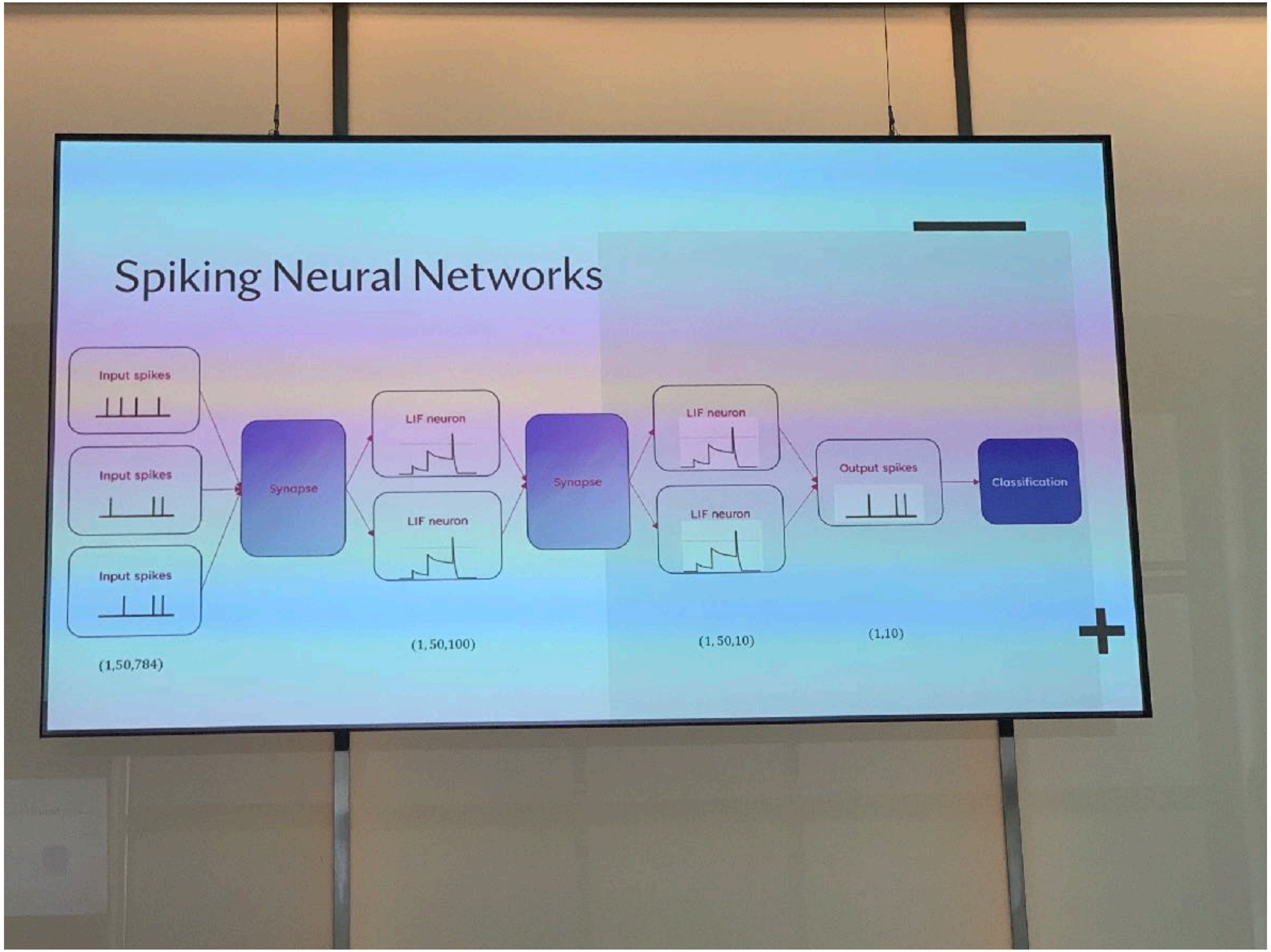
Exact Gradient Computation for Spiking Neural Networks

Amin Karbasi

Yale

Google AI

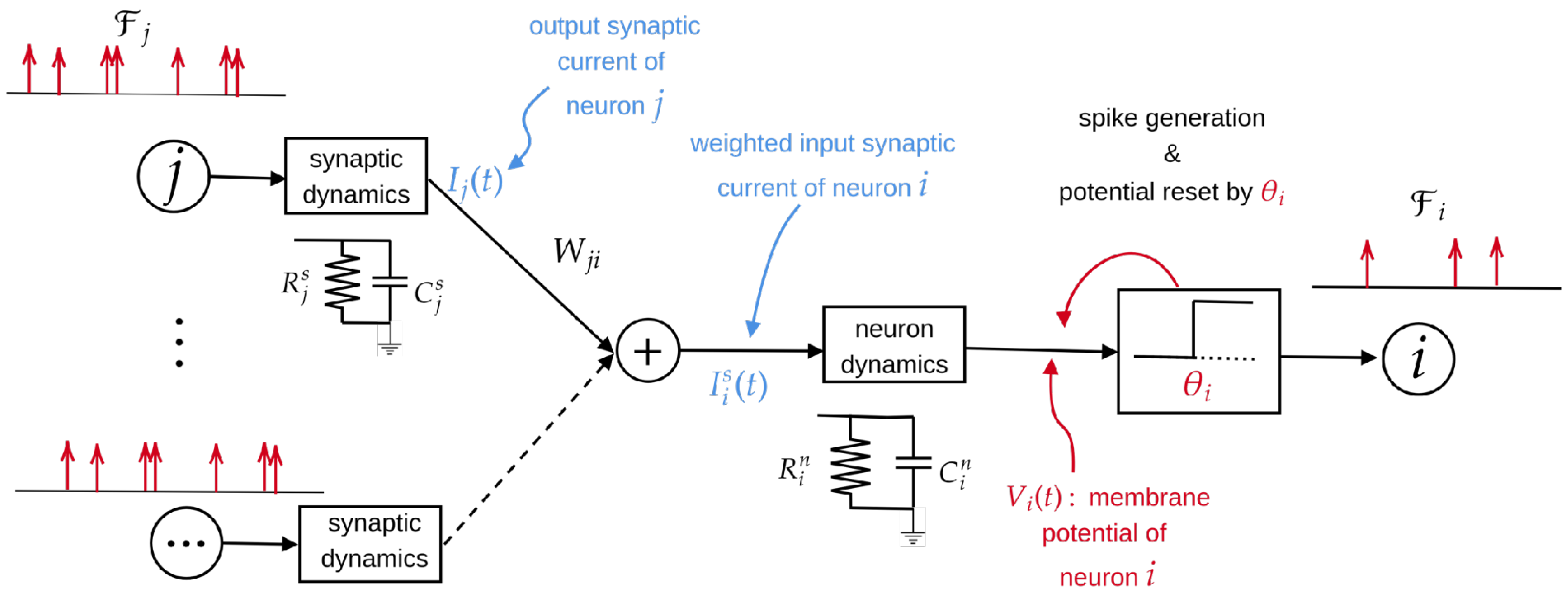
Spiking Neural Networks

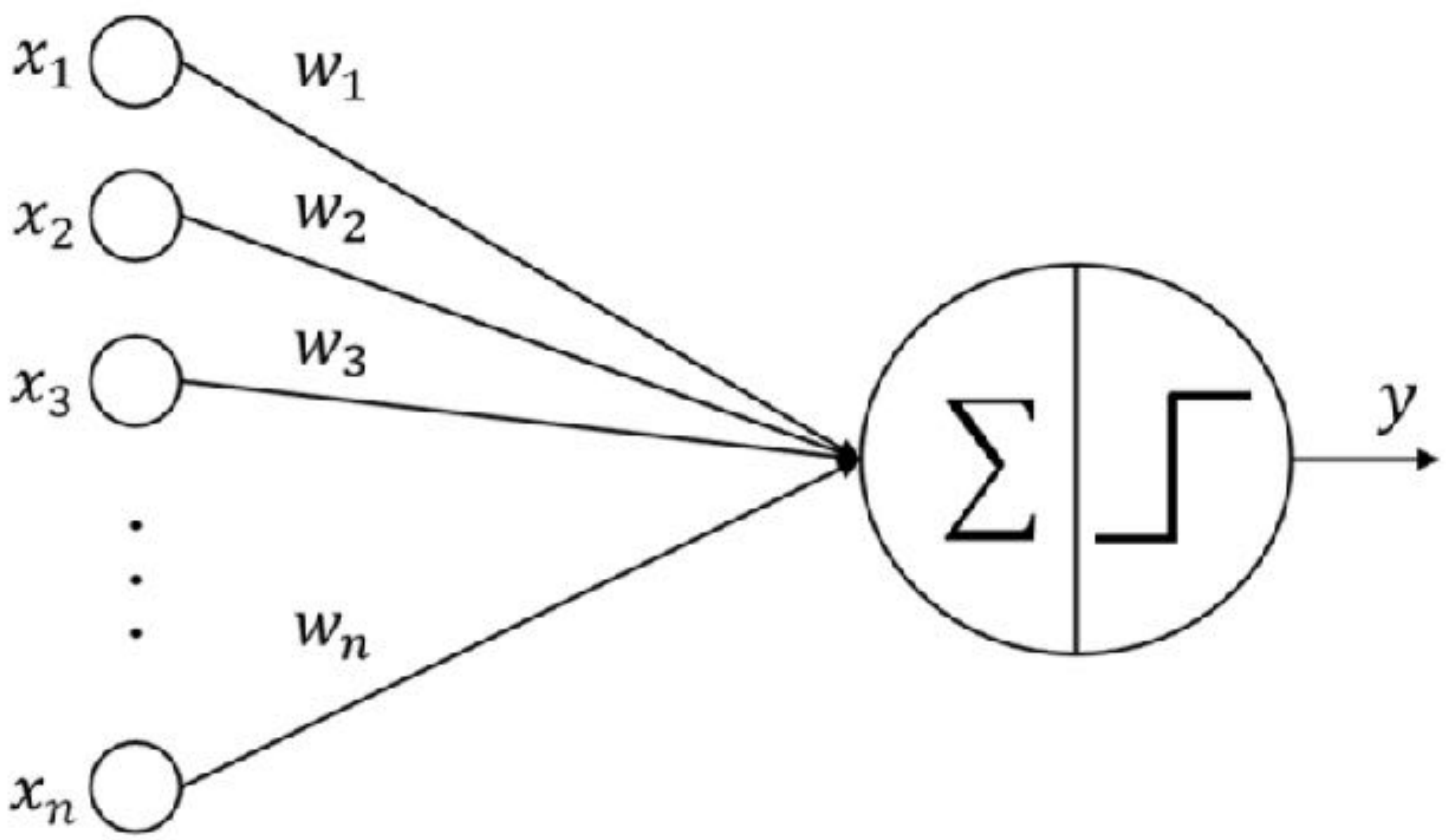


Spiking Neural Networks

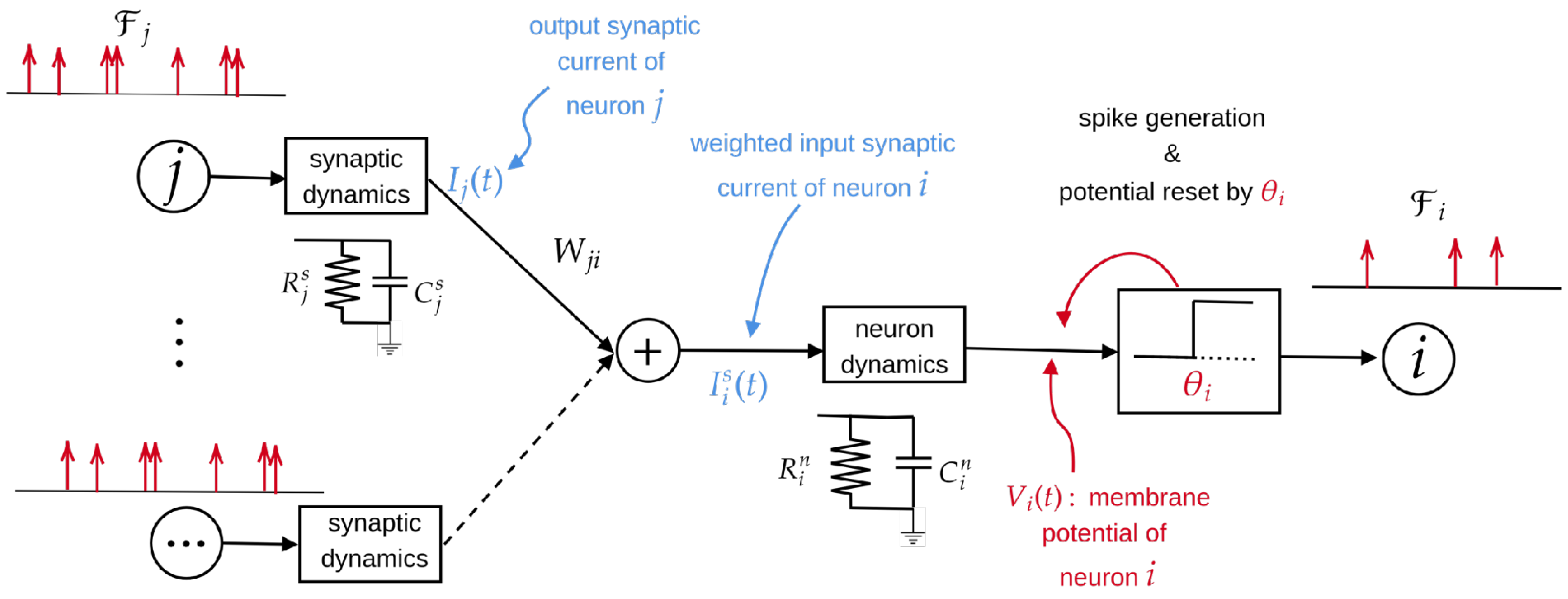
SNN versus ANN

- Key Differences:
  - Input representation: continuous vs discrete
  - Connections between neurons have some **dynamics**.
  - Neurons have internal **membrane potential**, but outputs **spike** when that potential reaches a threshold, after which it resets.

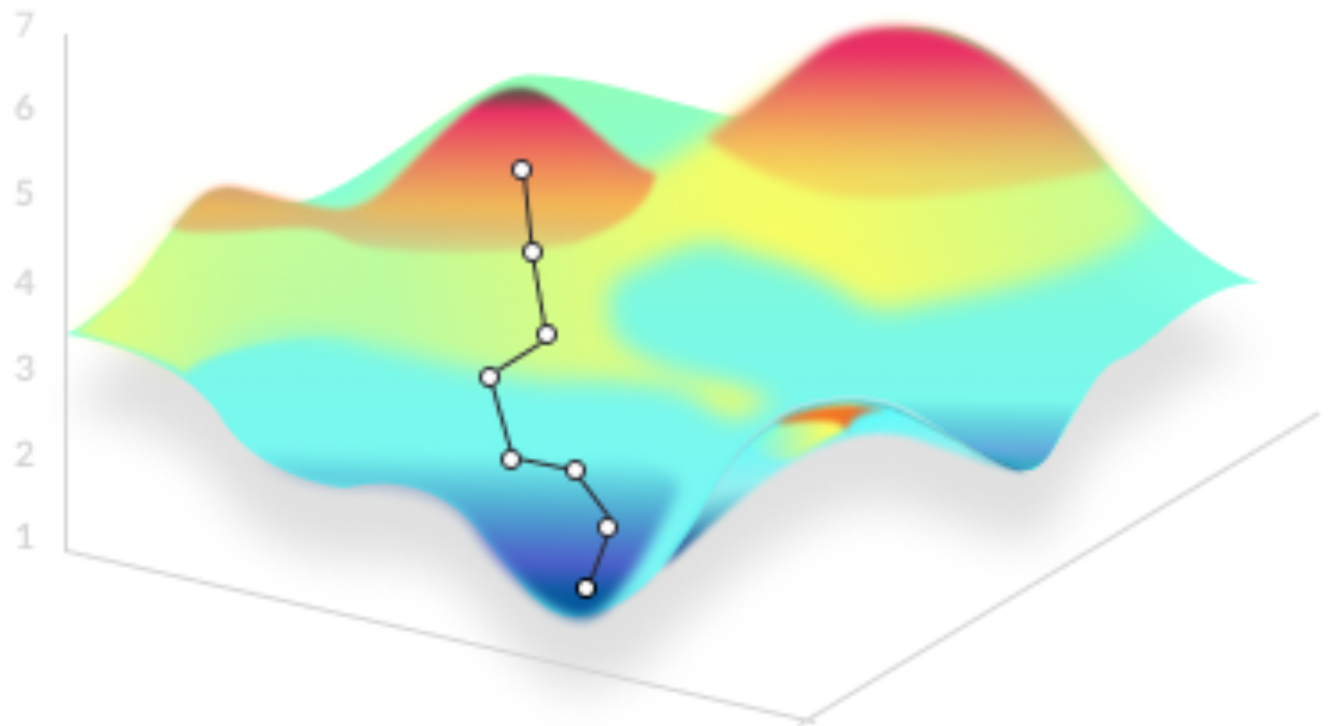


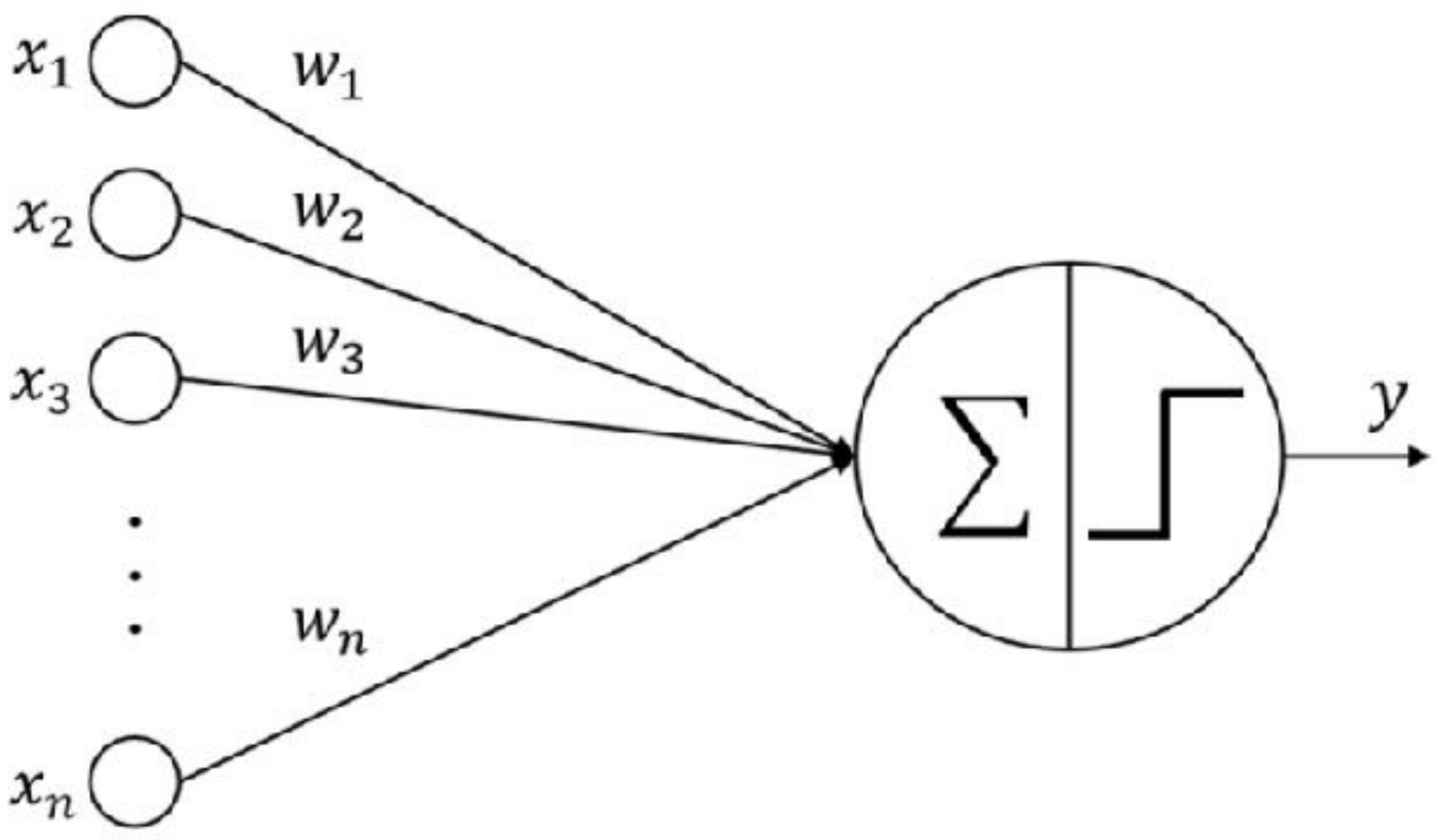


SNN versus ANN



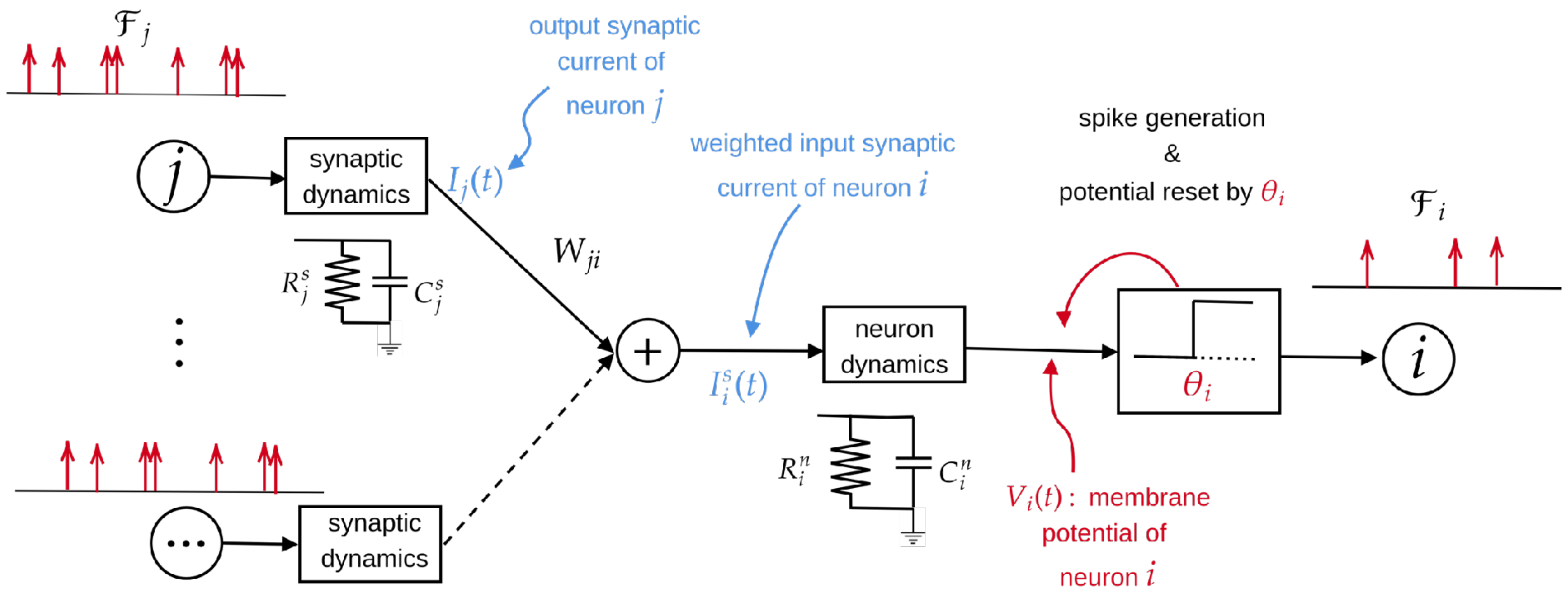
- How do we train ANN?
  - Gradient descent via back-propagation

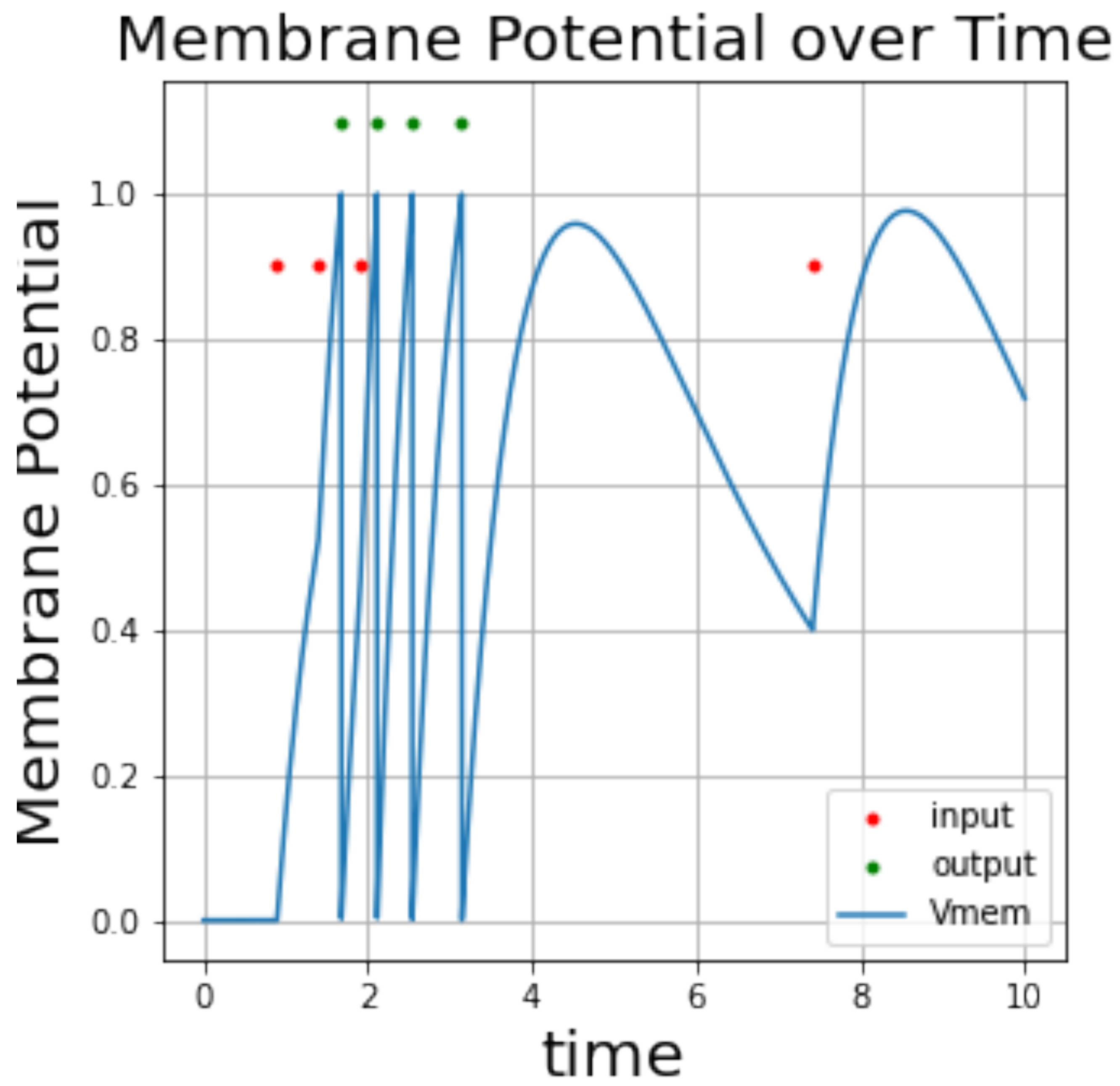


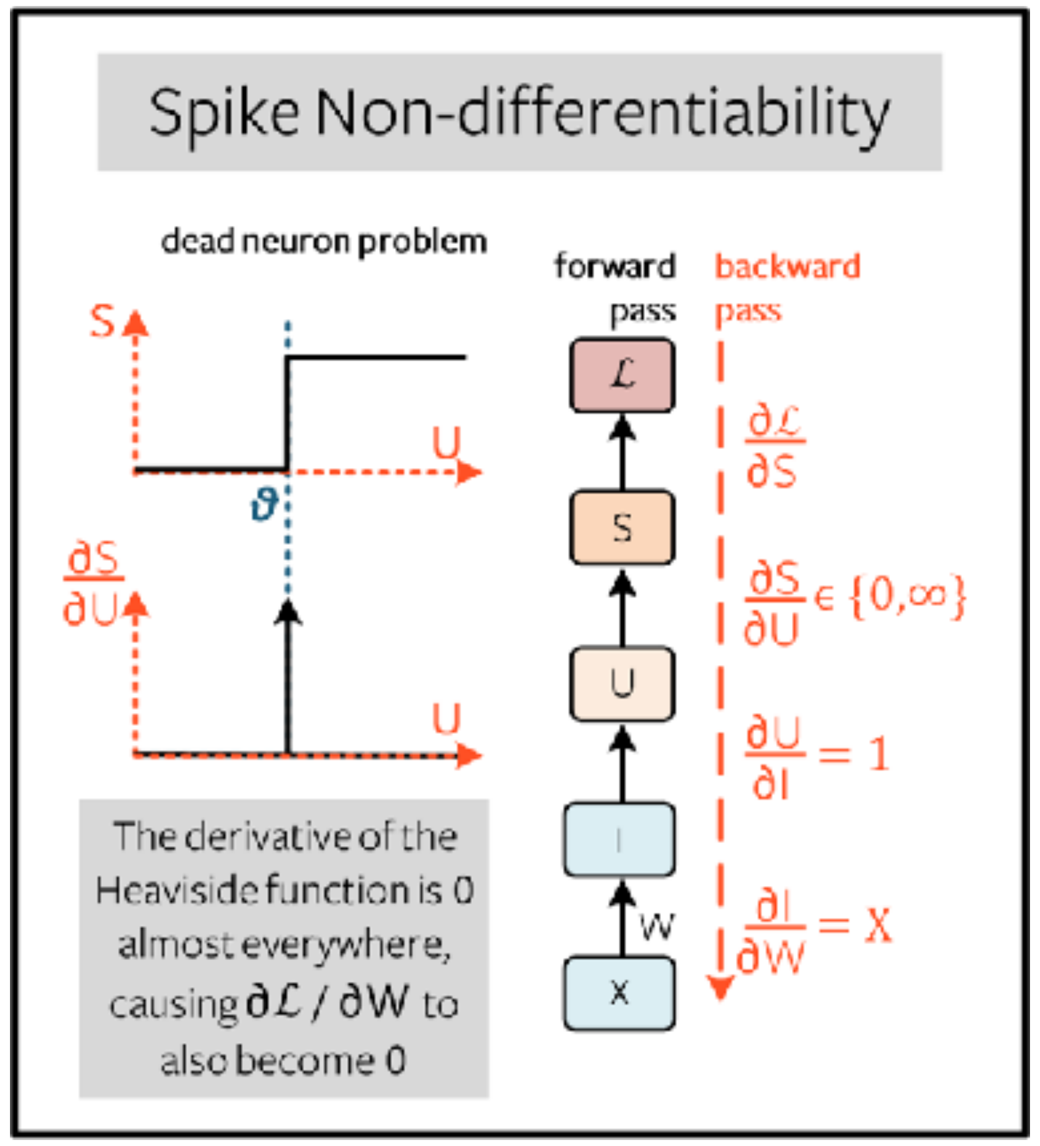


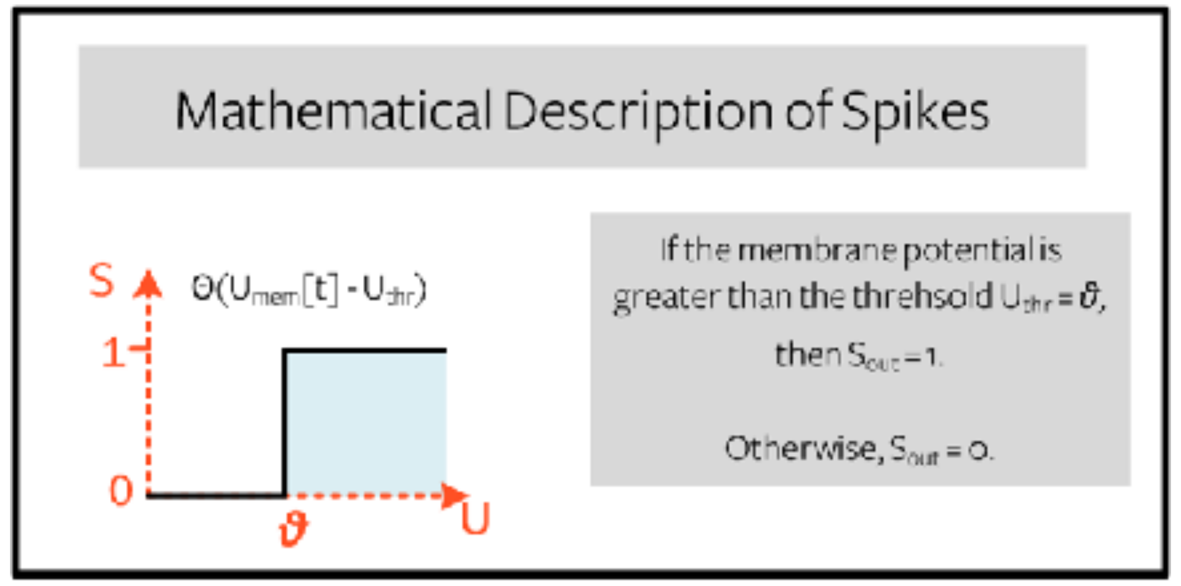
Main Question

- How do we train SNN?
  - Spikes are not differentiable functions!!!
  - Use surrogate gradients + back propagation



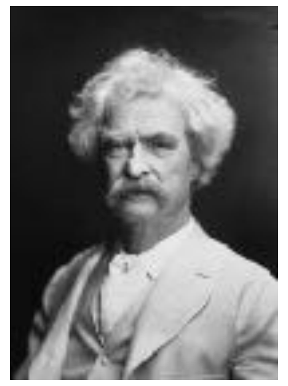




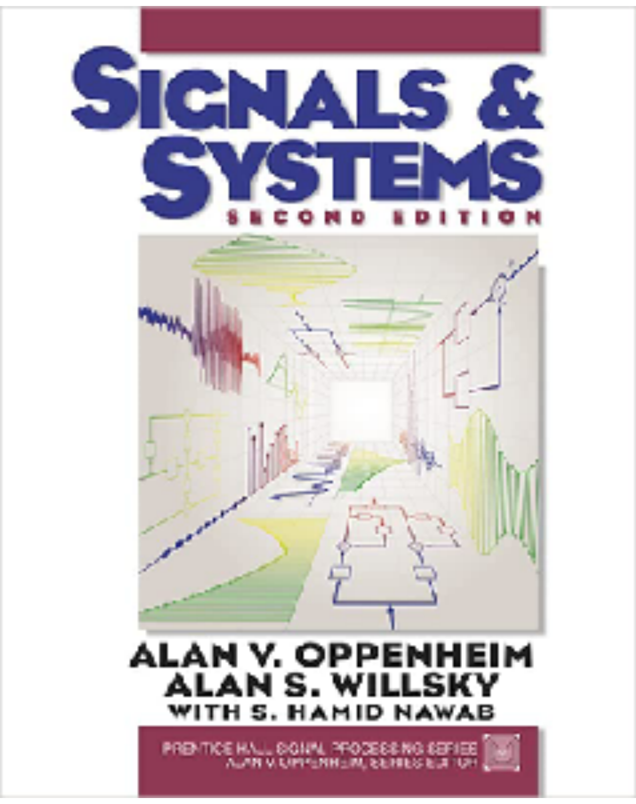


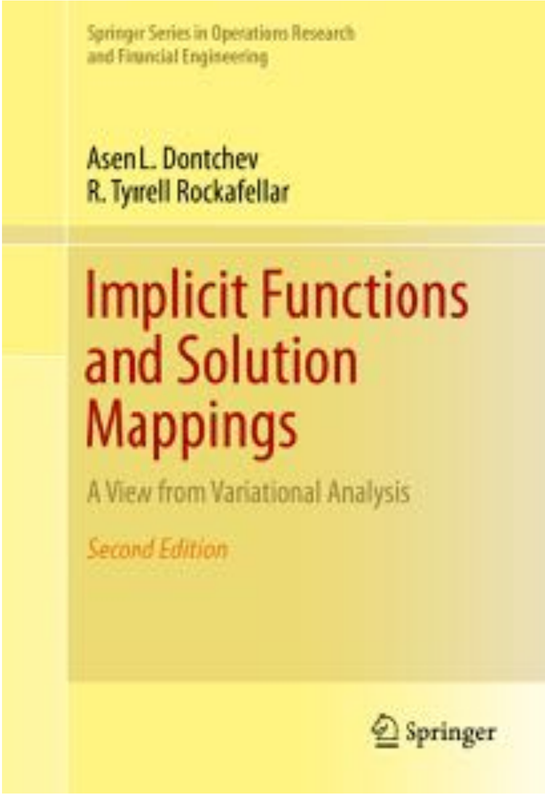
Images from snnTorch by Jason K. Eshraghian

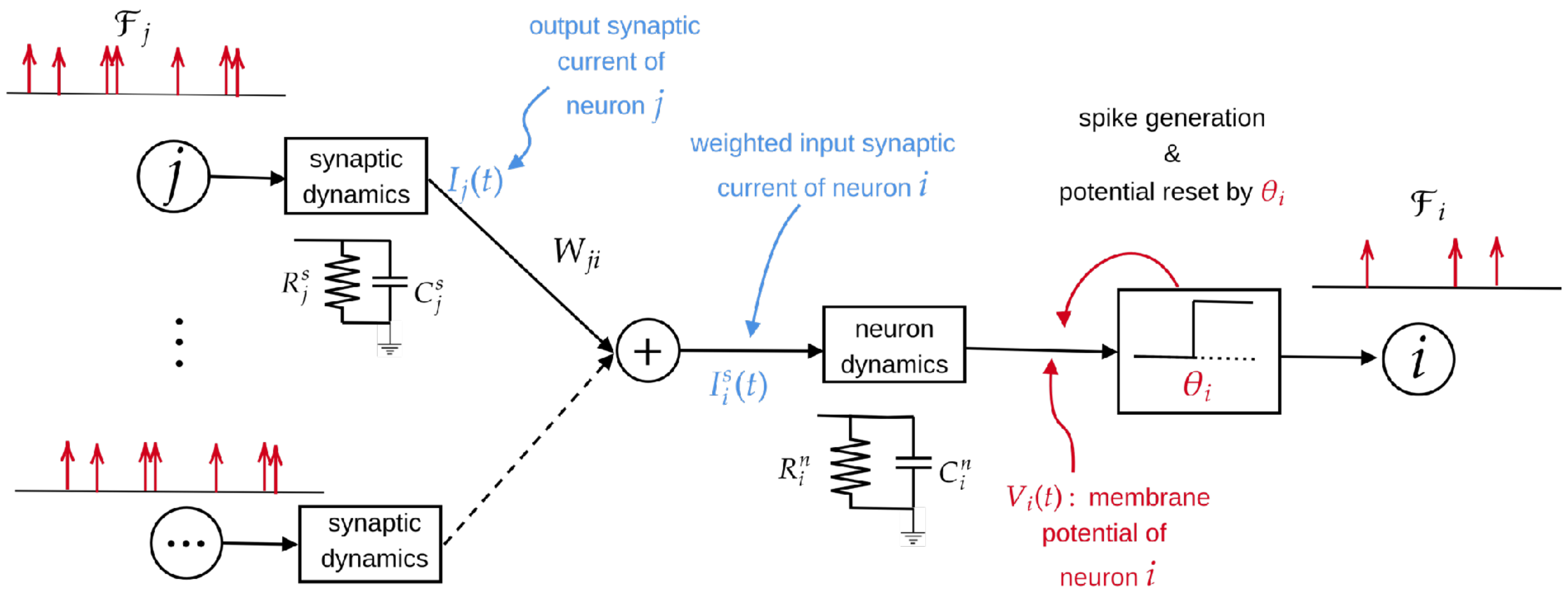
Do Gradients Exist?

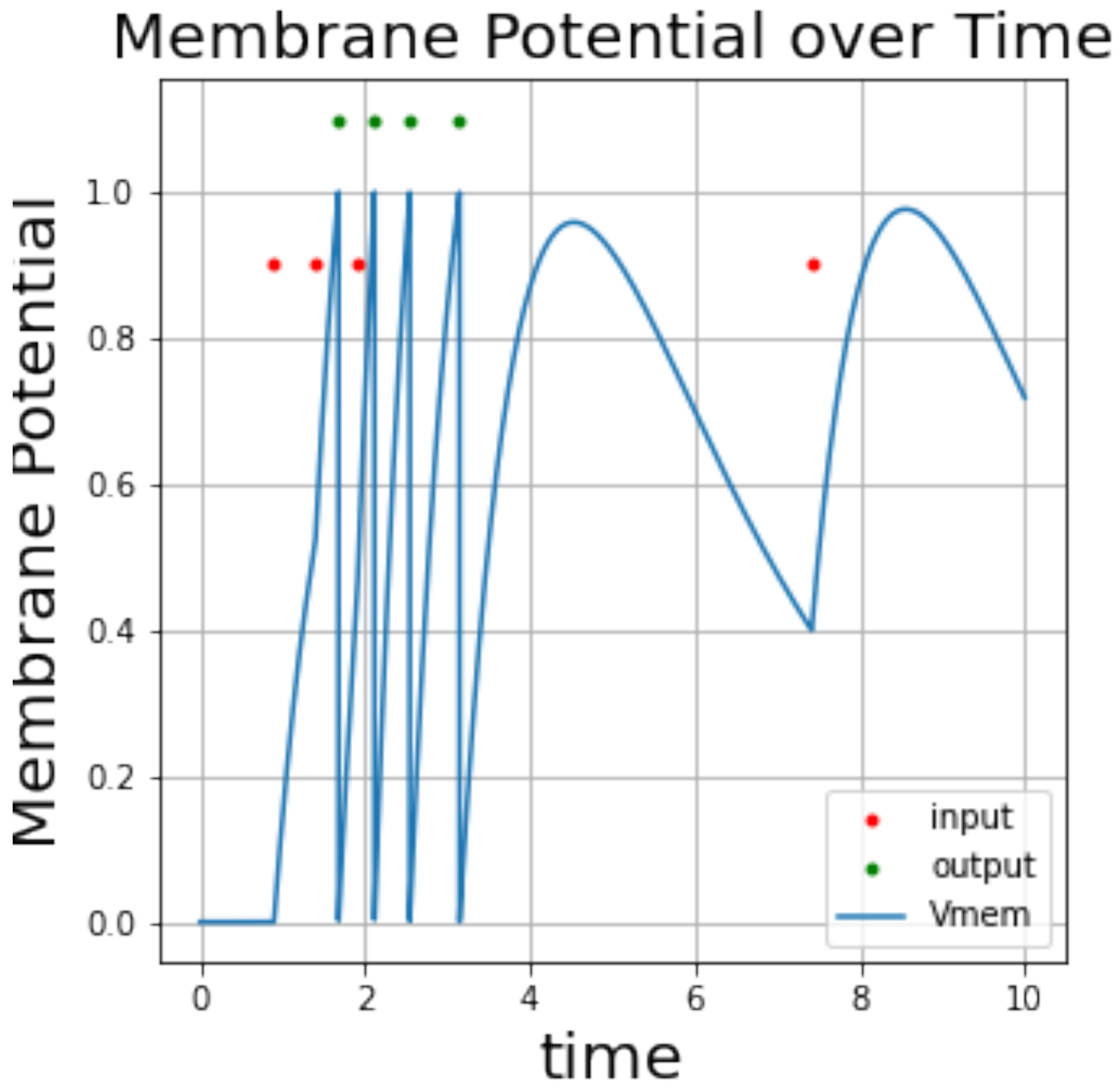


There is no such thing as a new idea

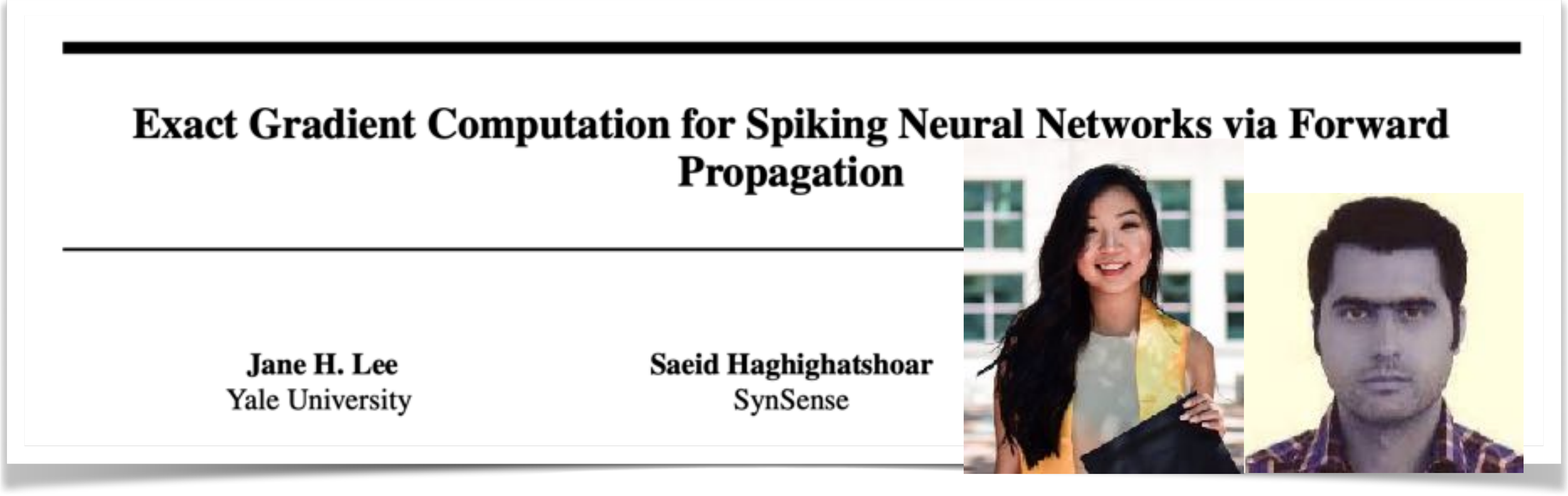








Yes, they do and we can compute them



Exact Gradient Computation for Spiking Neural Networks via Forward Propagation

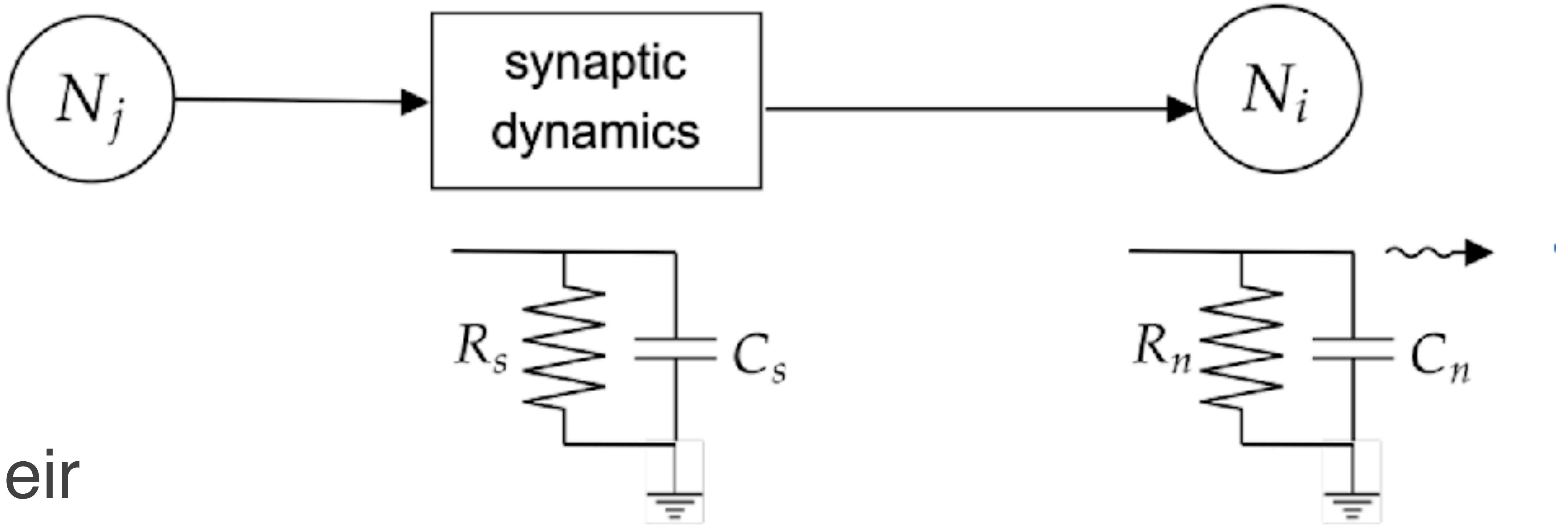
Jane H. Lee
Yale University

Saeid Haghighatshoar
SynSense

Leaky Integrate and Fire (LIF)

- Synaptic and neuron internal state dynamics are modeled as two RC circuits, govern by a differential equation

$$\tau \frac{dV}{dt} = -V + I$$



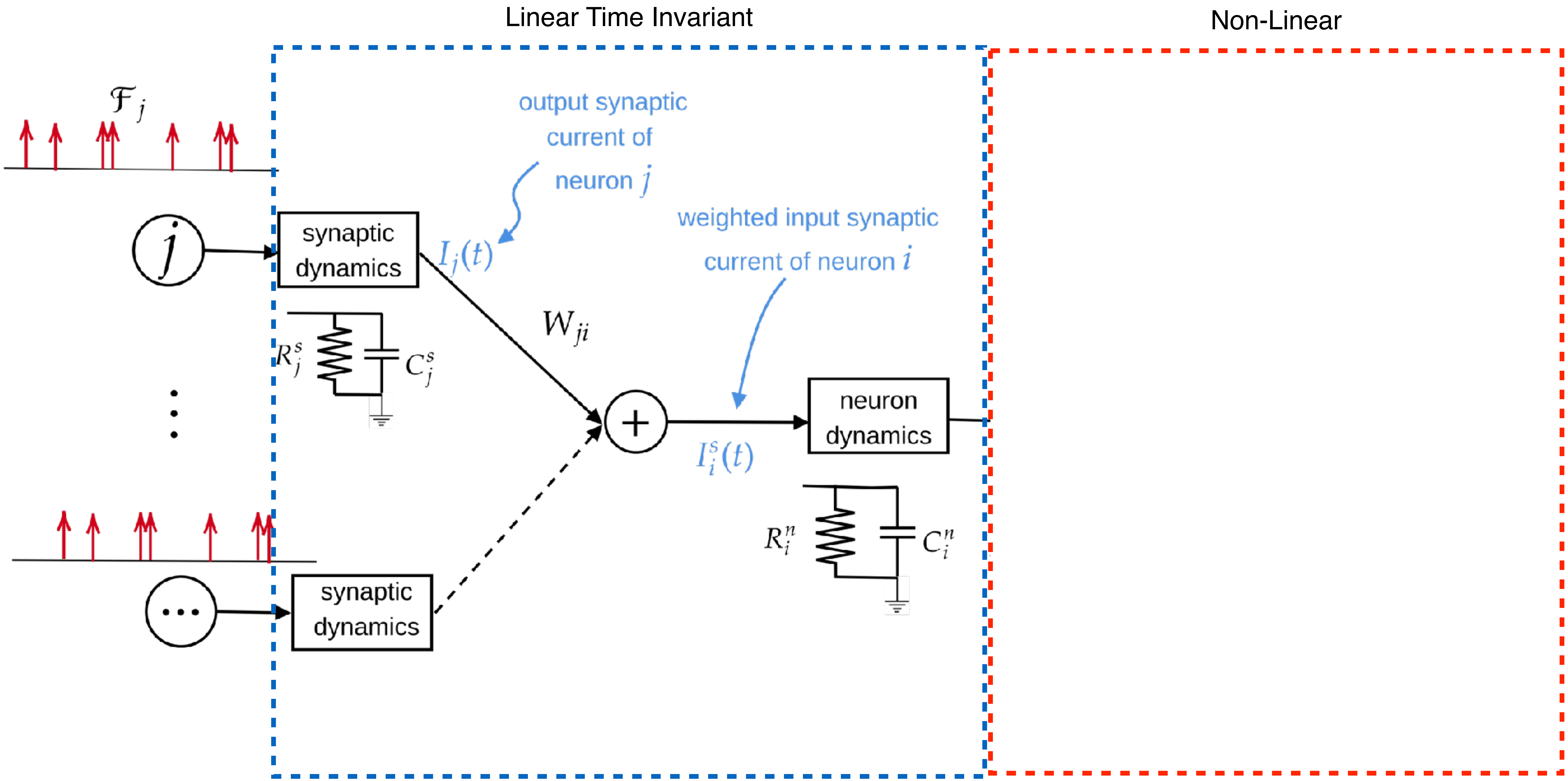
- Equivalently, they can be described by their impulse response

$$h^s(t) = e^{-\frac{1}{\tau_s}t}u(t) \qquad h^n(t) = e^{-\frac{1}{\tau_n}t}u(t)$$

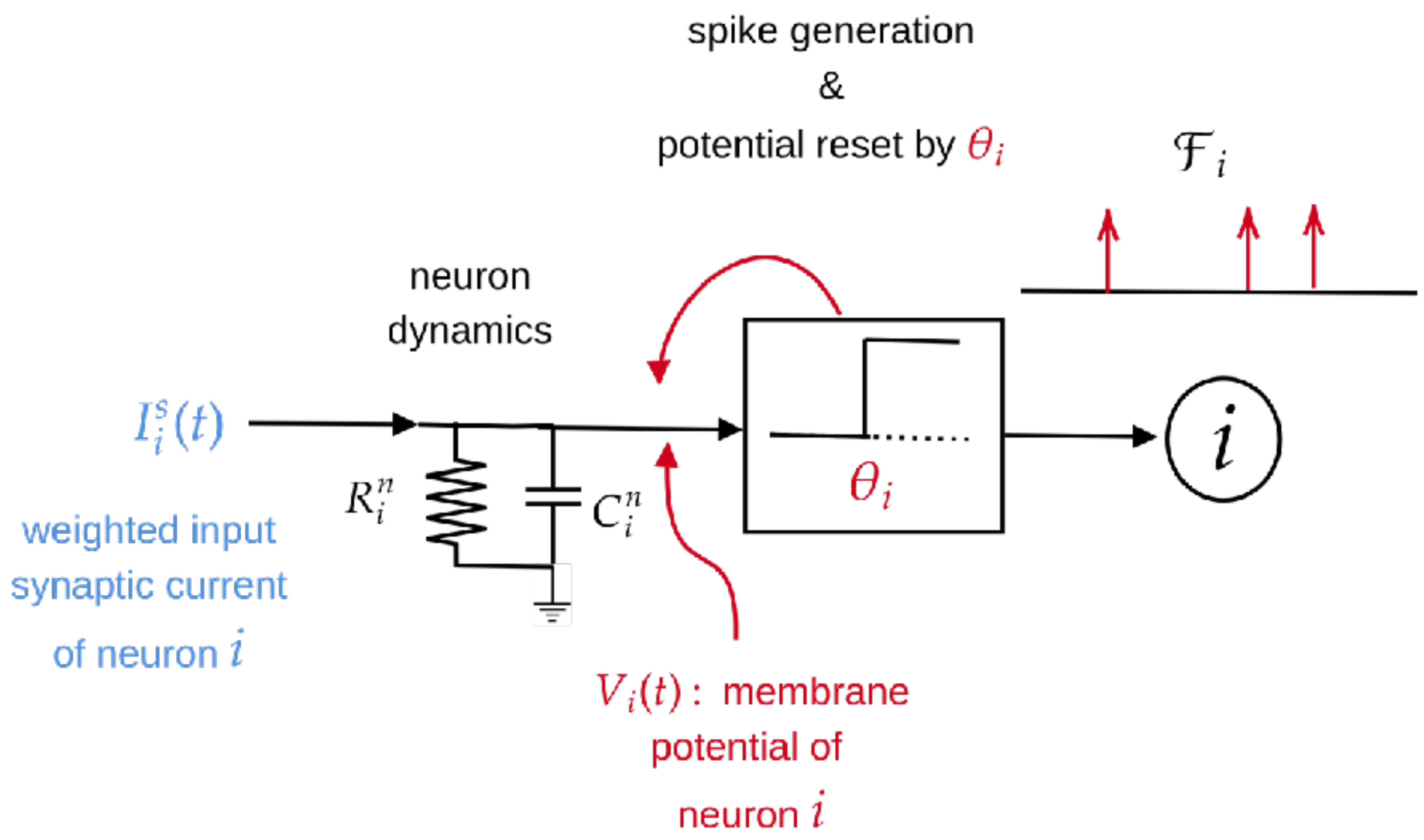
- The output synaptic current I_j for neuron j

$$I_j(t) = h_j^s(t) \star \sum_{f \in \mathcal{F}_j} \delta(t - f) = \sum_{f \in \mathcal{F}_j} h_j^s(t - f)$$

Pre-Synaptic Model



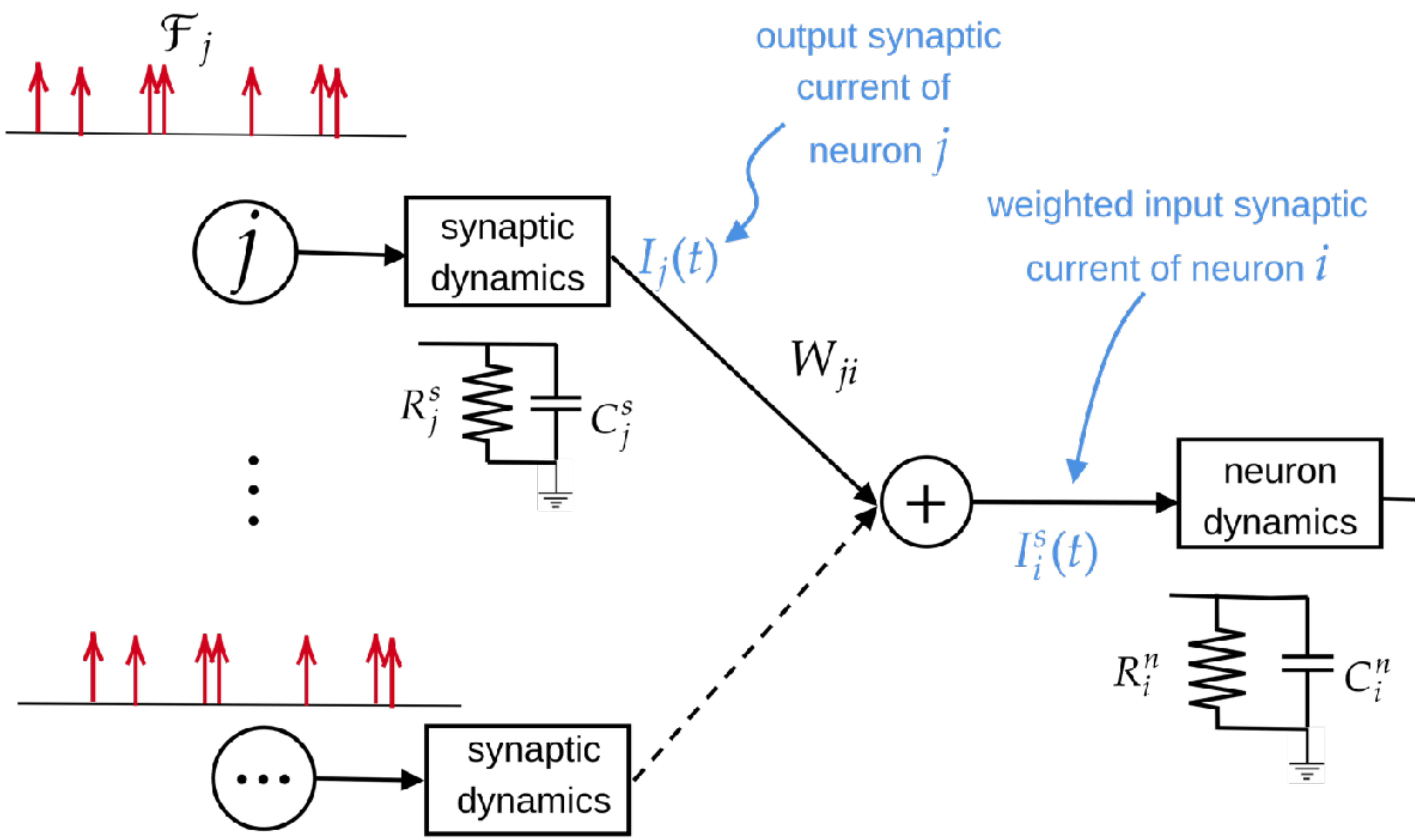
Non-Linearity



A **nonlinear** neuron with weighted synaptic currents $I(t)$ and spike generation

A **linear** neuron with input $I(t)$ and Heaviside voltages $\{-\theta_i u(t-f) : f \in \mathcal{F}_i\}$

Post-Synaptic Model



- Joint impulse response

$$h_{ji}(t) = h_j^s(t) \star h_i^n(t)$$

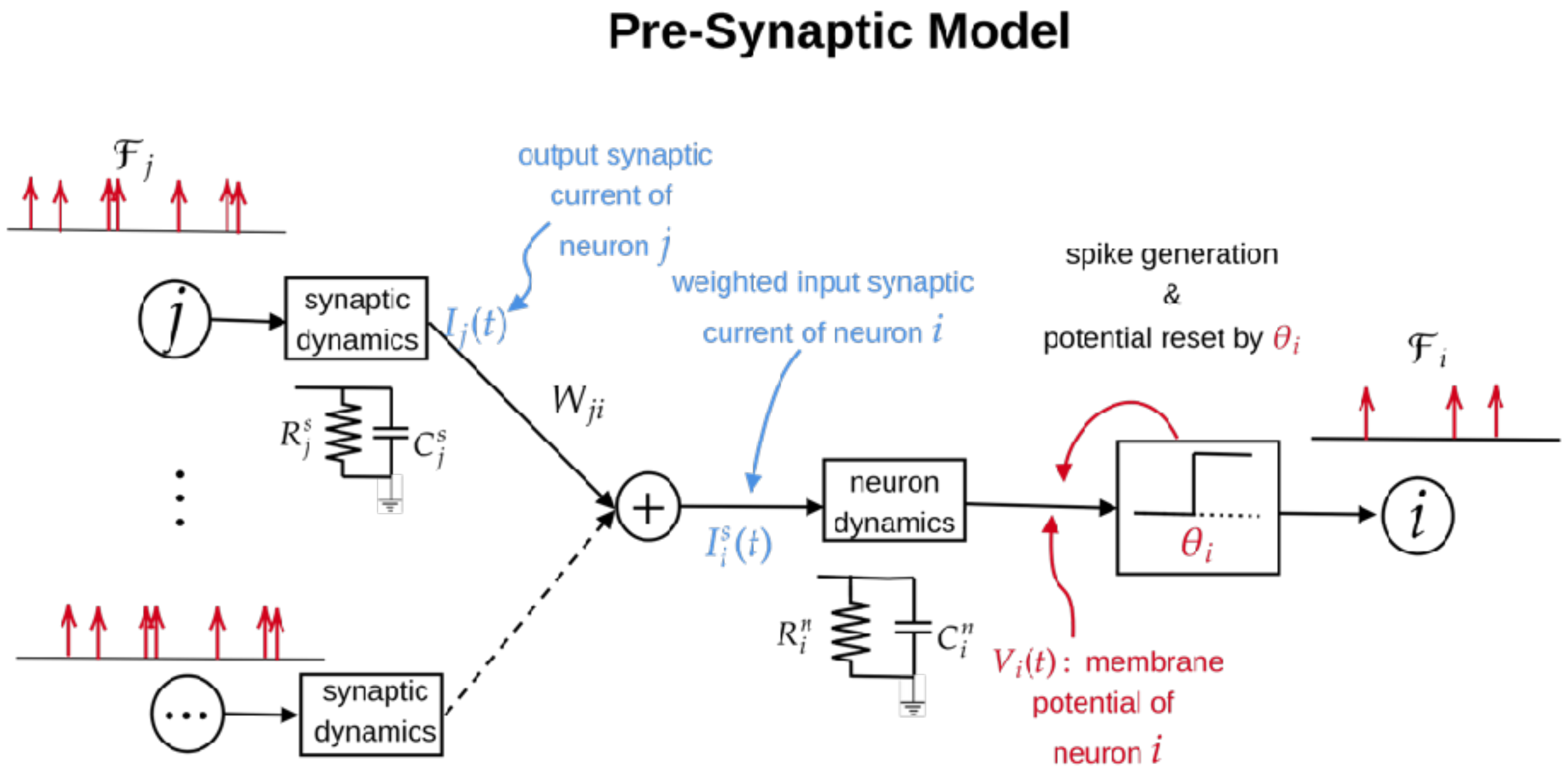
- Effects of spikes $\mathcal{F}_j$

$$y_{ji}(t) = \sum_{f \in \mathcal{F}_j} h_{ji}(t - f)$$

- Effects of all spikes

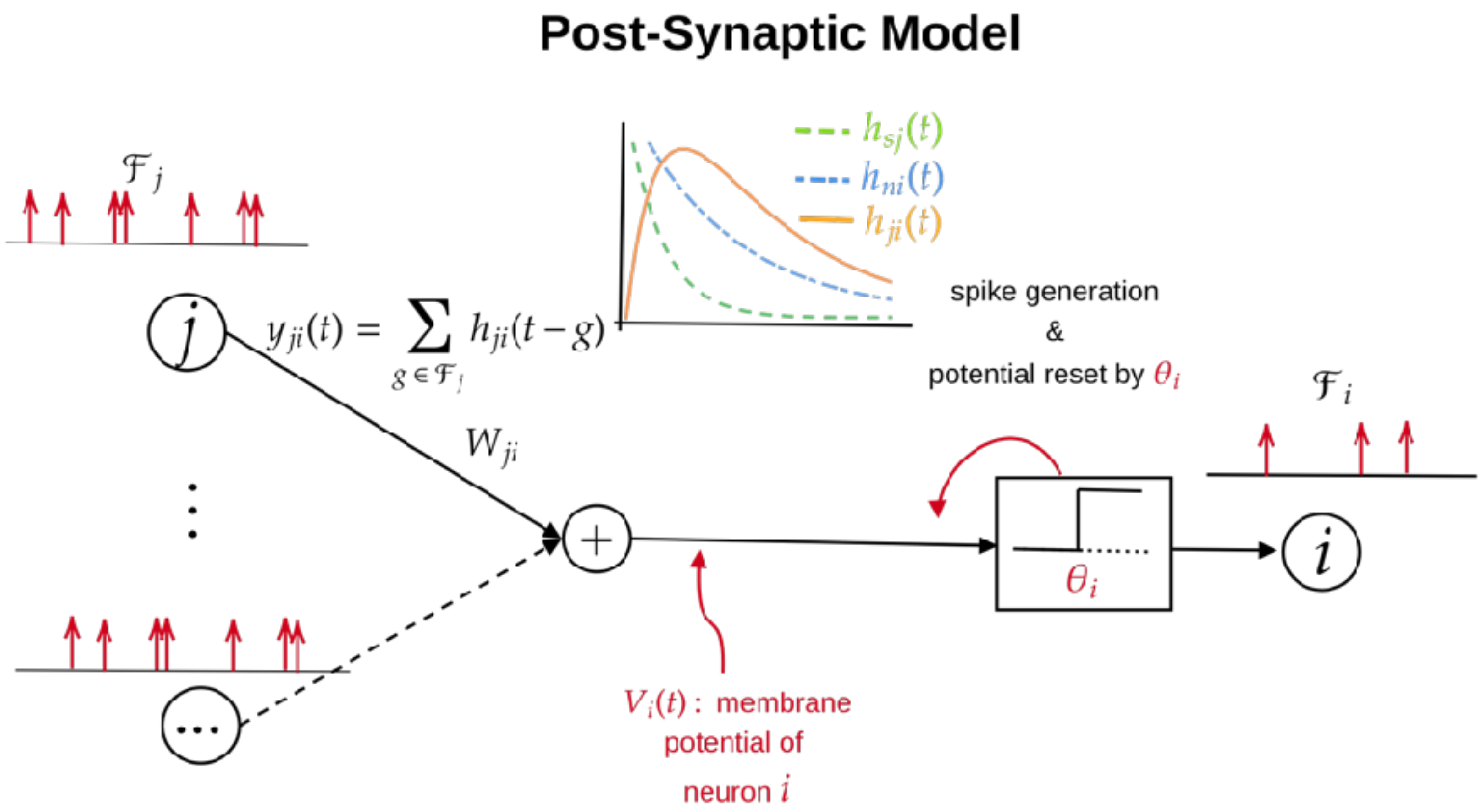
$$V_i^\circ(t) = \sum_{j \in \mathcal{N}_i} W_{ji} y_{ji}(t)$$

Post-Synaptic Model



- In this model neuron $j \in \mathcal{N}_i$ stimulates neuron i through **abrupt** spiking signal

$$\sum_{g \in \mathcal{F}_j} \delta(t-g)$$



- In this model neuron $j \in \mathcal{N}_i$ stimulates neuron i through the **smooth** kernel

$$V_i(t) = V_i^\circ(t) - \sum_{f \in \mathcal{F}_i} \theta_i h_i^n(t-f)$$

$$\sum_{g \in \mathcal{F}_i} h_{ji}(t-g)$$

Loss Function

- W consider a quite generic loss function:

$$\mathscr{L} = \ell_{\mathscr{F}}(\mathscr{F}; W) + \int_0^T \ell_V(V_o(t), \mathscr{F}; W)dt$$

Classification — Regression

[**Lee, Haghighatshoar**, Karbasi]

Theorem:

I. The loss $\mathscr{L}$ depends only on the spike firing times $\mathscr{F}$ and the weights W, i.e., $\mathscr{L} = \mathscr{L}(\mathscr{F}, W)$
II. The loss $\mathscr{L}$ is a differentiable function of $\mathscr{F}$ and W if $\ell_{\mathscr{F}}(\mathscr{F}; W)$ and $\ell_V(V_o(t), \mathscr{F}; W)$ are differentiable functions of all their arguments $(V_o(t), \mathscr{F}; W)$.
III. The loss $\mathscr{L}$ has well-defined gradients w.r.t. the weights W if the spike **firing times** $\mathscr{F}$ are differentiable w.r.t. the weights W.

Implicit Relationship

- The output of a neuron i, i.e., a spike generate at time t, is describe by an implicit function:

$$V_i(f) = \sum_{j \in \mathcal{N}_i} W_{ji} y_{ji}(f) - \theta_i \sum_{m<f} h_i^n(f-m) - \theta_i = 0$$

- We can write the equations for all the firing times as:

$$\mathbb{V}(\mathcal{F}, W) = \mathbf{0}$$

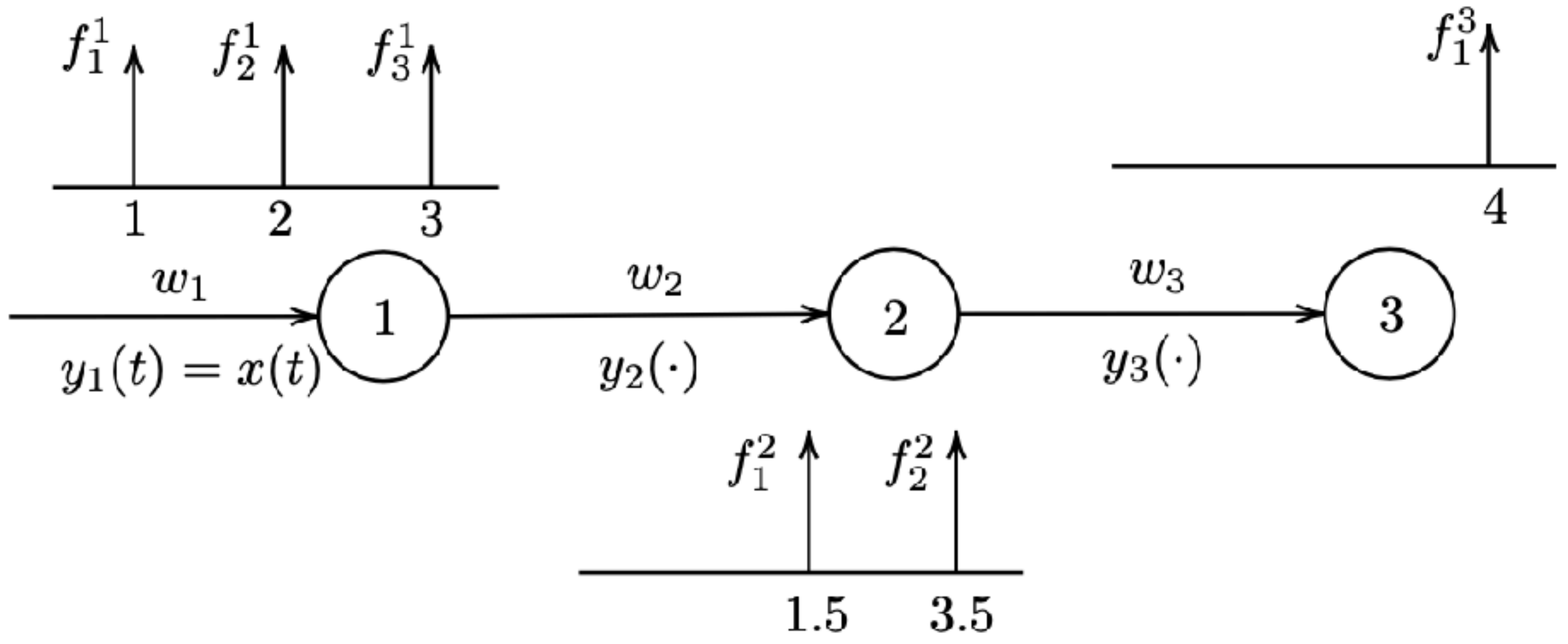
[**Lee, Haghighatshoar**, Karbasi]

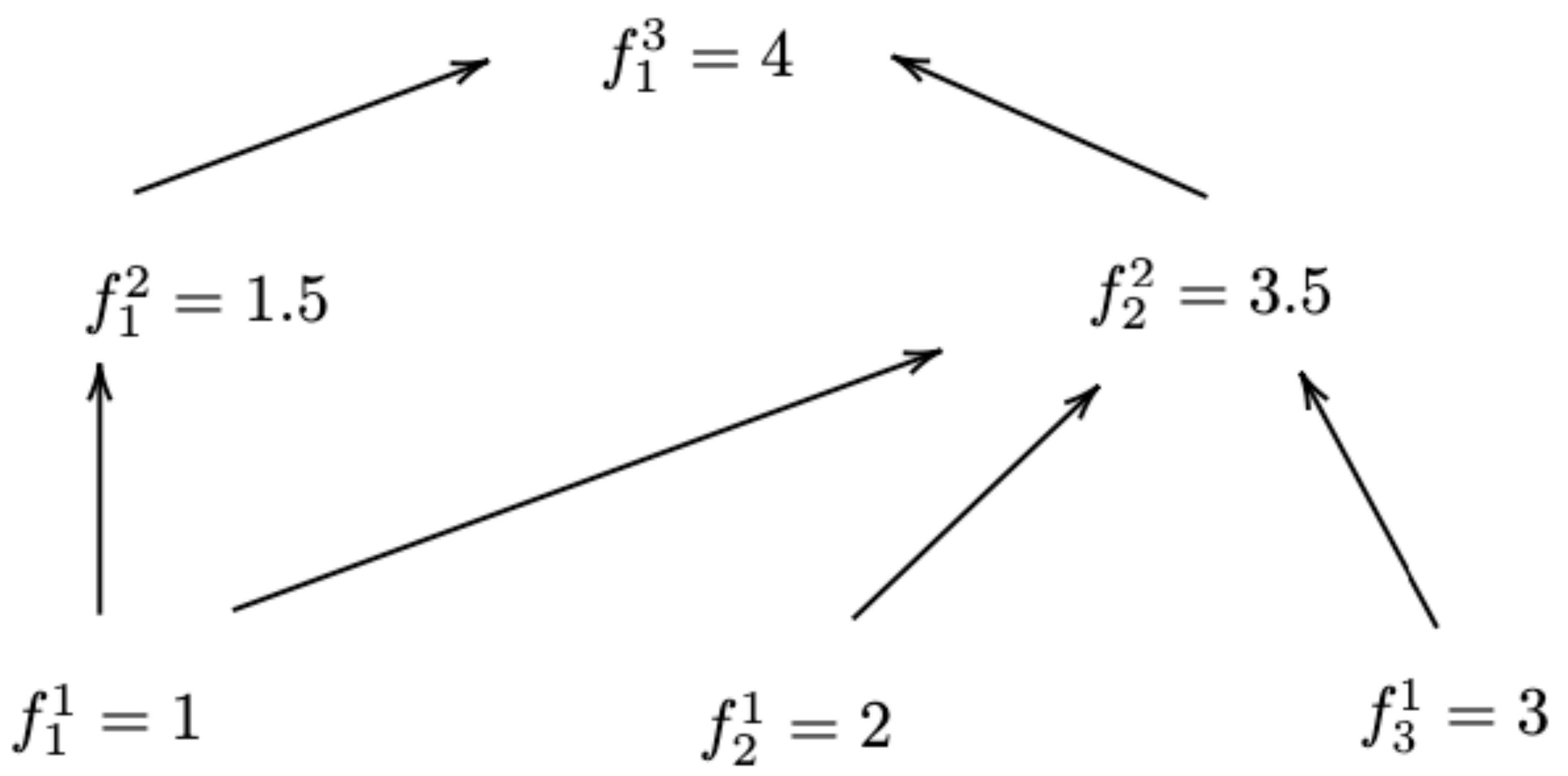
Theorem:

Let $\mathbf{P}$ be a permutation matrix sorting the firing times in $\mathcal{F}$ in an ascending order. Then,

1. $\dfrac{\partial \mathbb{V}}{\partial \mathcal{F}} = \mathbf{P}^T \mathbf{L} \mathbf{P}$ where $\mathbf{L}$ is a lower triangular matrix,
2. $\mathbf{L}$ has strictly positive diagonal elements $\mathbf{L}_{kk} > 0$.

Causal Structure





- The corresponding equation for firing time f can only have contributions from firing times $g < f$
  - By sorting the firing time equations in ascending time order, this results in a lower triangular structure for partial derivatives w.r.t. firing times.
- The potential $V_i(t)$ when it fires at $t = f$ should have a positive derivative.

$$
\begin{array}{l}
\\
V_1(f_1^1) - \theta = 0 \\
V_1(f_2^1) - \theta = 0 \\
V_1(f_3^1) - \theta = 0 \\
V_2(f_1^2) - \theta = 0 \\
V_2(f_2^2) - \theta = 0 \\
V_3(f_1^3) - \theta = 0
\end{array}
\begin{array}{c}
\begin{array}{cccccc} f_1^1 & f_2^1 & f_3^1 & f_1^2 & f_2^2 & f_1^3 \end{array} \\
\begin{pmatrix}
x & & & & & \\
x & x & & & & \\
x & x & x & & & \\
x & & & x & & \\
x & x & x & x & x & \\
 & & & x & x & x
\end{pmatrix}
\end{array}
$$

Implicit Function

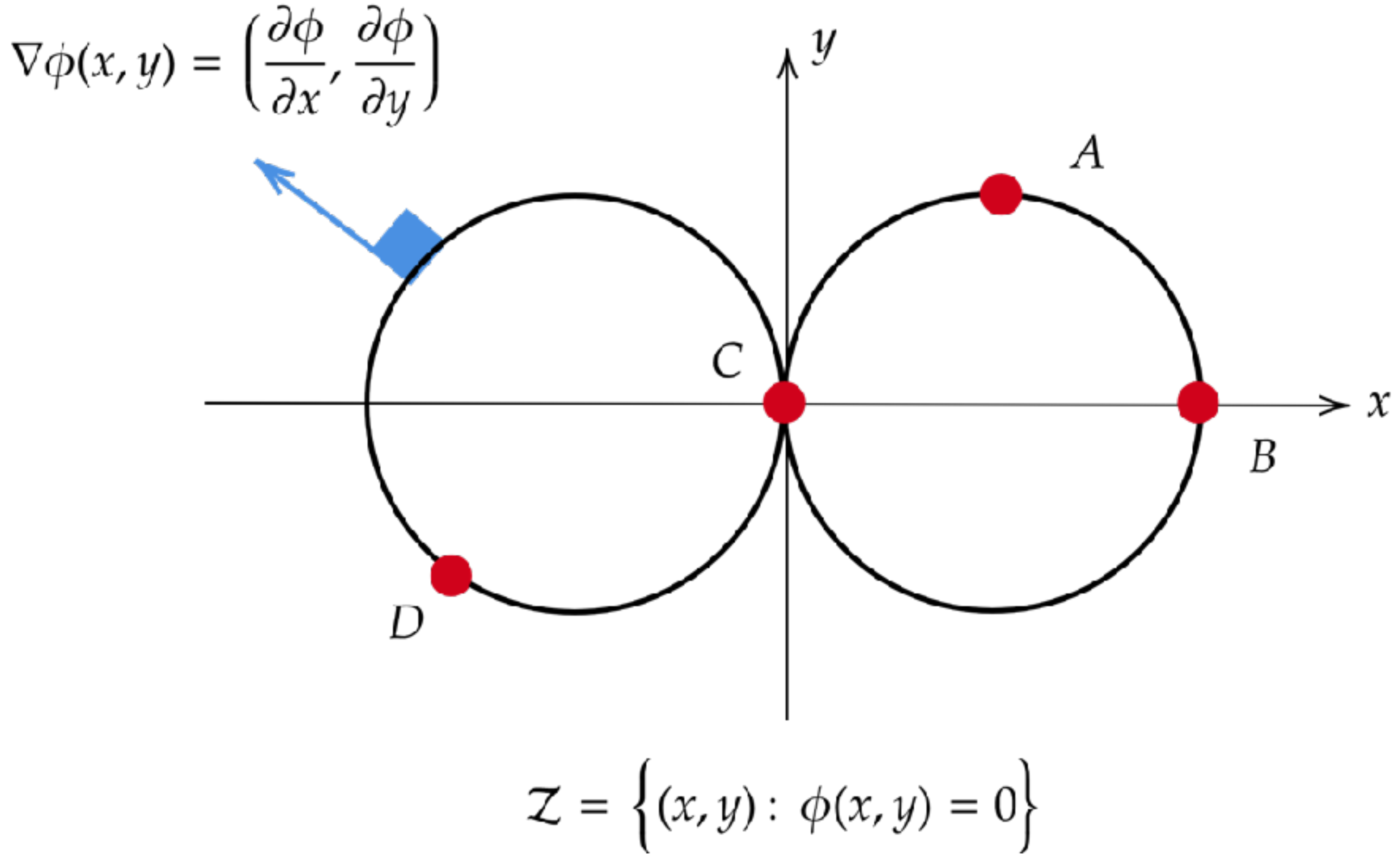
- Can we express $\mathscr{F}$ in terms of W?

$$\mathbb{V}(\mathscr{F}, W) = \mathbf{0}$$

- Not always

$$\ln(|f|) + f^3 w + 20w^2 - w = 0$$

Implicit Function Theorem



Theorem (IFT):

Let $\phi : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ be a differentiable function and let $\phi(x_0, y_0) = 0$. Assume that $\det\left(\frac{\partial\phi}{\partial y}(x_0, y_0)\right) \neq 0$.

1. There is a function ψ such that $y = \psi(x)$ in an open neighborhood around (x_0, y_0).
2. ψ is a differentiable function of x:
$$\frac{\partial\psi}{\partial x} = -\left(\frac{\partial\phi}{\partial y}\right)^{-1} \times \frac{\partial\phi}{\partial x}.$$

Implicit Function Theorem

- We can write the equations for all the firing times as:

$$\mathbb{V}(\mathscr{F}, W) = \mathbf{0}$$

[**Lee, Haghighatshoar**, Karbasi]

Theorem:

Let $\mathbb{V}(\mathscr{F}, W) = \mathbf{0}$ be the set of equations corresponding to the firing times.

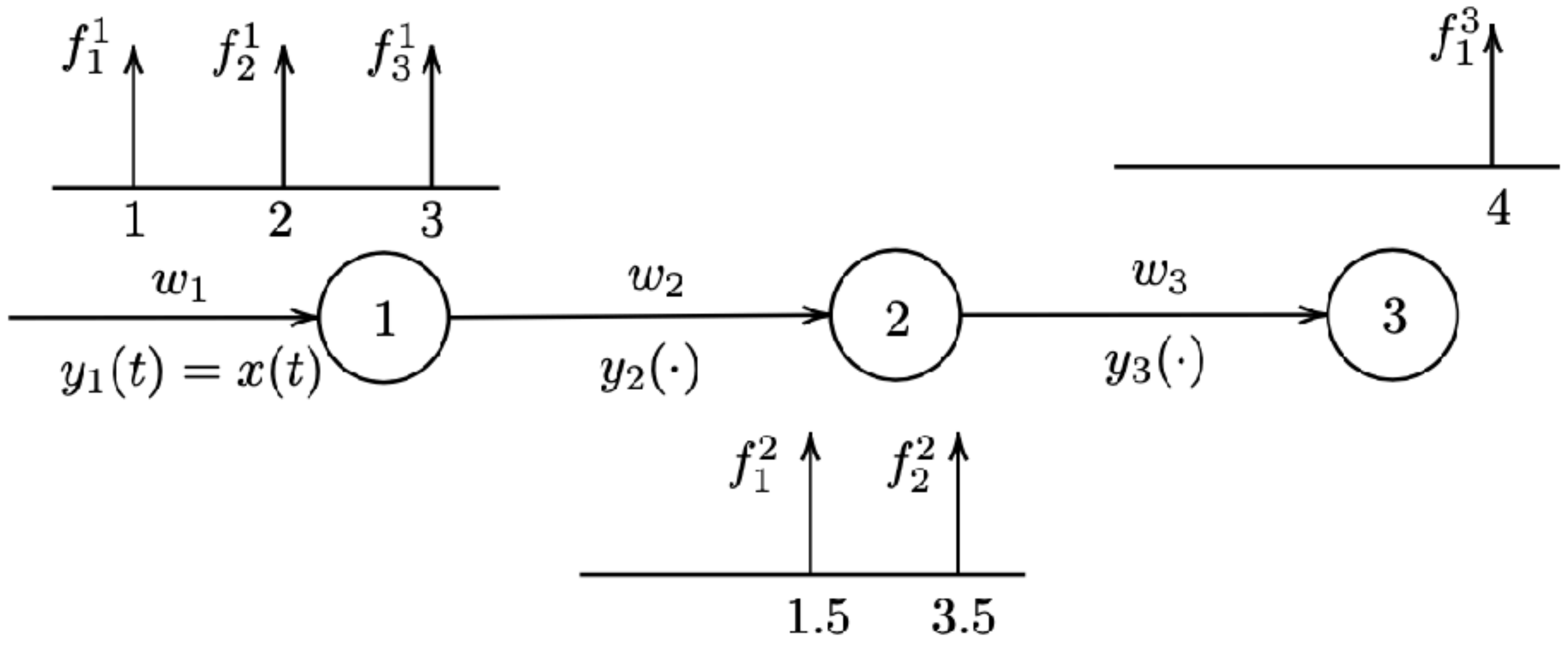
1. Then the Jacobian matrix $\dfrac{\partial \mathbb{V}}{\partial \mathscr{F}}$ is non-singular.
2. IFT implies that the firing times $\mathscr{F}$ can be written as a differentiable function of the weights W.

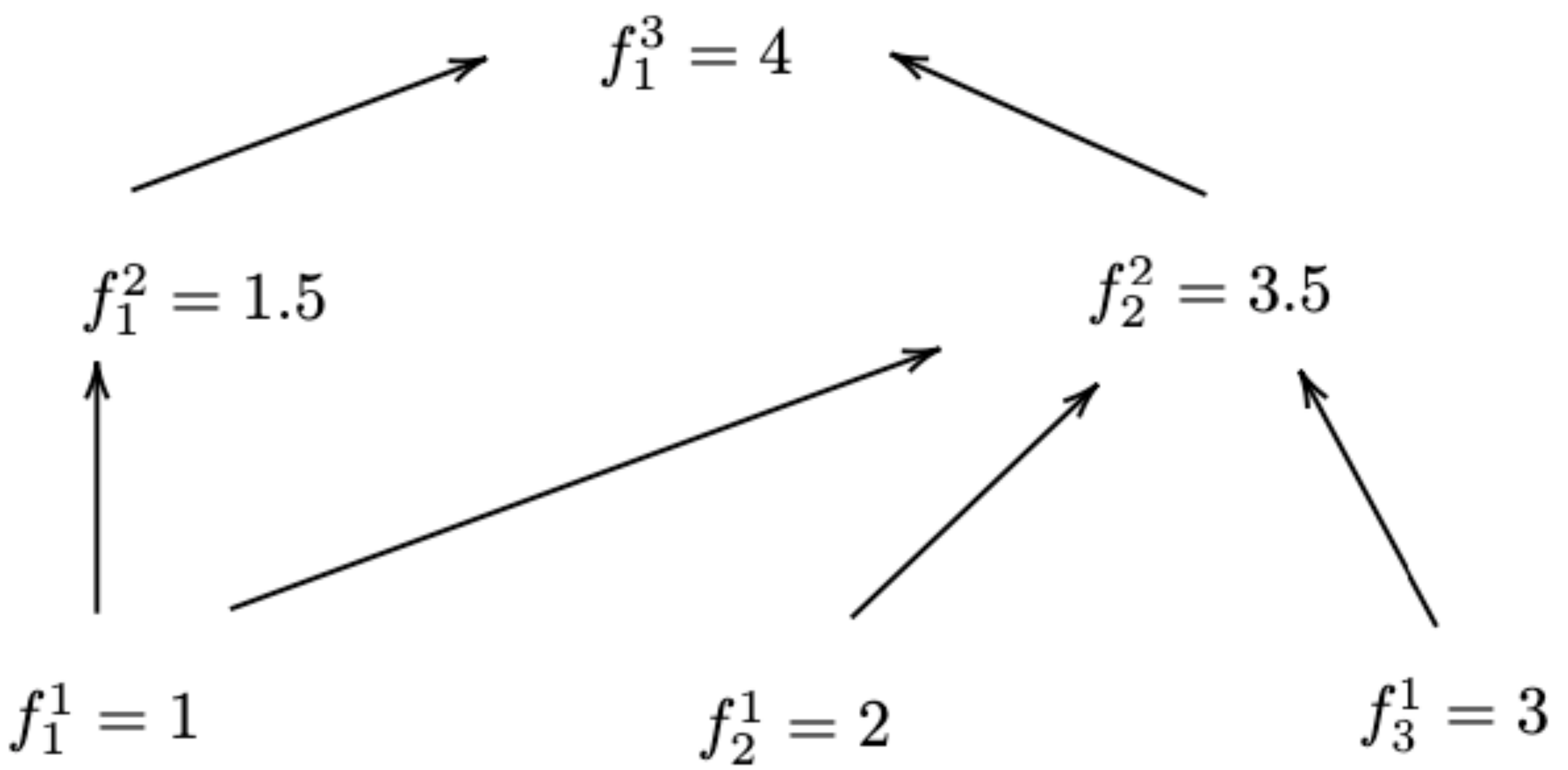
$$\mathbf{L}\frac{\partial \mathscr{F}}{\partial W} = -\frac{\partial \mathbb{V}}{\partial W}$$

Yale

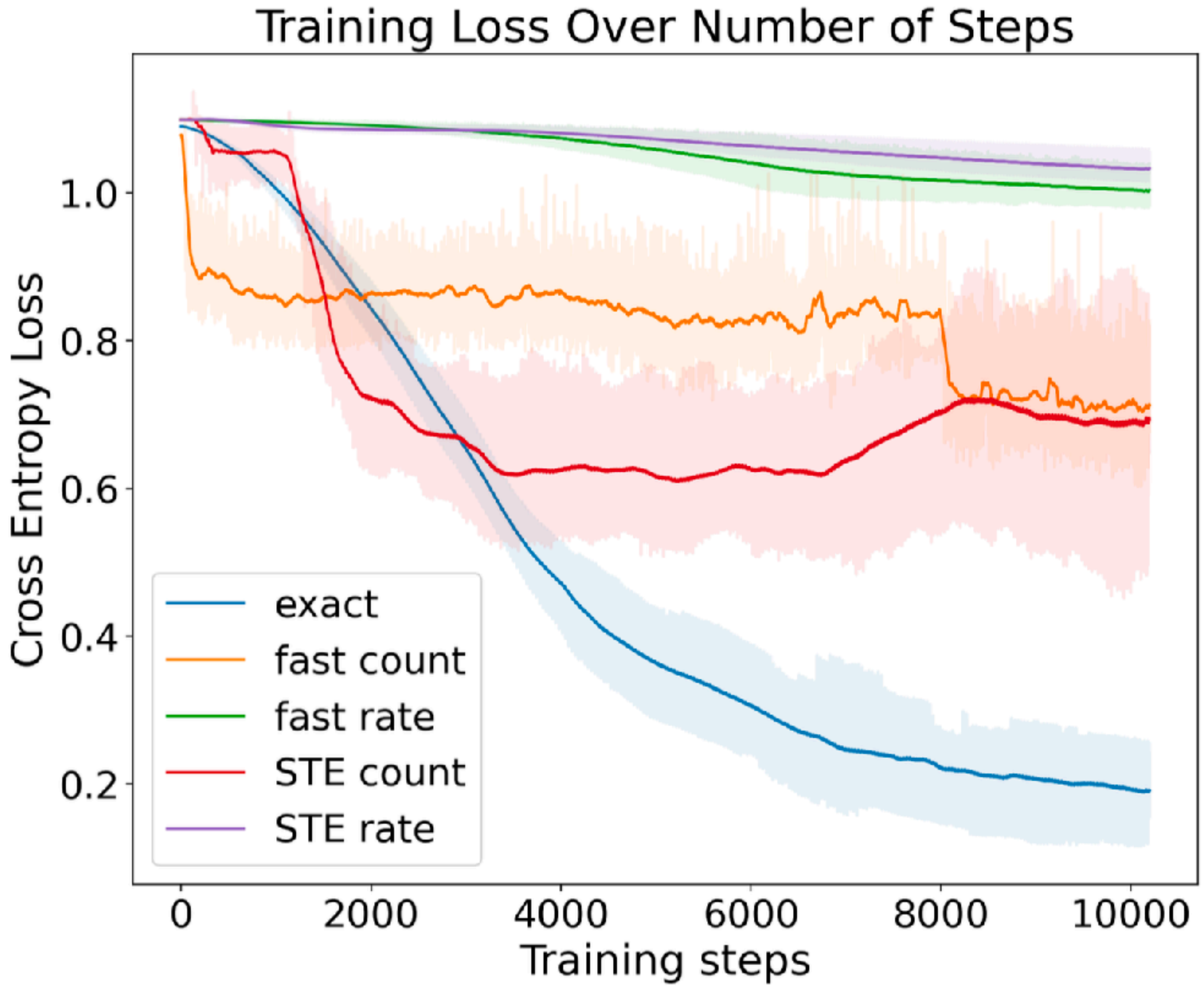
Forward Propagation

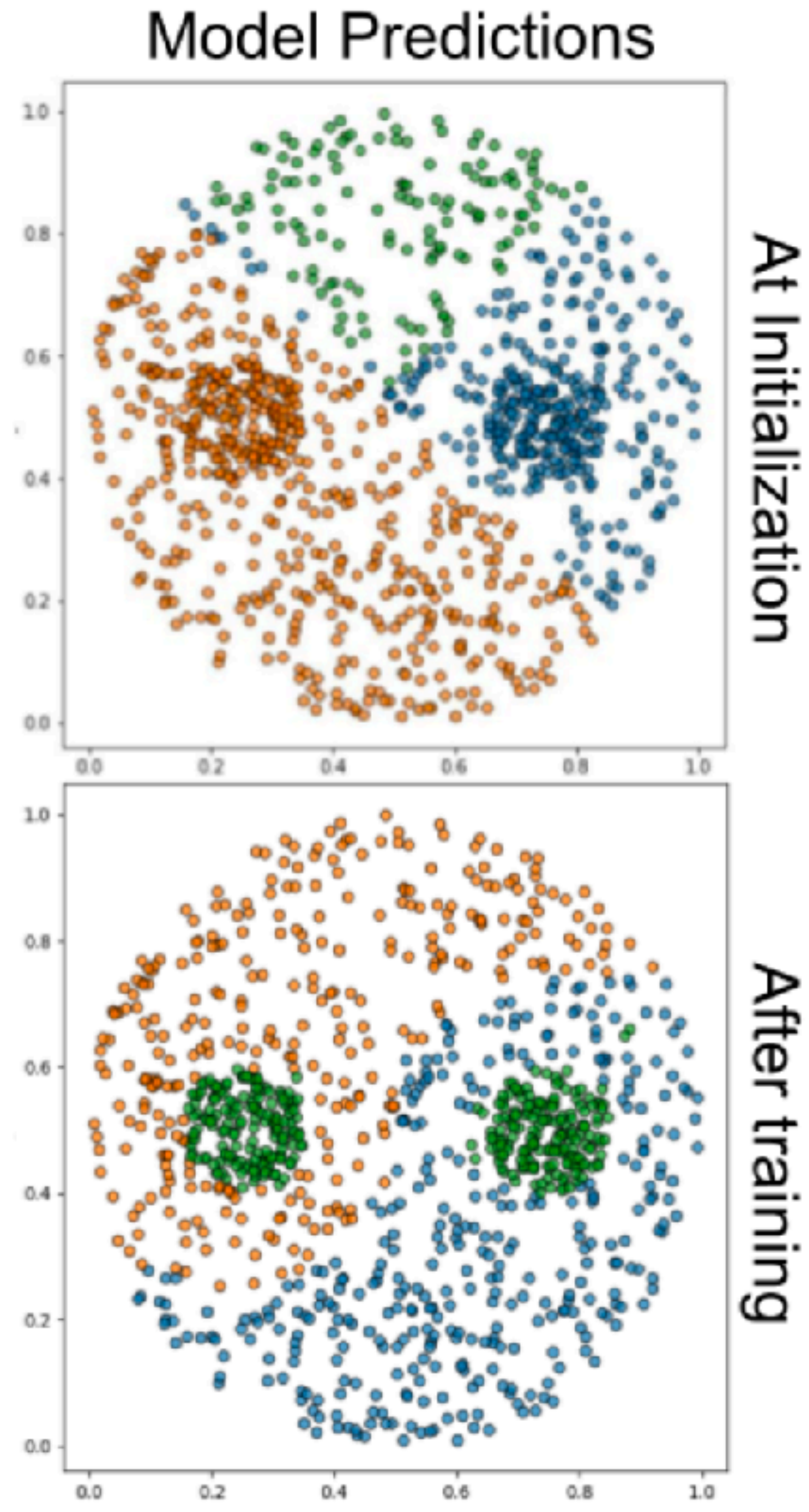
- Use the causal graph to describe $V_i(f) - \theta_i = 0$
- Calculate $\dfrac{\partial}{\partial f_{j\to i}}(V_i(f) - \theta_i)$ for each $f_{j\to i}$ in the casual graph
- Calculate $\dfrac{\partial}{\partial f}(V_i(f) - \theta_i)$
- Calculate $\dfrac{\partial}{\partial W_{ji}}(V_i(f) - \theta_i)$ for all neurons W_{ji} attached to neuron i.
- Solve $\mathbf{L}\dfrac{\partial \mathscr{F}}{\partial W} = -\dfrac{\partial \mathbb{V}}{\partial W}$ by back substitution
- Calculate $\dfrac{\partial \mathscr{L}}{\partial W} = \dfrac{\partial \mathbb{L}}{\partial \mathscr{F}} \times \dfrac{\partial \mathscr{F}}{\partial W} + \dfrac{\partial \mathscr{L}}{\partial W}$





Yin-Yang


Training Loss Over Number of Steps
Cross Entropy Loss
0.2
0.4
0.6
0.8
1.0
0
2000
4000
6000
8000
10000
Training steps
exact
fast count
fast rate
STE count
STE rate


Model Predictions
At Initialization
After training

Conclusion

- How do we train SNN?
  - Spikes are not differentiable functions!!!
  - SNNs are differentiable in the parameter space W
  - We can use **forward propagation** to compute the gradients

Implicit Function Theorem

Exact Gradient Computation for Spiking Neural Networks via Forward Propagation

Jane H. Lee
Yale University

Saeid Haghighatshoar
SynSense

Amin Karbasi
Yale University

Adjoint State Method

Event-based backpropagation can compute exact gradients for spiking neural networks

Timo C. Wunderlich[1,2,3] & Christian Pehle[1,3]

Thank You !