)	30	99.30%±0.04%	99.35%	21K	291K
IDE-LIF (ours)	64C5s (F64C5)	**30**	**99.42%±0.04%**	**99.47%**	21K	291K

[1] 128C3-128C3-P2-128C3-256C3-P2-1024

Experiments

Complex Static Inputs

- Relatively small time steps
- Fewer neurons and params
- Superior results

CIFAR-10

Method	Network structure	Time steps	Mean±Std	Best	Neurons	Params
ANN-SNN [8]	CIFARNet	400-600	/	90.61%	726K	45M
ANN-SNN [37]	VGG-16	2500	/	91.55%	311K	15M
ANN-SNN [8]	VGG-16	400-600	/	92.26%	318K	40M
Hybrid Training [32]	VGG-16	100	/	91.13%	318K	40M
STBP [42]	AlexNet	12	/	85.24%	595K	21M
TSSL-BP [46]	AlexNet	5	88.98%±0.27%	89.22%	595K	21M
STBP [42]	CIFARNet	12	/	90.53%	726K	45M
TSSL-BP [46]	CIFARNet	5	/	91.41%	726K	45M
Surrogate gradient [20]	VGG-9	100	/	90.45%	274K	5.9M
ASF-BP [40]	VGG-7	400	/	91.35%	>240K	>30M
IDE-LIF (ours)	AlexNet-F	30	91.74%±0.09%	91.92%	159K	3.7M
IDE-LIF (ours)	AlexNet-F	100	**92.03%±0.07%**	**92.15%**	**159K**	**3.7M**
IDE-LIF (ours)	CIFARNet-F	30	92.08%±0.14%	92.23%	232K	11.8M
IDE-LIF (ours)	CIFARNet-F	100	**92.52%±0.17%**	**92.82%**	**232K**	**11.8M**

[1] AlexNet [42]: 96C3-256C3-P2-384C3-P2-384C3-256C3-1024-1024
[2] AlexNet-F: 96C3s-256C3-384C3s-384C3-256C3 (F96C3u)
[3] CIFARNet [42]: 128C3-256C3-P2-512C3-P2-1024C3-512C3-1024-512
[4] CIFARNet-F: 128C3s-256C3-512C3s-1024C3-512C3 (F128C3u)

Experiments

Complex Static Inputs

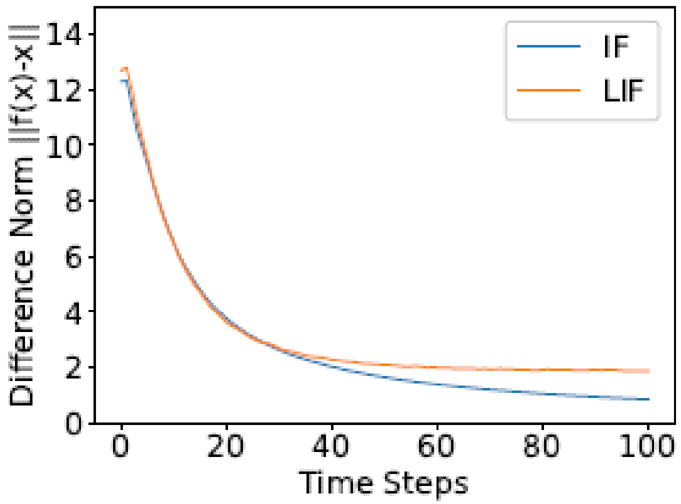
- Relatively small time steps
- Fewer neurons and params
- Superior results

CIFAR-100

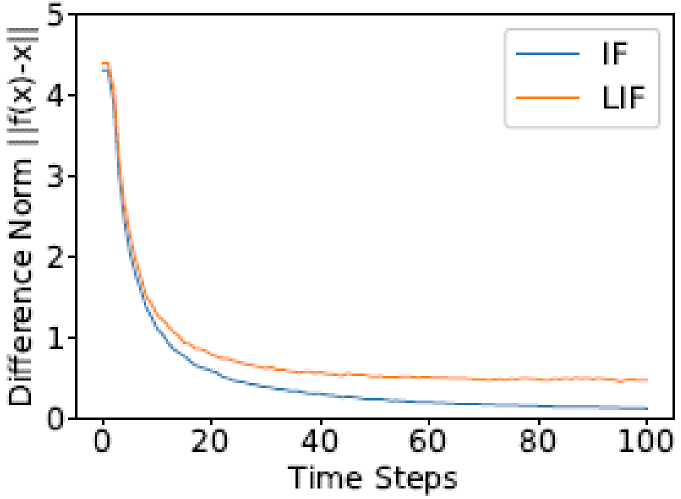
Method	Network structure	Time steps	Mean±Std	Best	Neurons	Params
ANN [37]	VGG-16	/	/	71.22%	311K	15M
ANN-SNN [37]	VGG-16	2500	/	70.77%	311K	15M
ANN-SNN [8]	VGG-16	400-600	/	70.55%	318K	40M
ANN-SNN [44]	VGG-*	300	/	71.84%	540K	9.7M
IDE-IF (ours)	CIFARNet-F	**30**	**71.56%±0.31%**	**72.10%**	232K	14.8M
IDE-IF (ours)	AlexNet-F	**100**	**72.02%±0.16%**	**72.23%**	**159K**	**5.2M**
IDE-IF (ours)	CIFARNet-F	**100**	**73.07%±0.21%**	**73.43%**	232K	14.8M

Experiments

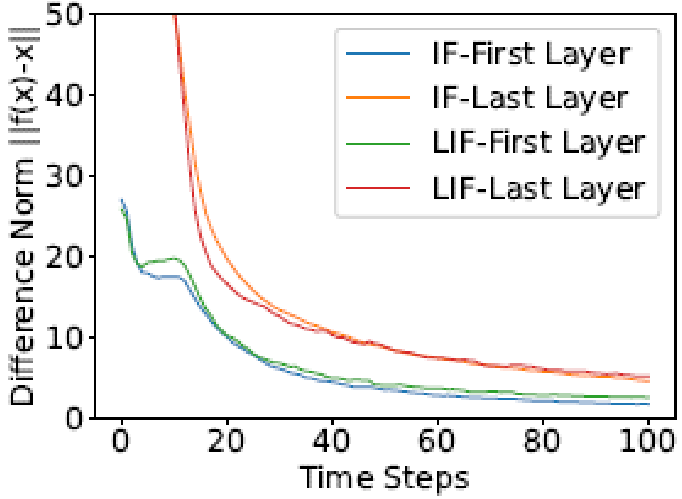
Convergence to Equilibrium



(a) MNIST: 64C5s (F64C5)



(b) Fashion-MNIST: 400 (F400)



(c) CIFAR-10: CIFARNet-F

Experiments

Training Memory Costs

Method	Time steps	Accuracy	GPU memory
IDE (ours)	30	91.74%±0.09%	2.8G
STBP*	30	87.18%	11G
IDE (ours)	100	92.03%±0.07%	2.8G
STBP*	100	/	out of memory (≈36G)

* Our implementation

Firing Sparsity

Layer	IDE-IF	IDE-LIF	STBP-LIF*
Layer 1	0.0345	0.0166	0.0190
Layer 2	0.0041	0.0039	0.0082
Layer 3	0.0025	0.0024	0.0113
Layer 4	0.0008	0.0008	0.0055
Layer 5	0.0399	0.0177	0.0108
Total	0.0119	**0.0066**	0.0102

* Our implementation

Conclusion

Novel training method based on *equilibrium states* and *implicit differentiation*

- The forward computation of FSNNs can be interpreted as *solving a fixed-point equation*
- The backward calculation is *decoupled from* the forward computational graph
- *Avoid common SNN training problems* including non-differentiability and large memory costs
- Discussion of *biological plausibility* and connection to the Hebbian learning rule

Superior performance with *fewer neurons and parameters* in a *small number of time steps*, and *spikes are sparse*.

Future work

- Consideration of neuromorphic learning and biological issues
- Feedback network structures
- Large-scale datasets
- ……

Thanks for listening!

Paper

https://arxiv.org/abs/2109.14247

GitHub

https://github.com/pkuxmq/IDE-FSNN