On the Algorithmic Power of Spiking Neural Networks

Chi-Ning Chou
Harvard University

Kai-Min Chung
Academia Sinica

Chi-Jen Lu
Academia Sinica

ITCS 2019

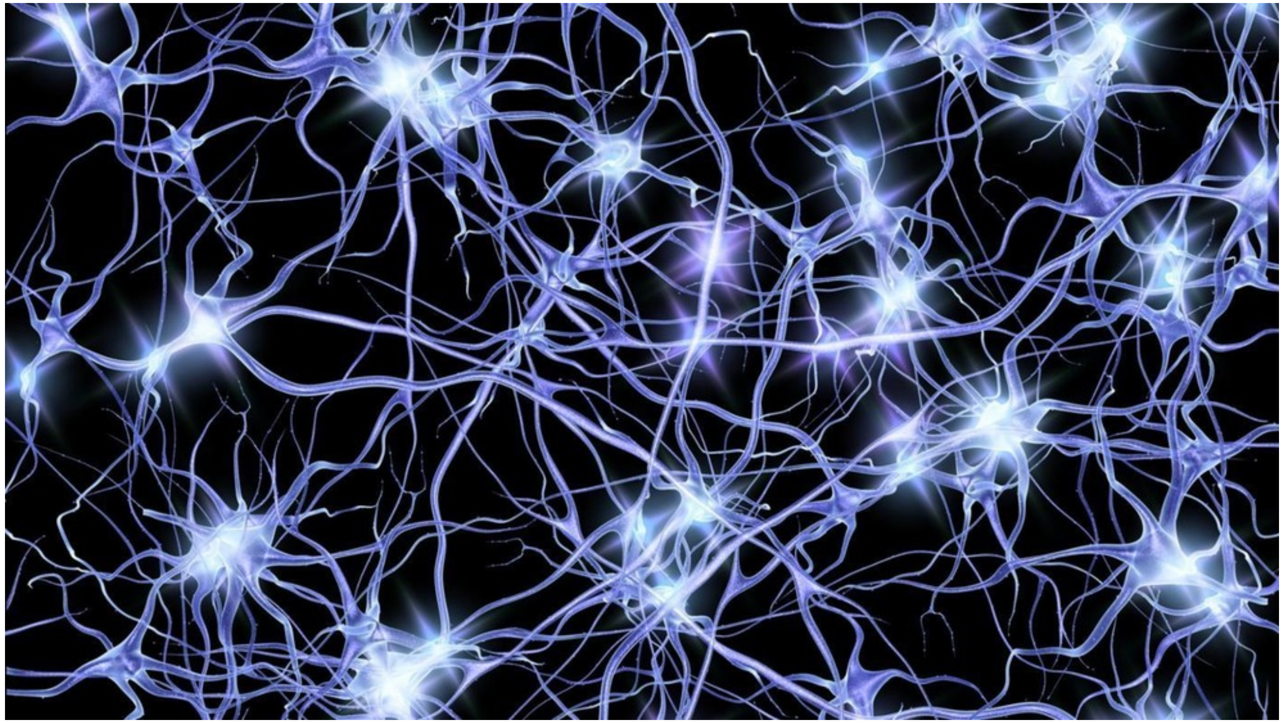
What is Spiking Neural Networks (SNNs)?

What is Spiking Neural Networks (SNNs)?

- Mathematical models for "biological neural networks".

What is Spiking Neural Networks (SNNs)?

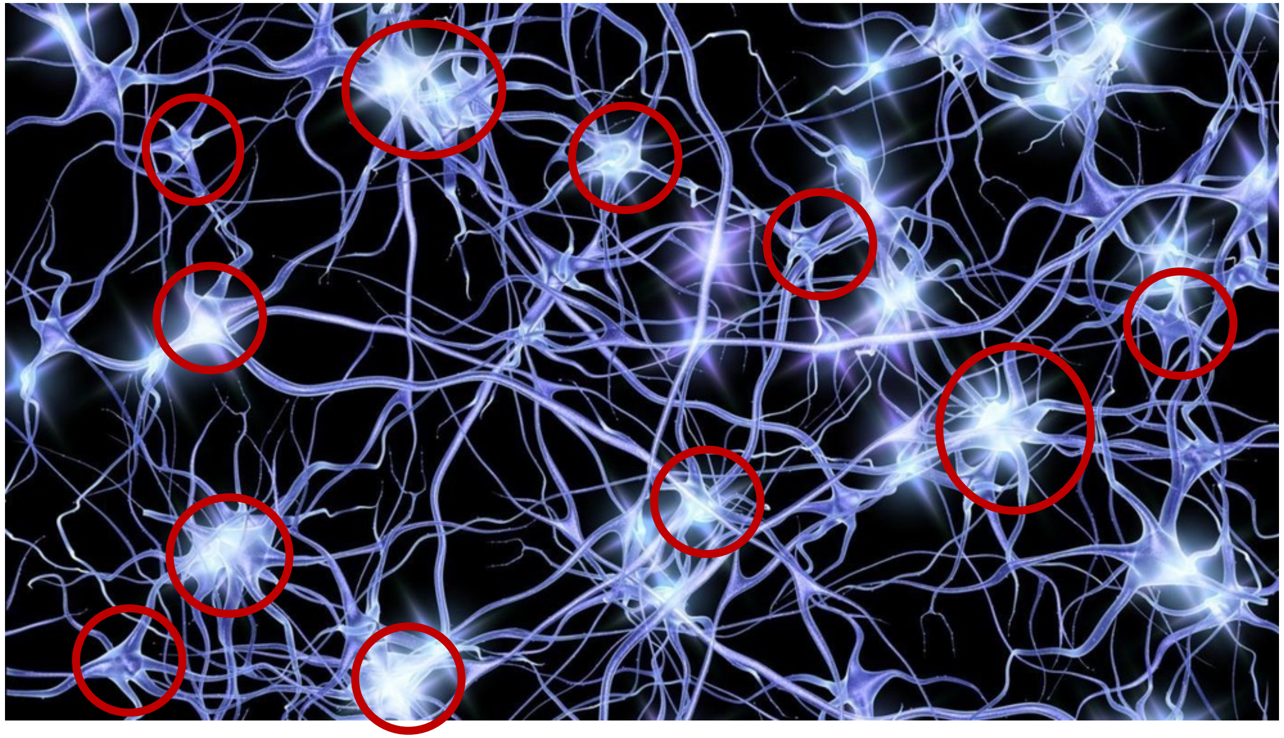
- Mathematical models for "biological neural networks".



* By Pennstatenews https://www.flickr.com/photos/pennstatelive/37247502805

What is Spiking Neural Networks (SNNs)?

- Mathematical models for "biological neural networks".

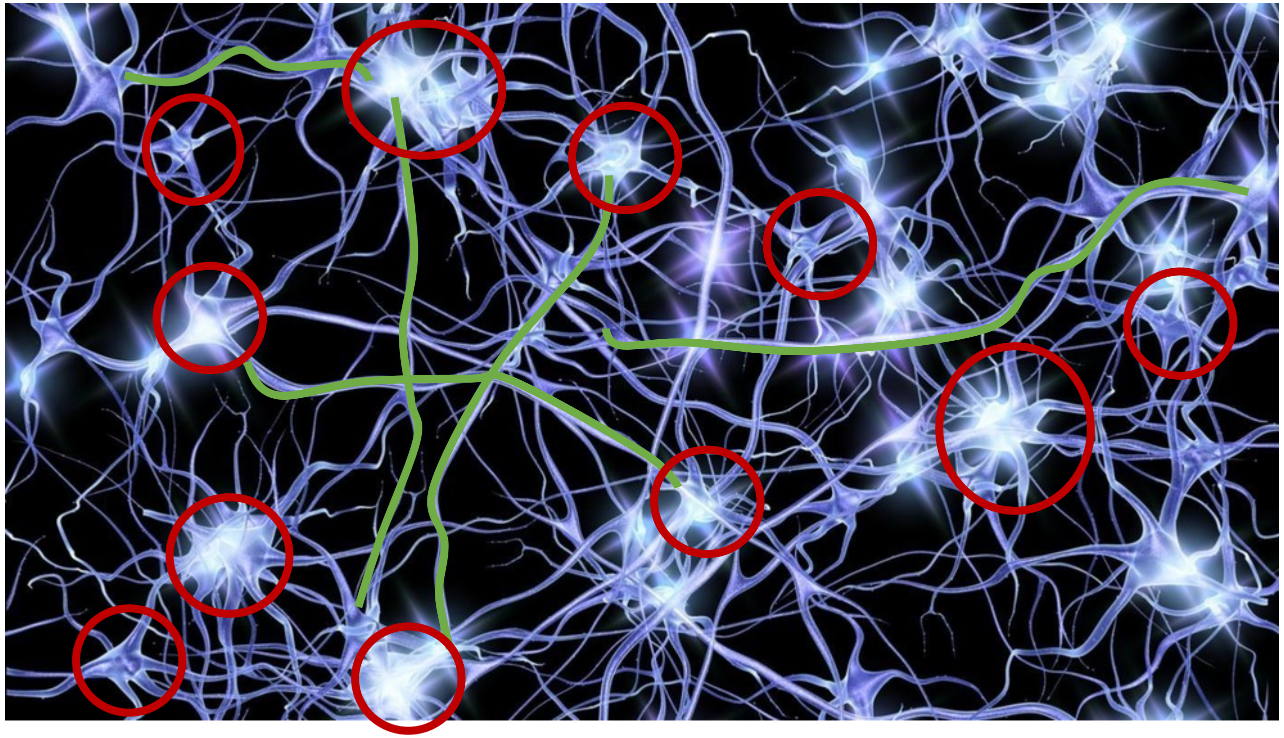


* By Pennstatenews https://www.flickr.com/photos/pennstatelive/37247502805

Neurons: Nerve cells

What is Spiking Neural Networks (SNNs)?

- Mathematical models for “biological neural networks”.



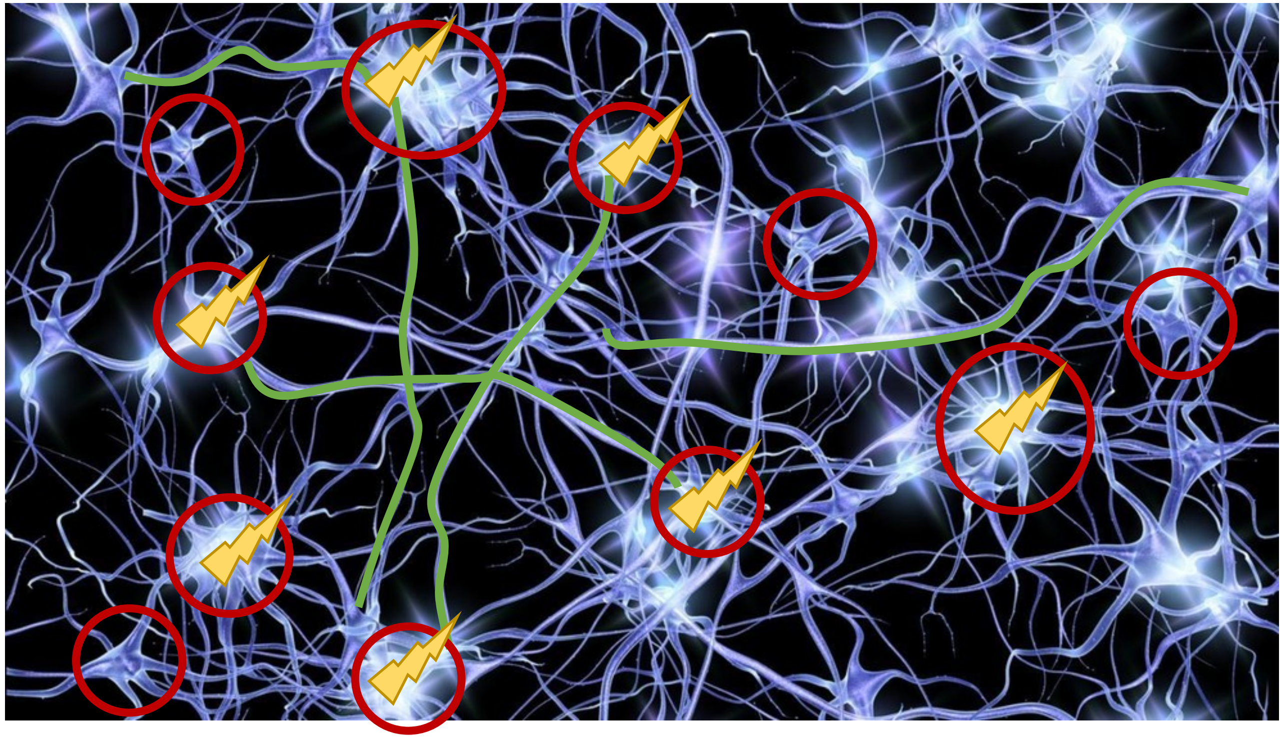
* By Pennstatenews https://www.flickr.com/photos/pennstatelive/37247502805

Neurons: Nerve cells

Synapses: Connections between neurons

What is Spiking Neural Networks (SNNs)?

- Mathematical models for "biological neural networks".



* By Pennstatenews https://www.flickr.com/photos/pennstatelive/37247502805

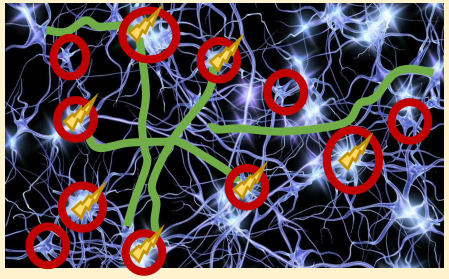
Neurons: Nerve cells

Synapses: Connections between neurons

Spikes: Instantaneous signals

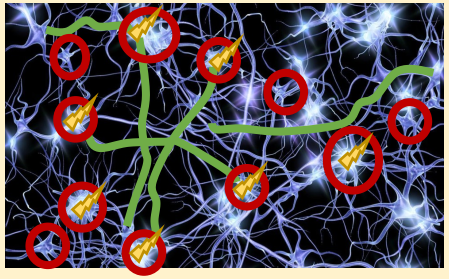
What is Spiking Neural Networks (SNNs)?

- Mathematical models for "biological neural networks".



Neurons
Synapses
Spikes

What is Spiking Neural Networks (SNNs)?



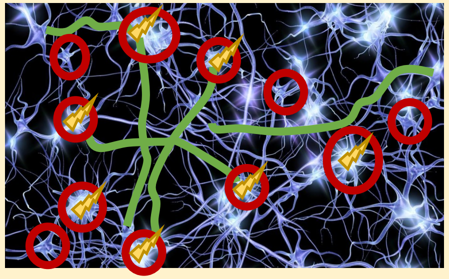
Neurons

Synapses

Spikes

- Mathematical models for “biological neural networks”.
- Various models since the 1900s.
  - Integrate-and-fire [Lap07], Hodgkin-Huxley [HH52], their variants [Fit61, Ste65, ML81, HR84, Ger95, KGH97, BL03, FTHVVB03, I+03, TMS14].

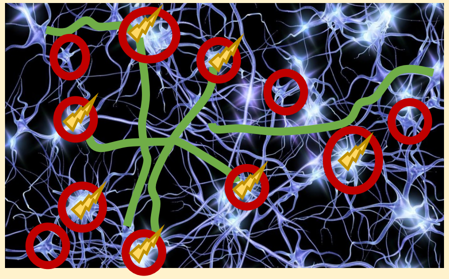
What is Spiking Neural Networks (SNNs)?



Neurons
Synapses
Spikes

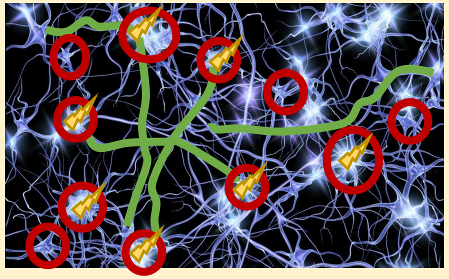
- Mathematical models for "biological neural networks".
- Various models since the 1900s.
  - Integrate-and-fire [Lap07], Hodgkin-Huxley [HH52],
    their variants [Fit61, Ste65, ML81, HR84, Ger95, KGH97, BL03, FTHVVB03, I+03, TMS14].
- Study the behaviors/statistics of SNNs, e.g., firing rate.

What is Spiking Neural Networks (SNNs)?



- Mathematical models for “biological neural networks”.
- Various models since the 1900s.
    - Integrate-and-fire [Lap07], Hodgkin-Huxley [HH52], their variants [Fit61, Ste65, ML81, HR84, Ger95, KGH97, BL03, FTHVVB03, I+03, TMS14].
- Study the behaviors/statistics of SNNs, e.g., firing rate.
- [Barret-Denève-Machens 2013] empirically showed a connection between the firing rate of integrate-and-fire SNNs and an optimization problem.

What is Spiking Neural Networks (SNNs)?



Neurons

Synapses

Spikes

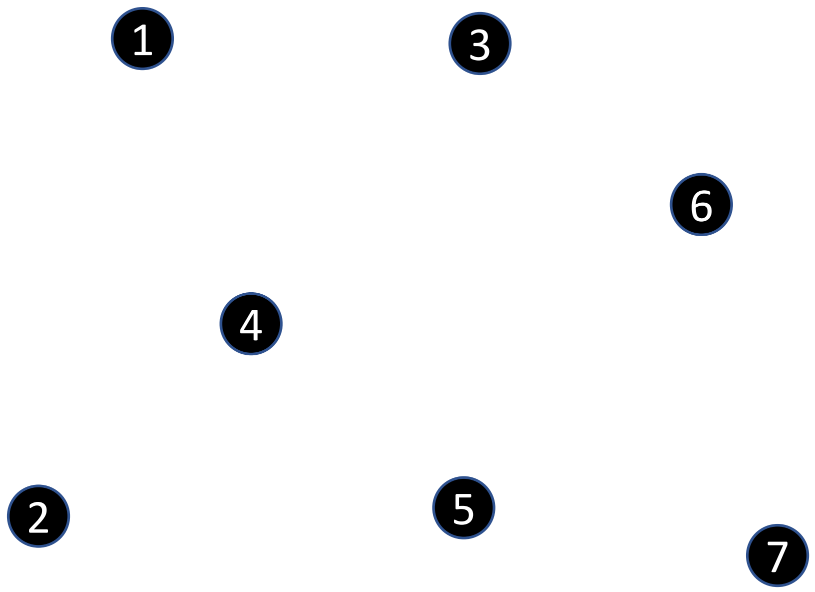
- Mathematical models for "biological neural networks".
- Various models since the 1900s.
  - Integrate-and-fire [Lap07], Hodgkin-Huxley [HH52], their variants [Fit61, Ste65, ML81, HR84, Ger95, KGH97, BL03, FTHVVB03, I+03, TMS14].
- Study the behaviors/statistics of SNNs, e.g., firing rate.
- [Barret-Denève-Machens 2013] empirically showed a connection between the firing rate of integrate-and-fire SNNs and an optimization problem.

SNNs seem to have non-trivial computational power. Can we understand them better through the lens of algorithms?

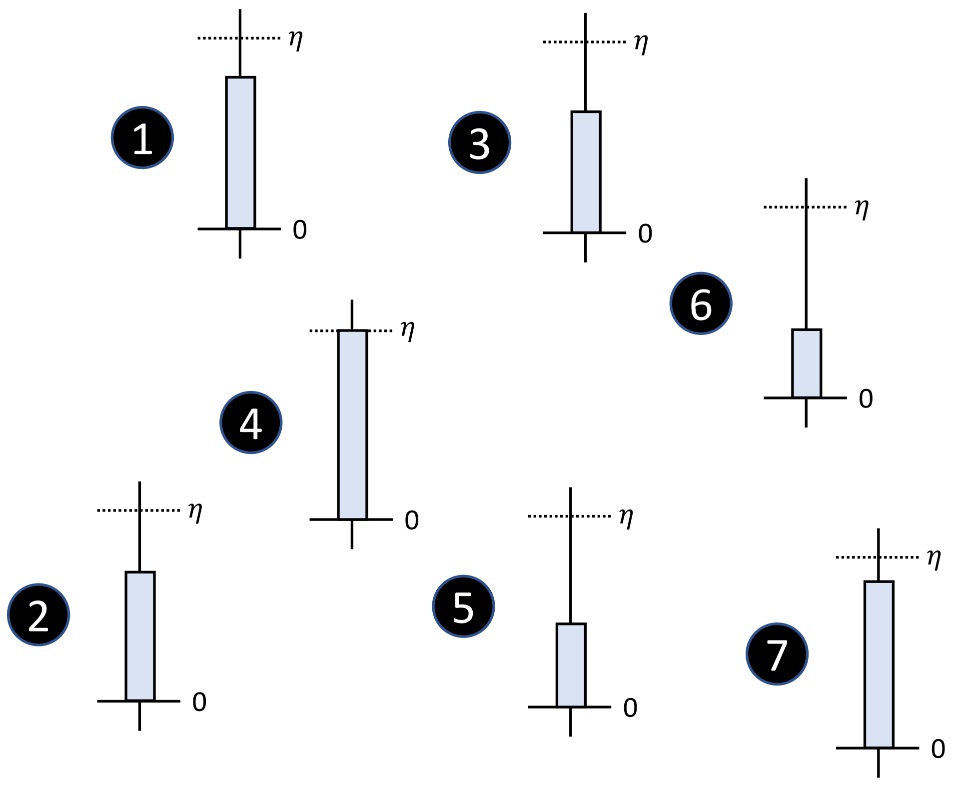
Integrate-and-Fire (IAF) Model [Lapicque 1907]

Integrate-and-Fire (IAF) Model [Lapicque 1907]

- Neurons: $[n] = \{1,2,\ldots,n\}$

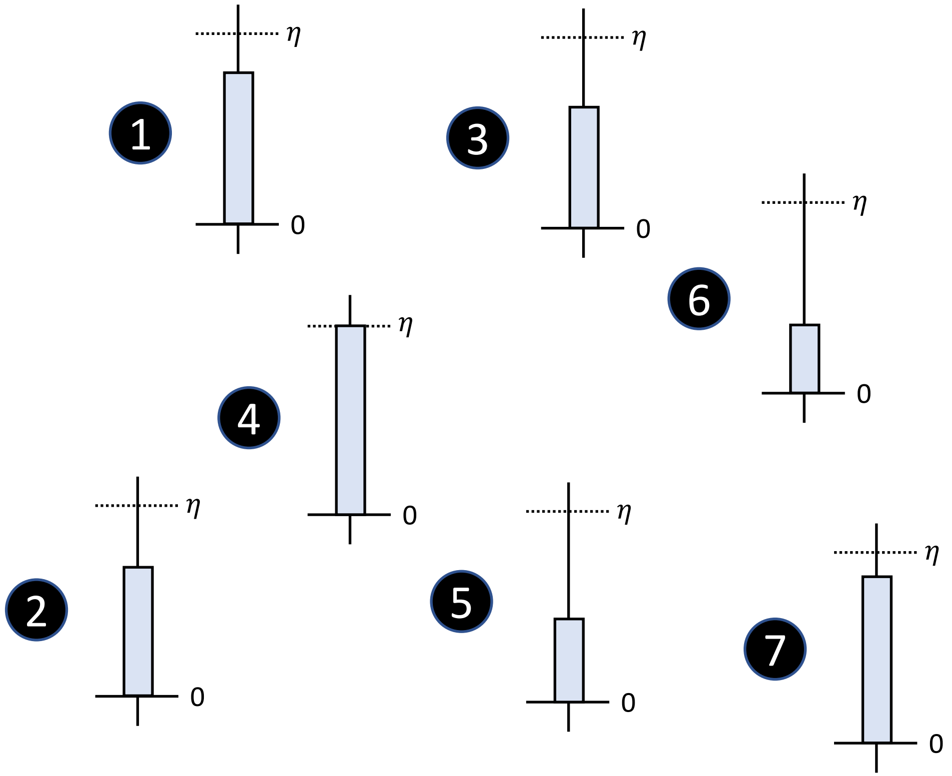


Integrate-and-Fire (IAF) Model [Lapicque 1907]



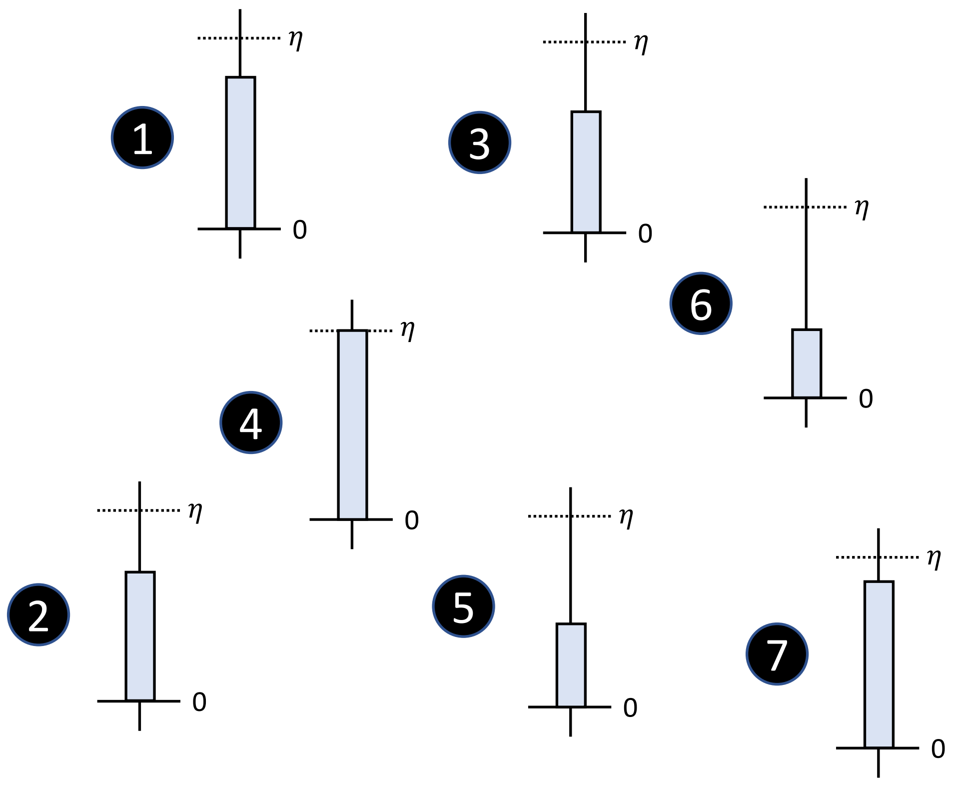
- Neurons: $[n] = \{1, 2, \ldots, n\}$
- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$

Integrate-and-Fire (IAF) Model [Lapicque 1907]



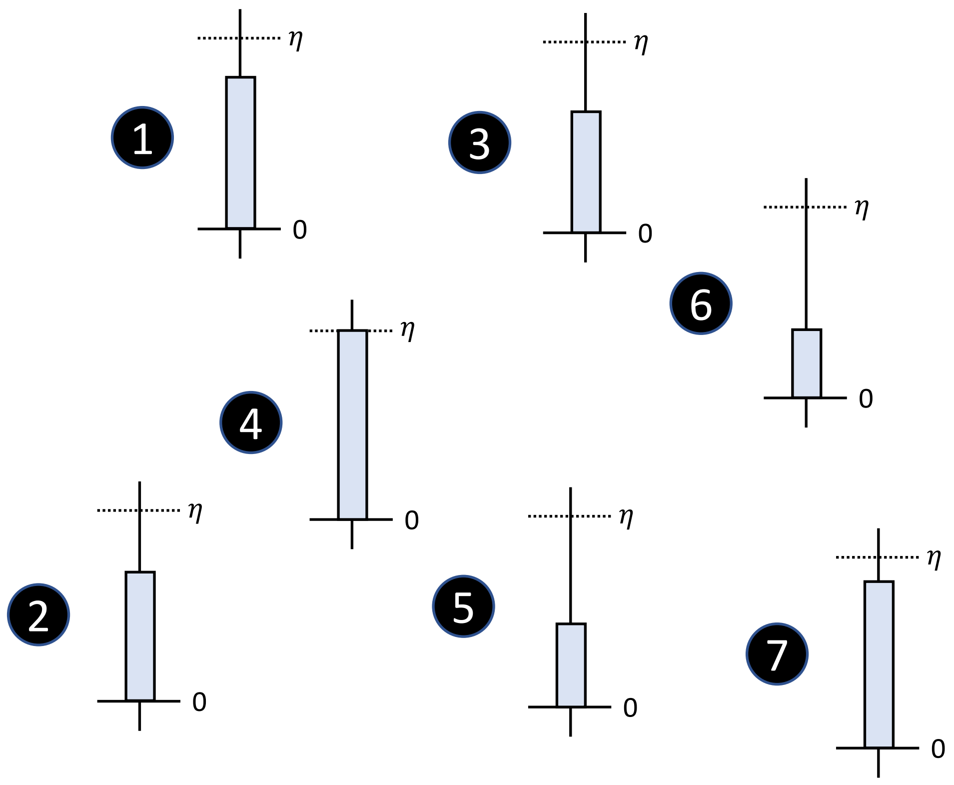
- Neurons: $[n] = \{1,2,\ldots,n\}$
- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- Dynamics: $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

Integrate-and-Fire (IAF) Model [Lapicque 1907]



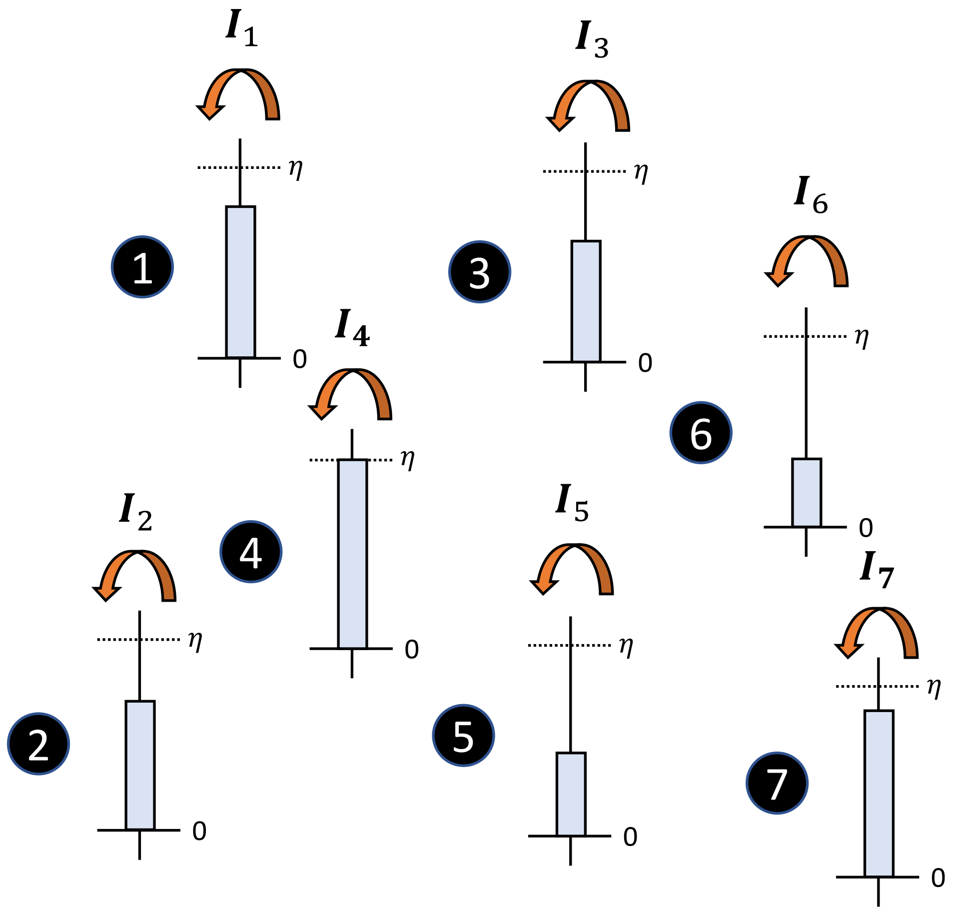
- Neurons: $[n] = \{1,2,\dots,n\}$
- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- Dynamics: $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \underbrace{\boldsymbol{I}}_{\text{External charging}}$

Integrate-and-Fire (IAF) Model [Lapicque 1907]



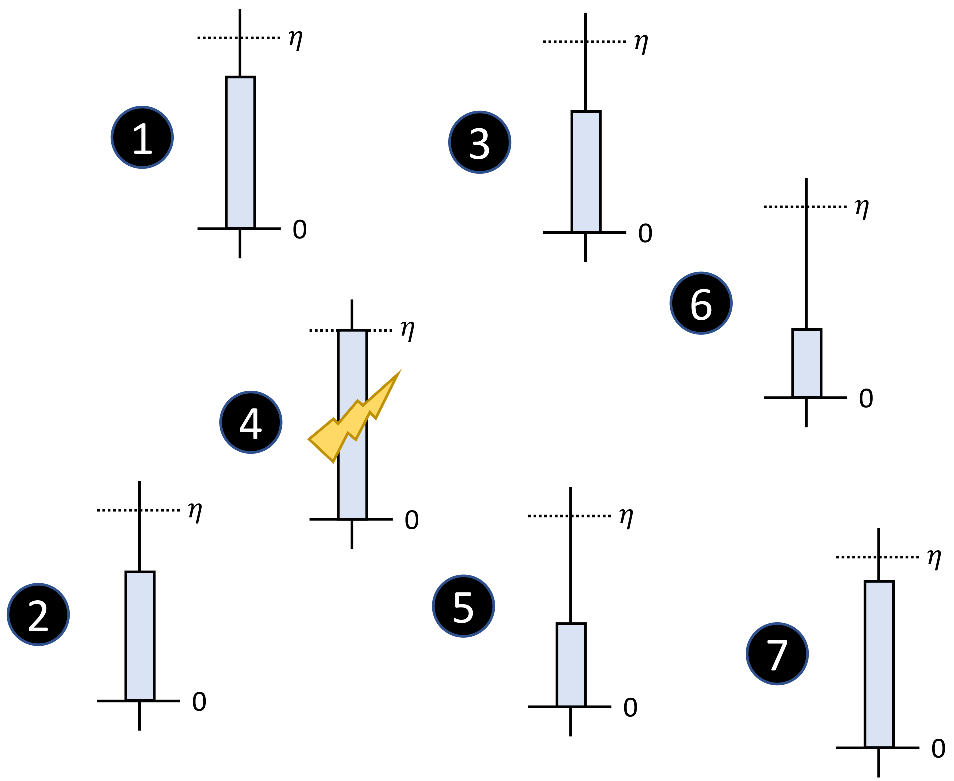
- Neurons: $[n] = \{1,2,\ldots,n\}$
- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- Dynamics: $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) \underbrace{- C\boldsymbol{s}(t)}_{\text{Spiking effects}} + \boldsymbol{I}$

Integrate-and-Fire (IAF) Model [Lapicque 1907]



- Neurons: $[n] = \{1,2,\ldots,n\}$
- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- Dynamics: $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$

Integrate-and-Fire (IAF) Model [Lapicque 1907]

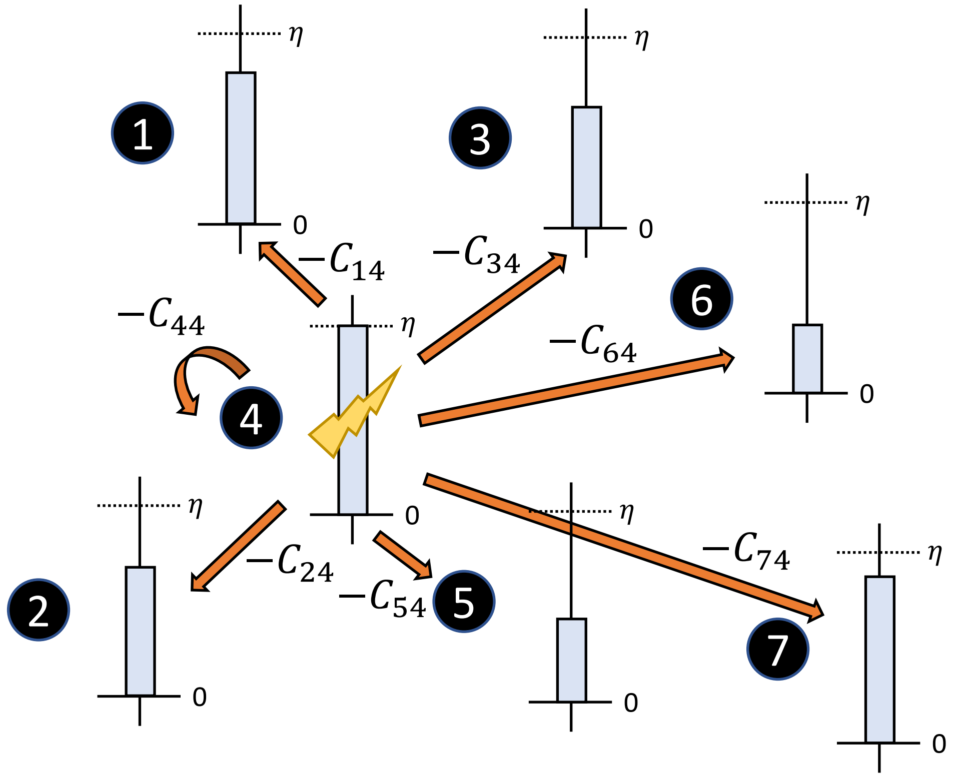


- Neurons: $[n] = \{1,2,\ldots,n\}$
- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- Dynamics: $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$

Firing Rule

$$\boldsymbol{s}_i(t) = 1 \Leftrightarrow \boldsymbol{u}_i(t) \geq \eta$$

Integrate-and-Fire (IAF) Model [Lapicque 1907]

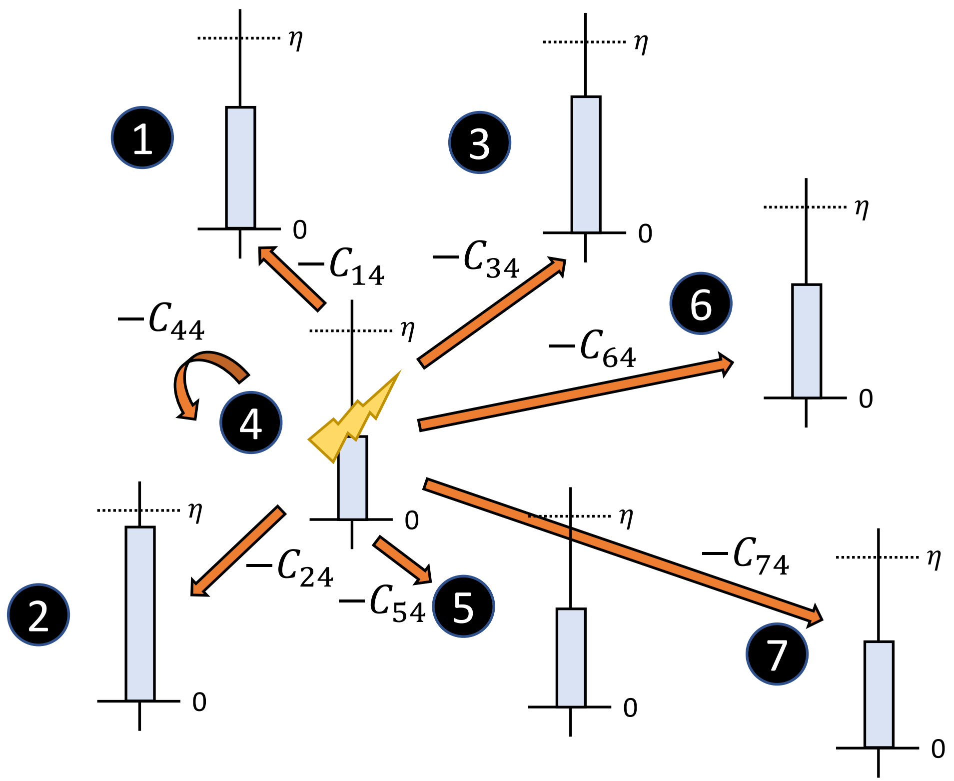


- Neurons: $[n] = \{1,2,\dots,n\}$
- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- Dynamics: $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n\times n}$

Firing Rule

$$\boldsymbol{s}_i(t) = 1 \Leftrightarrow \boldsymbol{u}_i(t) \geq \eta$$

Integrate-and-Fire (IAF) Model [Lapicque 1907]

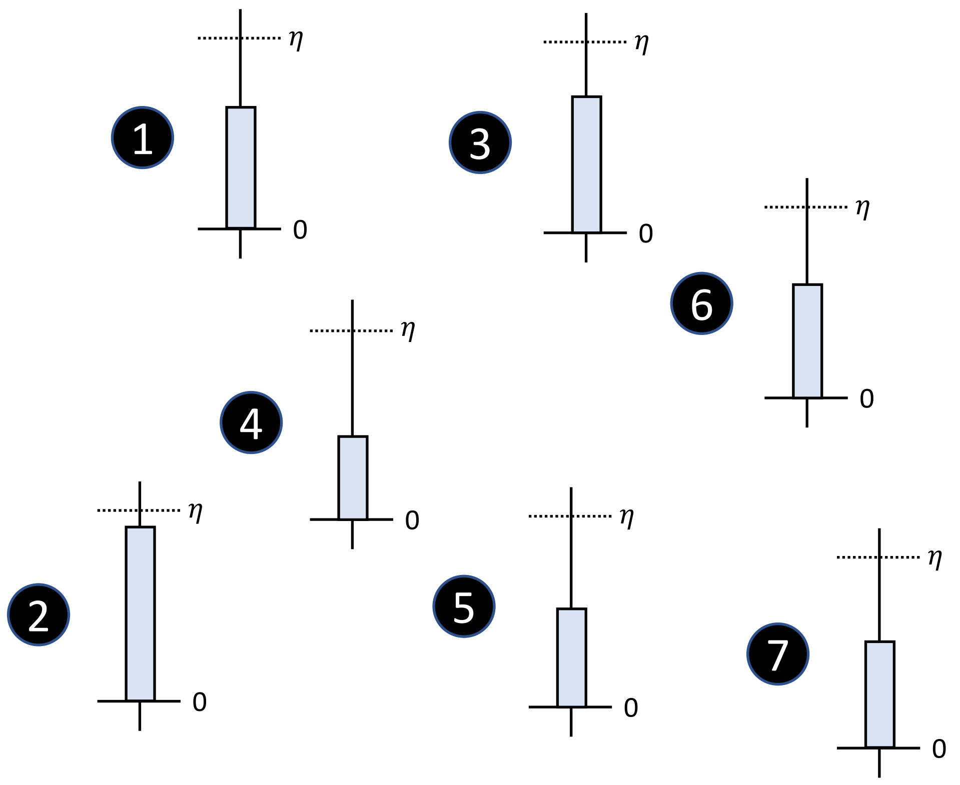


- Neurons: $[n] = \{1,2,\ldots,n\}$
- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- Dynamics: $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n\times n}$

Firing Rule

$\boldsymbol{s}_i(t) = 1 \Leftrightarrow \boldsymbol{u}_i(t) \geq \eta$

Integrate-and-Fire (IAF) Model [Lapicque 1907]

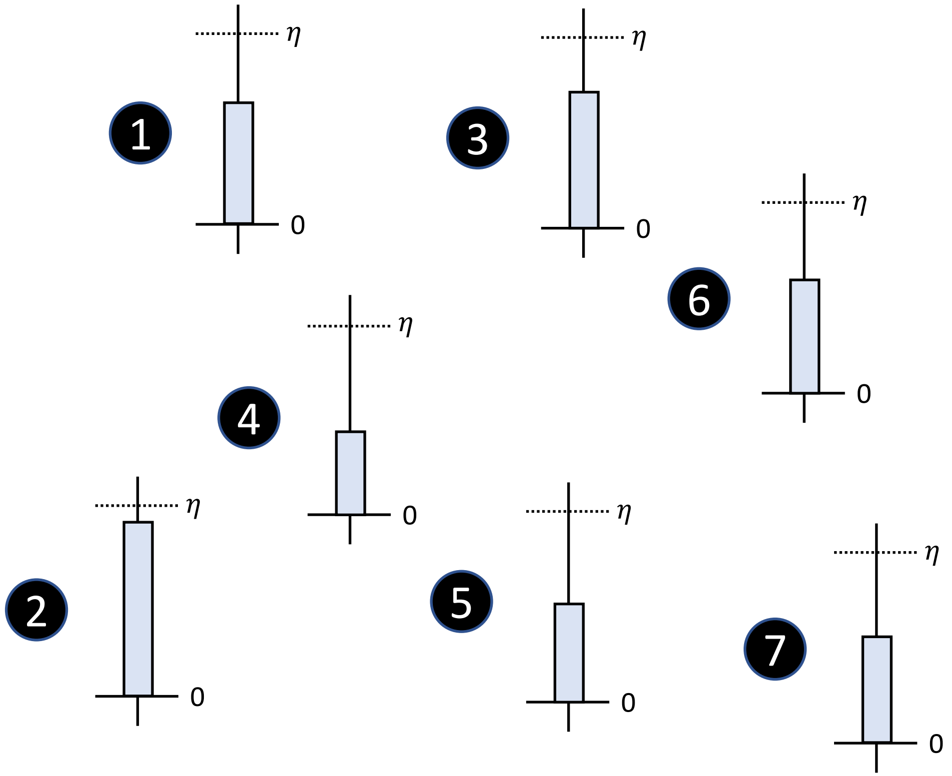


- Neurons: $[n] = \{1,2,\dots,n\}$
- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- Dynamics: $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n\times n}$

Firing Rule

$$\boldsymbol{s}_i(t) = 1 \Leftrightarrow \boldsymbol{u}_i(t) \geq \eta$$

Integrate-and-Fire (IAF) Model [Lapicque 1907]



- Neurons: $[n] = \{1,2,\ldots,n\}$
- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- Dynamics: $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n\times n}$
- Firing rate: $\boldsymbol{x}(t) = (\#\text{spikes before time } t)/t$

Firing Rule

$\boldsymbol{s}_i(t) = 1 \Leftrightarrow \boldsymbol{u}_i(t) \geq \eta$

Example

- Setup: $C = \begin{pmatrix} 1 & 0 \\ -0.5 & 1 \end{pmatrix}$, $\boldsymbol{I} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}$, $\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\eta = 1$.

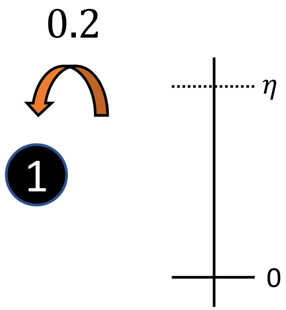
- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n\times n}$
- Firing rate: $\boldsymbol{x}(t)$
- $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

Example

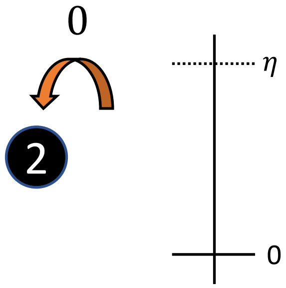
- Setup: $C = \begin{pmatrix} 1 & 0 \\ -0.5 & 1 \end{pmatrix}$, $\boldsymbol{I} = \boxed{\begin{pmatrix} 0.2 \\ 0 \end{pmatrix}}$, $\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \eta = 1.$

- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n \times n}$
- Firing rate: $\boldsymbol{x}(t)$
- $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

$t = 0$



$\#\text{spikes} = 0;\ \boldsymbol{x}_1(t) = 0; \boldsymbol{u}_1(t) = 0$



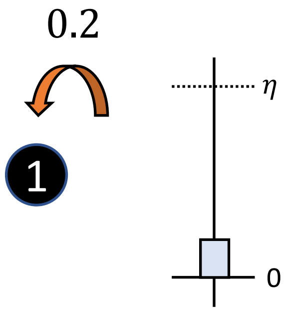
$\#\text{spikes} = 0;\ \boldsymbol{x}_2(t) = 0; \boldsymbol{u}_2(t) = 0$

Example

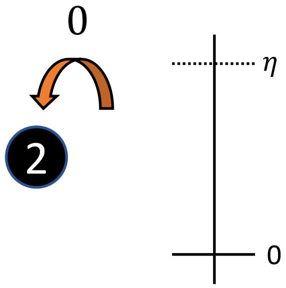
- Setup: $C = \begin{pmatrix} 1 & 0 \\ -0.5 & 1 \end{pmatrix}$, $\boldsymbol{I} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}$, $\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \eta = 1.$

- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n \times n}$
- Firing rate: $\boldsymbol{x}(t)$
- $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

$t = 1$



$$\#\text{spikes} = 0;\ \boldsymbol{x}_1(t) = 0; \boldsymbol{u}_1(t) = 0.2$$



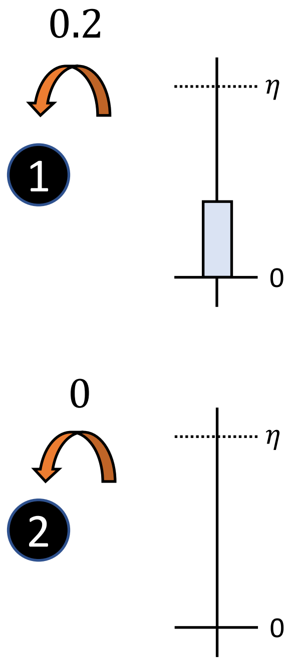
$$\#\text{spikes} = 0;\ \boldsymbol{x}_2(t) = 0; \boldsymbol{u}_2(t) = 0$$

Example

- Setup: $C = \begin{pmatrix} 1 & 0 \\ -0.5 & 1 \end{pmatrix}$, $\boldsymbol{I} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}$, $\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\eta = 1$.

- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n\times n}$
- Firing rate: $\boldsymbol{x}(t)$
- $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

$t = 2$



$\#\text{spikes} = 0;\ \boldsymbol{x}_1(t) = 0; \boldsymbol{u}_1(t) = 0.4$

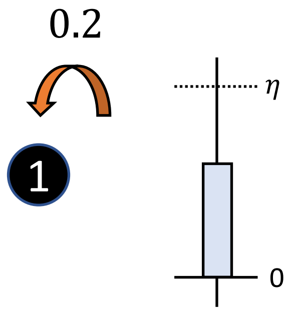
$\#\text{spikes} = 0;\ \boldsymbol{x}_2(t) = 0; \boldsymbol{u}_2(t) = 0$

Example

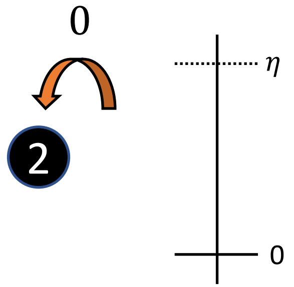
- Setup: $C = \begin{pmatrix} 1 & 0 \\ -0.5 & 1 \end{pmatrix}$, $\boldsymbol{I} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}$, $\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \eta = 1.$

- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n \times n}$
- Firing rate: $\boldsymbol{x}(t)$
- $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

$t = 3$



$\#\text{spikes} = 0;\ \boldsymbol{x}_1(t) = 0; \boldsymbol{u}_1(t) = 0.6$



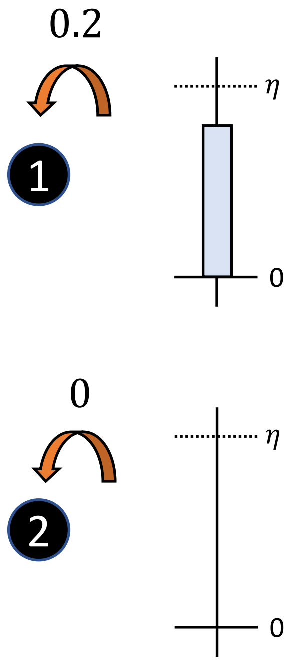
$\#\text{spikes} = 0;\ \boldsymbol{x}_2(t) = 0; \boldsymbol{u}_2(t) = 0$

Example

- Setup: $C = \begin{pmatrix} 1 & 0 \\ -0.5 & 1 \end{pmatrix}$, $\boldsymbol{I} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}$, $\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \eta = 1.$

- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n \times n}$
- Firing rate: $\boldsymbol{x}(t)$
- $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

$t = 4$



$$\#\text{spikes} = 0;\ \boldsymbol{x}_1(t) = 0; \boldsymbol{u}_1(t) = 0.8$$

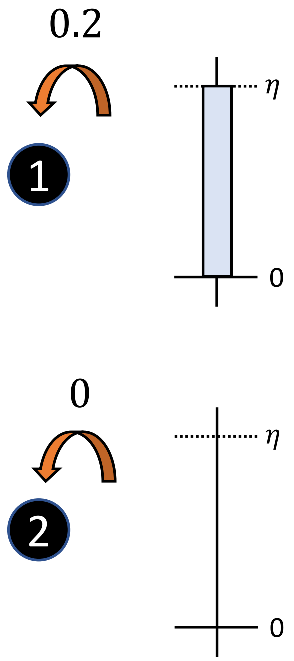
$$\#\text{spikes} = 0;\ \boldsymbol{x}_2(t) = 0; \boldsymbol{u}_2(t) = 0$$

Example

- Setup: $C = \begin{pmatrix} 1 & 0 \\ -0.5 & 1 \end{pmatrix}$, $\boldsymbol{I} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}$, $\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \eta = 1.$

- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n \times n}$
- Firing rate: $\boldsymbol{x}(t)$
- $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

$t = 5$



$\#\text{spikes} = 0;\ \boldsymbol{x}_1(t) = 0; \boldsymbol{u}_1(t) = 1$

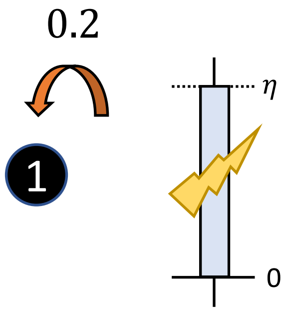
$\#\text{spikes} = 0;\ \boldsymbol{x}_2(t) = 0; \boldsymbol{u}_2(t) = 0$

Example

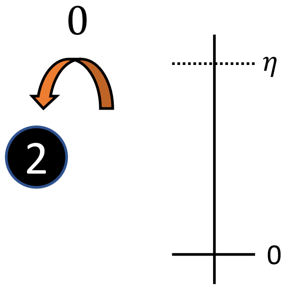
- Setup: $C = \begin{pmatrix} 1 & 0 \\ -0.5 & 1 \end{pmatrix}$, $\boldsymbol{I} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}$, $\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\eta = 1$.

- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n \times n}$
- Firing rate: $\boldsymbol{x}(t)$
- $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

$t = 5$



$\#\text{spikes} = 1;\ \boldsymbol{x}_1(t) = 0.2; \boldsymbol{u}_1(t) = 1$



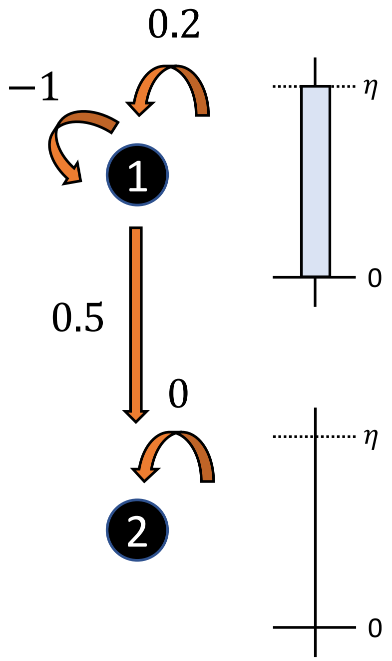
$\#\text{spikes} = 0;\ \boldsymbol{x}_2(t) = 0; \boldsymbol{u}_2(t) = 0$

Example

- Setup: $C = \begin{pmatrix} \boxed{\begin{matrix} 1 \\ -0.5 \end{matrix}} & \begin{matrix} 0 \\ 1 \end{matrix} \end{pmatrix}$, $\boldsymbol{I} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}$, $\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \eta = 1.$

- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n \times n}$
- Firing rate: $\boldsymbol{x}(t)$
- $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

$t = 5$



$\#\text{spikes} = 1;\ \boldsymbol{x}_1(t) = 0.2; \boldsymbol{u}_1(t) = 1$

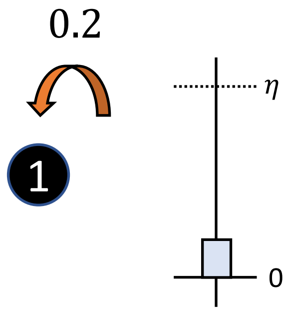
$\#\text{spikes} = 0;\ \boldsymbol{x}_2(t) = 0; \boldsymbol{u}_2(t) = 0$

Example

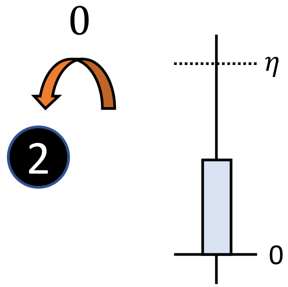
- Setup: $C = \begin{pmatrix} 1 & 0 \\ -0.5 & 1 \end{pmatrix}$, $\boldsymbol{I} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}$, $\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \eta = 1.$

- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n \times n}$
- Firing rate: $\boldsymbol{x}(t)$
- $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

$t = 6$



$$\#\text{spikes} = 1;\ \boldsymbol{x}_1(t) = 0.167; \boldsymbol{u}_1(t) = 0.2$$



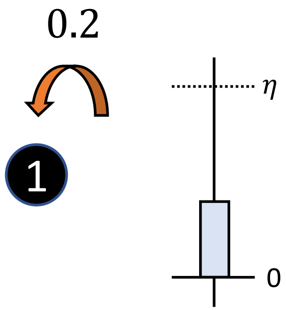
$$\#\text{spikes} = 0;\ \boldsymbol{x}_2(t) = 0; \boldsymbol{u}_2(t) = 0.5$$

Example

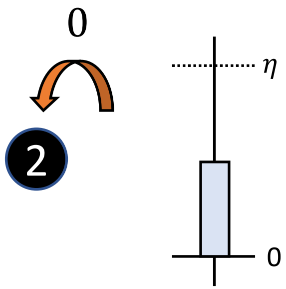
- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n \times n}$
- Firing rate: $\boldsymbol{x}(t)$
- $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

- Setup: $C = \begin{pmatrix} 1 & 0 \\ -0.5 & 1 \end{pmatrix}$, $\boldsymbol{I} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}$, $\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\eta = 1$.

$t = 7$



$\#\text{spikes} = 1;\ \boldsymbol{x}_1(t) = 0.143; \boldsymbol{u}_1(t) = 0.4$



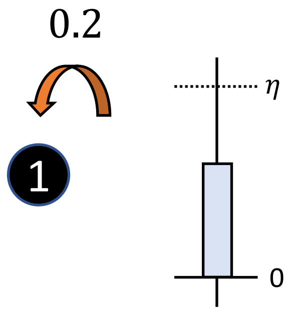
$\#\text{spikes} = 0;\ \boldsymbol{x}_2(t) = 0; \boldsymbol{u}_2(t) = 0.5$

Example

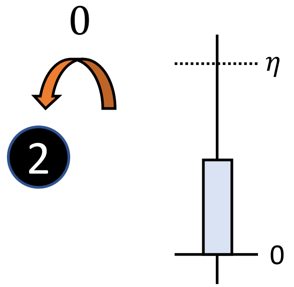
- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n \times n}$
- Firing rate: $\boldsymbol{x}(t)$
- $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

- Setup: $C = \begin{pmatrix} 1 & 0 \\ -0.5 & 1 \end{pmatrix}$, $\boldsymbol{I} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}$, $\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\eta = 1$.

$t = 8$



#spikes $= 1$; $\boldsymbol{x}_1(t) = 0.125$; $\boldsymbol{u}_1(t) = 0.6$



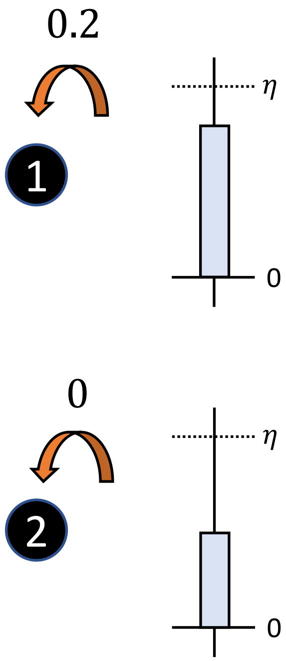
#spikes $= 0$; $\boldsymbol{x}_2(t) = 0$; $\boldsymbol{u}_2(t) = 0.5$

Example

- Setup: $C = \begin{pmatrix} 1 & 0 \\ -0.5 & 1 \end{pmatrix}$, $\boldsymbol{I} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}$, $\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\eta = 1$.

- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n \times n}$
- Firing rate: $\boldsymbol{x}(t)$
- $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

$t = 9$



$\#\text{spikes} = 1;\ \boldsymbol{x}_1(t) = 0.111; \boldsymbol{u}_1(t) = 0.8$

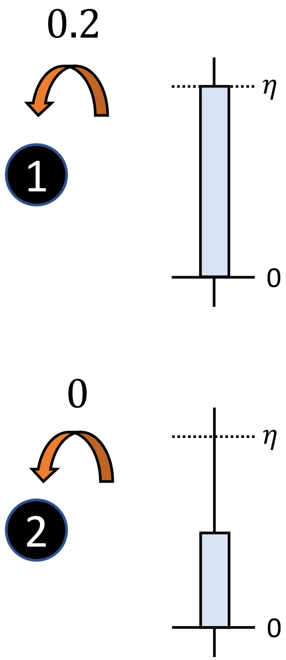
$\#\text{spikes} = 0;\ \boldsymbol{x}_2(t) = 0; \boldsymbol{u}_2(t) = 0.5$

Example

- Setup: $C = \begin{pmatrix} 1 & 0 \\ -0.5 & 1 \end{pmatrix}$, $\boldsymbol{I} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}$, $\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\eta = 1$.

- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n \times n}$
- Firing rate: $\boldsymbol{x}(t)$
- $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

$t = 10$



$\#\text{spikes} = 1;\ \boldsymbol{x}_1(t) = 0.1; \boldsymbol{u}_1(t) = 1$

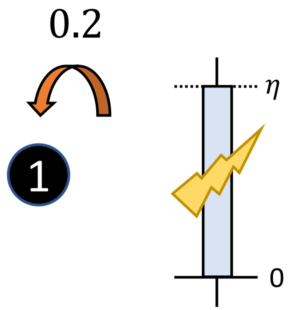
$\#\text{spikes} = 0;\ \boldsymbol{x}_2(t) = 0; \boldsymbol{u}_2(t) = 0.5$

Example

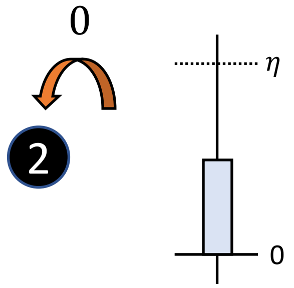
- Setup: $C = \begin{pmatrix} 1 & 0 \\ -0.5 & 1 \end{pmatrix}$, $\boldsymbol{I} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}$, $\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \eta = 1.$

- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n \times n}$
- Firing rate: $\boldsymbol{x}(t)$
- $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

$t = 10$



$\#\text{spikes} = 2;\ \boldsymbol{x}_1(t) = 0.2; \boldsymbol{u}_1(t) = 1$



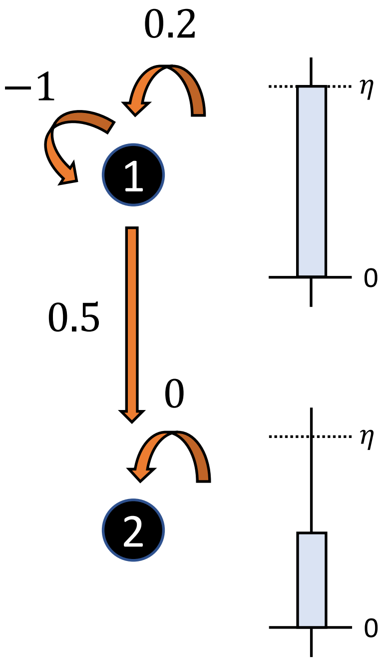
$\#\text{spikes} = 0;\ \boldsymbol{x}_2(t) = 0; \boldsymbol{u}_2(t) = 0.5$

Example

- Setup: $C = \begin{pmatrix} 1 & 0 \\ -0.5 & 1 \end{pmatrix}$, $\boldsymbol{I} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}$, $\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\eta = 1$.

- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n\times n}$
- Firing rate: $\boldsymbol{x}(t)$
- $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

$t = 10$



$\#\text{spikes} = 2;\ \boldsymbol{x}_1(t) = 0.2; \boldsymbol{u}_1(t) = 1$

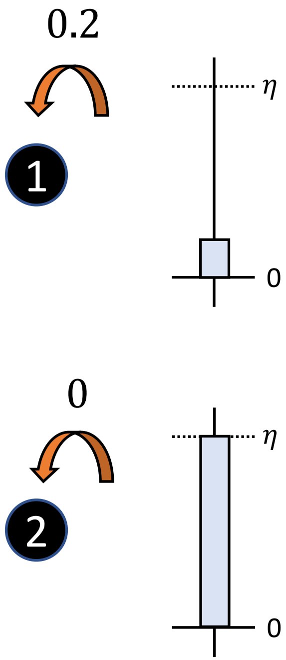
$\#\text{spikes} = 0;\ \boldsymbol{x}_2(t) = 0; \boldsymbol{u}_2(t) = 0.5$

Example

- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n\times n}$
- Firing rate: $\boldsymbol{x}(t)$
- $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

- Setup: $C = \begin{pmatrix} 1 & 0 \\ -0.5 & 1 \end{pmatrix}$, $\boldsymbol{I} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}$, $\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\eta = 1$.

$t = 11$



$\#\text{spikes} = 2;\ \boldsymbol{x}_1(t) = 0.182; \boldsymbol{u}_1(t) = 0.2$

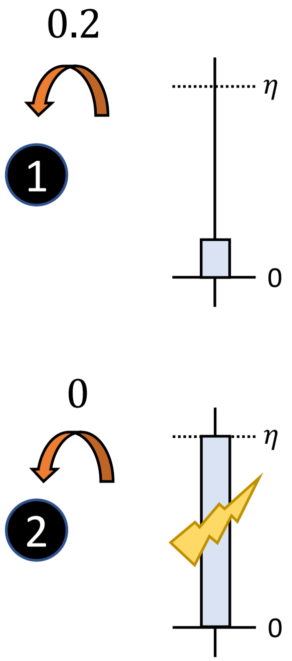
$\#\text{spikes} = 0;\ \boldsymbol{x}_2(t) = 0; \boldsymbol{u}_2(t) = 1$

Example

- Setup: $C = \begin{pmatrix} 1 & 0 \\ -0.5 & 1 \end{pmatrix}$, $\boldsymbol{I} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}$, $\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\eta = 1$.

- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n \times n}$
- Firing rate: $\boldsymbol{x}(t)$
- $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

$t = 11$



$\#\text{spikes} = 2;\ \boldsymbol{x}_1(t) = 0.182; \boldsymbol{u}_1(t) = 0.2$

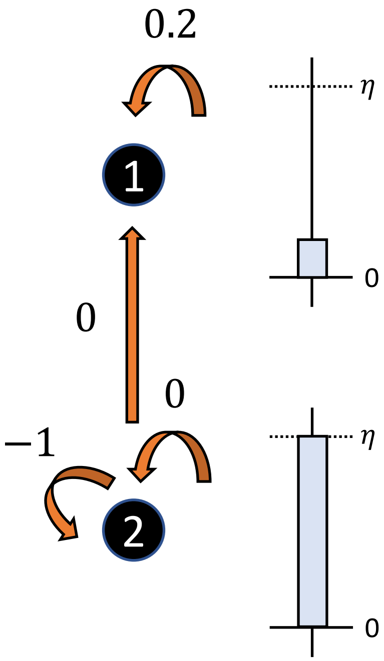
$\#\text{spikes} = 1;\ \boldsymbol{x}_2(t) = 0.09; \boldsymbol{u}_2(t) = 1$

Example

- Setup: $C = \begin{pmatrix} 1 & 0 \\ -0.5 & 1 \end{pmatrix}$, $\boldsymbol{I} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}$, $\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\eta = 1$.

- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n \times n}$
- Firing rate: $\boldsymbol{x}(t)$
- $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

$t = 11$



$\#\text{spikes} = 2;\ \boldsymbol{x}_1(t) = 0.182; \boldsymbol{u}_1(t) = 0.2$

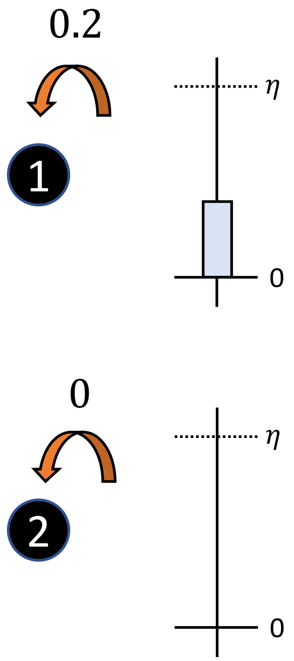
$\#\text{spikes} = 1;\ \boldsymbol{x}_2(t) = 0.09; \boldsymbol{u}_2(t) = 1$

Example

- Setup: $C = \begin{pmatrix} 1 & 0 \\ -0.5 & 1 \end{pmatrix}$, $\boldsymbol{I} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}$, $\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \eta = 1.$

- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n \times n}$
- Firing rate: $\boldsymbol{x}(t)$
- $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

$t = 12$



$\#\text{spikes} = 2;\ \boldsymbol{x}_1(t) = 0.167; \boldsymbol{u}_1(t) = 0.4$

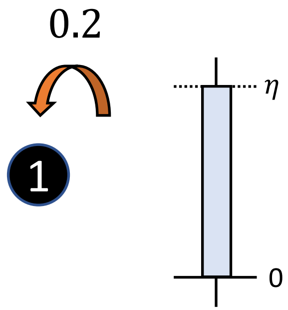
$\#\text{spikes} = 1;\ \boldsymbol{x}_2(t) = 0.83; \boldsymbol{u}_2(t) = 0$

Example

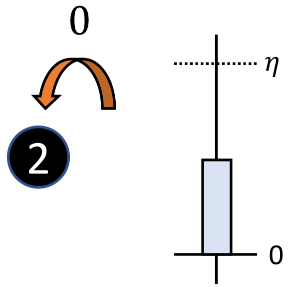
- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n \times n}$
- Firing rate: $\boldsymbol{x}(t)$
- $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

- Setup: $C = \begin{pmatrix} 1 & 0 \\ -0.5 & 1 \end{pmatrix}$, $\boldsymbol{I} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}$, $\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\eta = 1$.

$t = 1000$



$\#\text{spikes} = 200;\ \boldsymbol{x}_1(t) = 0.200; \boldsymbol{u}_1(t) = 1$



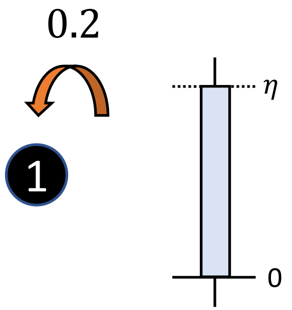
$\#\text{spikes} = 99;\ \boldsymbol{x}_2(t) = 0.099; \boldsymbol{u}_2(t) = 0.5$

Example

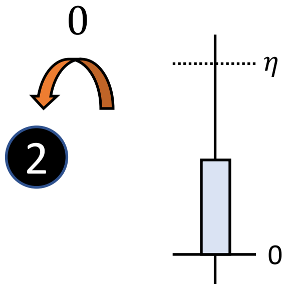
- Setup: $C = \begin{pmatrix} 1 & 0 \\ -0.5 & 1 \end{pmatrix}$, $\boldsymbol{I} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}$, $\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\eta = 1$.

- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$
- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n \times n}$
- Firing rate: $\boldsymbol{x}(t)$
- $\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}$

$t = 1000$



$\#\text{spikes} = 200;\ \boxed{\boldsymbol{x}_1(t) = 0.200},\ \boldsymbol{u}_1(t) = 1$



$\#\text{spikes} = 99;\ \boxed{\boldsymbol{x}_2(t) = 0.099};\ \boldsymbol{u}_2(t) = 0.5$

The Algorithmic Power of Integrate-and-fire SNN?

The Algorithmic Power of Integrate-and-fire SNN?

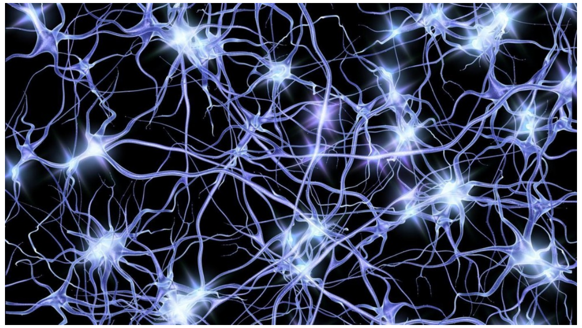
[Barrett-Denève-Machens, NIPS 2013]

Using an "optimization problem" to analyze the firing rate of an integrate-and-fire SNN.

The Algorithmic Power of Integrate-and-fire SNN?

[Barrett-Denève-Machens, NIPS 2013]

Using an "optimization problem" to analyze the firing rate of an integrate-and-fire SNN.



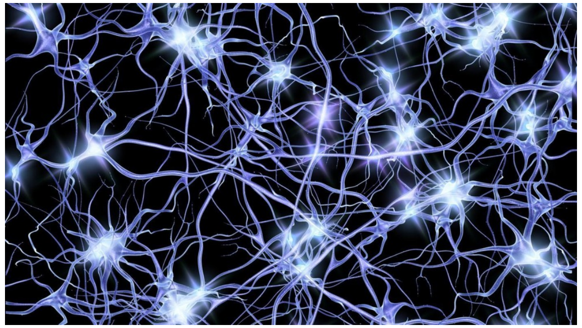
$(C, \boldsymbol{I})$

firing rate

The Algorithmic Power of Integrate-and-fire SNN?

[Barrett-Denève-Machens, NIPS 2013]

Using an "optimization problem" to analyze the firing rate of an integrate-and-fire SNN.



$(C, \boldsymbol{I})$

Non-negative Least Squares

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} ||C\boldsymbol{x} - \boldsymbol{I}||_2^2$$

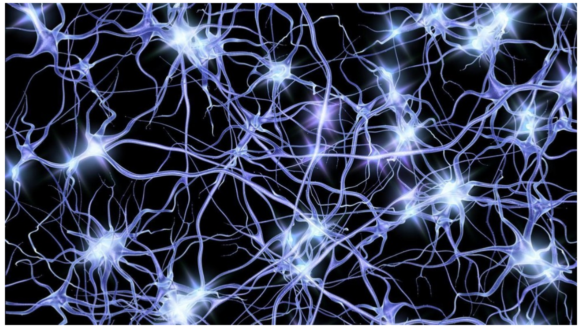
s.t. $\boldsymbol{x}_i \geq 0, \forall i \in [n]$

firing rate $\approx$ **solution**

The Algorithmic Power of Integrate-and-fire SNN?

[Barrett-Denève-Machens, NIPS 2013]

Using an "optimization problem" to analyze the firing rate of an integrate-and-fire SNN.



$(C, \boldsymbol{I})$

$(C, \boldsymbol{I})$ →

Non-negative Least Squares

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \|C\boldsymbol{x} - \boldsymbol{I}\|_2^2$$
s.t. $\boldsymbol{x}_i \geq 0, \forall i \in [n]$

solution →

Using **solution** to estimate **firing rate**

firing rate $\approx$ **solution**

The Algorithmic Power of Integrate-and-fire SNN?

[Barrett-Denève-Machens, NIPS 2013]

Using an "optimization problem" to analyze the firing rate of an integrate-and-fire SNN.

$(C, \boldsymbol{I})$

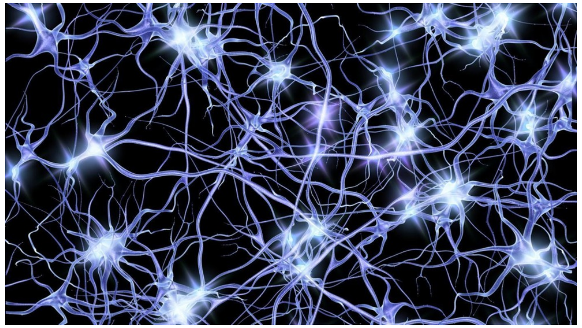
Non-negative Least Squares

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|C\boldsymbol{x} - \boldsymbol{I}\|_2^2$$
$$\text{s.t. } \boldsymbol{x}_i \geq 0, \forall i \in [n]$$

solution

Using solution to estimate firing rate

$(C, \boldsymbol{I})$



$(C, \boldsymbol{I})$

firing rate

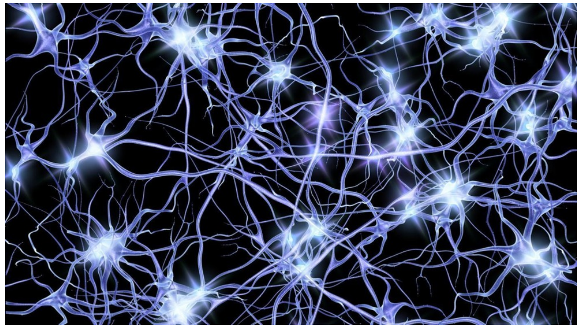
Using firing rate to estimate solution

firing rate $\approx$ solution

The Algorithmic Power of Integrate-and-fire SNN?

[Barrett-Denève-Machens, NIPS 2013]

Using an “optimization problem” to analyze the firing rate of an integrate-and-fire SNN.



$(C, \boldsymbol{I})$

$(C, \boldsymbol{I})$

Non-negative Least Squares

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \|C\boldsymbol{x} - \boldsymbol{I}\|_2^2$$

s.t. $\boldsymbol{x}_i \geq 0, \forall i \in [n]$

solution

Using solution to estimate firing rate

$(C, \boldsymbol{I})$

firing rate

Using firing rate to estimate solution

firing rate $\approx$ **solution**

No provable analysis!

Our Contributions

Our Contributions

The first proof for the firing rate of integrate-and-fire SNNs efficiently solving the non-negative least squares problem.

- Confirm the empirical discovery of [Barret-Denève-Machens 2013].

Our Contributions

The first proof for the firing rate of integrate-and-fire SNNs efficiently solving the non-negative least squares problem.

- Confirm the empirical discovery of [Barret-Denève-Machens 2013].

What if there are infinitely many solutions?

Our Contributions

The first proof for the firing rate of integrate-and-fire SNNs efficiently solving the non-negative least squares problem.

- Confirm the empirical discovery of [Barret-Denève-Machens 2013].

What if there are infinitely many solutions?

- Further show that the firing rate of integrate-and-fire SNN efficiently finds the sparse solution (in the ℓ_1 sense)

Our Contributions

The first proof for the firing rate of integrate-and-fire SNNs efficiently solving the non-negative least squares problem.

- Confirm the empirical discovery of [Barret-Denève-Machens 2013].

What if there are infinitely many solutions?

- Further show that the firing rate of integrate-and-fire SNN efficiently finds the sparse solution (in the ℓ_1 sense), by implementing a primal-dual + projected gradient descent algorithm.

Theorem (ℓ_1 minimization problem)

Theorem (ℓ_1 minimization problem)

Given $A \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, and $\epsilon > 0$. Suppose A satisfies some *regular conditions*. $\boxed{\text{Set } C = \begin{pmatrix} A^\top A & -A^\top A \\ -A^\top A & A^\top A \end{pmatrix}, \boldsymbol{I} = \begin{pmatrix} A^\top \boldsymbol{b} \\ -A^\top \boldsymbol{b} \end{pmatrix}}$, and properly set the integrate-and-fire SNN.

Theorem (ℓ_1 minimization problem)

Given $A \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, and $\epsilon > 0$. Suppose A satisfies some *regular conditions*. $\boxed{\text{Set } C = \begin{pmatrix} A^\top A & -A^\top A \\ -A^\top A & A^\top A \end{pmatrix}, \boldsymbol{I} = \begin{pmatrix} A^\top \boldsymbol{b} \\ -A^\top \boldsymbol{b} \end{pmatrix}}$, and properly set the integrate-and-fire SNN. Let $\boldsymbol{x}^*$ be the optimal solution to the $\boldsymbol{\ell_1}$ **minimization problem**.

$$\begin{array}{ll} \min_{\boldsymbol{x}\in\mathbb{R}^n} & \|\boldsymbol{x}\|_1 \\ \text{s.t.} & A\boldsymbol{x} = \boldsymbol{b} \end{array}$$

Theorem (ℓ_1 minimization problem)

Given $A \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, and $\epsilon > 0$. Suppose A satisfies some *regular conditions*. $\boxed{\text{Set } C = \begin{pmatrix} A^\top A & -A^\top A \\ -A^\top A & A^\top A \end{pmatrix}, \boldsymbol{I} = \begin{pmatrix} A^\top \boldsymbol{b} \\ -A^\top \boldsymbol{b} \end{pmatrix}}$, and properly set the integrate-and-fire SNN. Let $\boldsymbol{x}^*$ be the optimal solution to the **ℓ_1 minimization problem**.

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n} \quad & \|\boldsymbol{x}\|_1 \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \end{aligned}$$

When $t \geq \Omega(\frac{n^3}{\epsilon^2})$, we have (i) $\|\boldsymbol{b} - A\boldsymbol{x}(t)\|_2 \leq \epsilon \cdot \|\boldsymbol{b}\|_2$ and (ii) $\|\boldsymbol{x}(t)\|_1 - \|\boldsymbol{x}^*\|_1 \leq \epsilon \cdot \|\boldsymbol{x}^*\|_1$.

Theorem (ℓ_1 minimization problem)

Given $A \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, and $\epsilon > 0$. Suppose A satisfies some *regular conditions*. $\boxed{\text{Set } C = \begin{pmatrix} A^\top A & -A^\top A \\ -A^\top A & A^\top A \end{pmatrix}, \boldsymbol{I} = \begin{pmatrix} A^\top \boldsymbol{b} \\ -A^\top \boldsymbol{b} \end{pmatrix}}$, and properly set the integrate-and-fire SNN. Let $\boldsymbol{x}^*$ be the optimal solution to the **ℓ_1 minimization problem**.

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n} \quad & \|\boldsymbol{x}\|_1 \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \end{aligned}$$

When $\boxed{t \geq \Omega(\frac{n^3}{\epsilon^2})}$, we have (i) $\|\boldsymbol{b} - A\boldsymbol{x}(t)\|_2 \leq \epsilon \cdot \|\boldsymbol{b}\|_2$ and (ii) $\|\boldsymbol{x}(t)\|_1 - \|\boldsymbol{x}^*\|_1 \leq \epsilon \cdot \|\boldsymbol{x}^*\|_1$.

Key Technique – A Dual View of SNN

Key Technique – A Dual View of SNN

	Primal SNN	**Dual SNN**
Dynamics	$\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - \begin{pmatrix} A^\top A & -A^\top A \\ -A^\top A & A^\top A \end{pmatrix} \boldsymbol{s}(t) + \begin{pmatrix} A^\top \boldsymbol{b} \\ -A^\top \boldsymbol{b} \end{pmatrix}$	
Optimization problem		

Key Technique – A Dual View of SNN

	Primal SNN	**Dual SNN**
Dynamics	$\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - \begin{pmatrix} A^\top A & -A^\top A \\ -A^\top A & A^\top A \end{pmatrix} \boldsymbol{s}(t) + \begin{pmatrix} A^\top \boldsymbol{b} \\ -A^\top \boldsymbol{b} \end{pmatrix}$	
Optimization problem	$\min_{\boldsymbol{x} \in \mathbb{R}^n} \ \|\boldsymbol{x}\|_1$ s.t. $A\boldsymbol{x} = \boldsymbol{b}$ (ℓ_1 minimization)	

Key Technique – A Dual View of SNN

	Primal SNN	**Dual SNN**
Dynamics	$\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - \begin{pmatrix} A^\top A & -A^\top A \\ -A^\top A & A^\top A \end{pmatrix} \boldsymbol{s}(t) + \begin{pmatrix} A^\top \boldsymbol{b} \\ -A^\top \boldsymbol{b} \end{pmatrix}$	$\boldsymbol{v}(t+1) = \boldsymbol{v}(t) - (A \quad -A)\boldsymbol{s}(t) + \boldsymbol{b}$
Optimization problem	$\min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{x}\|_1$ s.t. $A\boldsymbol{x} = \boldsymbol{b}$ (ℓ_1 minimization)	

Key Technique – A Dual View of SNN

	Primal SNN	**Dual SNN**
Dynamics	$\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - \begin{pmatrix} A^\top A & -A^\top A \\ -A^\top A & A^\top A \end{pmatrix} \boldsymbol{s}(t) + \begin{pmatrix} A^\top \boldsymbol{b} \\ -A^\top \boldsymbol{b} \end{pmatrix}$	$\boldsymbol{v}(t+1) = \boldsymbol{v}(t) - (A \quad -A)\boldsymbol{s}(t) + \boldsymbol{b}$
Optimization problem	$\min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{x}\|_1$ s.t. $A\boldsymbol{x} = \boldsymbol{b}$ (ℓ_1 minimization)	$\max_{\boldsymbol{v} \in \mathbb{R}^m} \boldsymbol{b}^\top \boldsymbol{v}$ s.t. $\|A^\top \boldsymbol{v}\|_\infty \le 1$ (The **dual** of ℓ_1 minimization)

Key Technique – A Dual View of SNN

	Primal SNN	**Dual SNN**
Dynamics	$\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - \begin{pmatrix} A^\top A & -A^\top A \\ -A^\top A & A^\top A \end{pmatrix} \boldsymbol{s}(t) + \begin{pmatrix} A^\top \boldsymbol{b} \\ -A^\top \boldsymbol{b} \end{pmatrix}$	$\boldsymbol{v}(t+1) = \boldsymbol{v}(t) - (A \quad -A)\boldsymbol{s}(t) + \boldsymbol{b}$
Optimization problem	$\min_{\boldsymbol{x}\in\mathbb{R}^n} \ \|\boldsymbol{x}\|_1$ s.t. $A\boldsymbol{x} = \boldsymbol{b}$ (ℓ_1 minimization)	$\max_{\boldsymbol{v}\in\mathbb{R}^m} \ \boldsymbol{b}^\top \boldsymbol{v}$ s.t. $\|A^\top \boldsymbol{v}\|_\infty \leq 1$ (The dual of ℓ_1 minimization)

$\boldsymbol{v}(t)$ is a **projected gradient descent** algorithm for dual program with **non-standard projection.**

Key Technique – A Dual View of SNN

External charging
≈
Gradient

	Primal SNN	Dual SNN
Dynamics	$\boldsymbol{u}(t+1)=\boldsymbol{u}(t)-\begin{pmatrix} A^\top A & -A^\top A \\ -A^\top A & A^\top A \end{pmatrix}\boldsymbol{s}(t)+\begin{pmatrix} A^\top \boldsymbol{b} \\ -A^\top \boldsymbol{b} \end{pmatrix}$	$\boldsymbol{v}(t+1)=\boldsymbol{v}(t)-\begin{pmatrix} A & -A \end{pmatrix}\boldsymbol{s}(t)\boxed{+\boldsymbol{b}}$
Optimization problem	$\min_{\boldsymbol{x}\in\mathbb{R}^n} \ \lVert\boldsymbol{x}\rVert_1$ s.t. $A\boldsymbol{x}=\boldsymbol{b}$ (ℓ_1 minimization)	$\boxed{\max_{\boldsymbol{v}\in\mathbb{R}^m} \ \boldsymbol{b}^\top\boldsymbol{v}}$ s.t. $\lVert A^\top\boldsymbol{v}\rVert_\infty\le 1$ (The dual of ℓ_1 minimization)

$\boldsymbol{v}(t)$ is a **projected gradient descent** algorithm for dual program with **non-standard projection.**

Key Technique – A Dual View of SNN

Spiking effect
≈
Projection

	Primal SNN	**Dual SNN**
Dynamics	$\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - \begin{pmatrix} A^\top A & -A^\top A \\ -A^\top A & A^\top A \end{pmatrix} \boldsymbol{s}(t) + \begin{pmatrix} A^\top \boldsymbol{b} \\ -A^\top \boldsymbol{b} \end{pmatrix}$	$\boldsymbol{v}(t+1) = \boldsymbol{v}(t) \boxed{- (A \quad -A)\boldsymbol{s}(t)} + \boldsymbol{b}$
Optimization problem	$\min_{\boldsymbol{x}\in\mathbb{R}^n} \ \|\|\boldsymbol{x}\|\|_1$ s.t. $A\boldsymbol{x} = \boldsymbol{b}$ (ℓ_1 minimization)	$\max_{\boldsymbol{v}\in\mathbb{R}^m} \ \boldsymbol{b}^\top \boldsymbol{v}$ s.t. $\boxed{\|\|A^\top \boldsymbol{v}\|\|_\infty \leq 1}$ (The dual of ℓ_1 minimization)

$\boldsymbol{v}(t)$ is a **projected gradient descent** algorithm for dual program with **non-standard projection.**

Key Technique – A Dual View of SNN

	Primal SNN	Dual SNN
Dynamics	$\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - \begin{pmatrix} A^\top A & -A^\top A \\ -A^\top A & A^\top A \end{pmatrix} \boldsymbol{s}(t) + \begin{pmatrix} A^\top \boldsymbol{b} \\ -A^\top \boldsymbol{b} \end{pmatrix}$	$\boldsymbol{v}(t+1) = \boldsymbol{v}(t) - (A \quad -A)\boldsymbol{s}(t) + \boldsymbol{b}$
Optimization problem	$\min_{\boldsymbol{x}\in\mathbb{R}^n} \ \|\boldsymbol{x}\|_1$ s.t. $A\boldsymbol{x} = \boldsymbol{b}$ (ℓ_1 minimization)	$\max_{\boldsymbol{v}\in\mathbb{R}^m} \ \boldsymbol{b}^\top \boldsymbol{v}$ s.t. $\|A^\top \boldsymbol{v}\|_\infty \leq 1$ (The **dual** of ℓ_1 minimization)

The **firing rate** in primal SNN solves the primal program.

KKT conditions

$\leftarrow$

Perturbation theory

$\boldsymbol{v}(t)$ is a **projected gradient descent** algorithm for dual program with **non-standard projection.**

Key Technique – A Dual View of SNN

Spikes are **non-monotone** and difficult to analyze!

	Primal SNN	**Dual SNN**
Dynamics	$\boldsymbol{u}(t+1) = \boldsymbol{u}(t) - \begin{pmatrix} A^\top A & -A^\top A \\ -A^\top A & A^\top A \end{pmatrix} \boldsymbol{s}(t) + \begin{pmatrix} A^\top \boldsymbol{b} \\ -A^\top \boldsymbol{b} \end{pmatrix}$	$\boldsymbol{v}(t+1) = \boldsymbol{v}(t) - (A \quad -A)\boldsymbol{s}(t) + \boldsymbol{b}$
Optimization problem	$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{x}\|_1$ s.t. $A\boldsymbol{x} = \boldsymbol{b}$ (ℓ_1 minimization)	$\max_{\boldsymbol{v}\in\mathbb{R}^m} \boldsymbol{b}^\top \boldsymbol{v}$ s.t. $\|A^\top \boldsymbol{v}\|_\infty \le 1$ (The **dual** of ℓ_1 minimization)

The **firing rate** in primal SNN solves the primal program.

KKT conditions

⟵

Perturbation theory

$\boldsymbol{v}(t)$ is a **projected gradient descent** algorithm for dual program with **non-standard projection.**

Perspectives – Natural Algorithms

Perspectives – Natural Algorithms

"How algorithmic ideas can enrich our understanding of nature?"
- Bernard Chazelle

Perspectives – Natural Algorithms

“How algorithmic ideas can enrich our understanding of nature?”

- Bernard Chazelle

Through the lens of natural algorithms, can we understand SNNs more via its algorithmic power and even discover new algorithmic ideas?

Perspectives – Natural Algorithms

"How algorithmic ideas can enrich our understanding of nature?"

- Bernard Chazelle

Through the lens of natural algorithms, can we understand SNNs more via its algorithmic power and even discover new algorithmic ideas?

In this work, we show that integrate-and-fire SNN uses its firing rate to efficiently solve some optimization problems in a primal-dual way!

Conclusions

Conclusions

- In this work, we give the first proof for the firing rate of integrate-and-fire SNNs efficiently solving non-negative least squares problem and ℓ_1 minimization problem.

Conclusions

- In this work, we give the first proof for the firing rate of integrate-and-fire SNNs efficiently solving non-negative least squares problem and ℓ_1 minimization problem.
- Related works.
  - Universality and computational complexity.
    - SNN is able to simulate Turing machines, random access machines (RAM), and threshold circuits etc. [Maa96, Maa97b, Maa99, MB01].
  - Using SNNs to solve computational/optimization problems.
    - Sparse coding [ZMD11, Tan16, TLD17], dictionary learning [LT18], pattern recognition [DC15, KGM16, BMF+17], and non-negative least squares[BDM13].
    - Implementing MCMC to solve traveling salesman problem (TSP) and constraint satisfaction problem (CSP) [BBNM11, JHM14, Maa15, JHM16].
    - Assemblies of neurons and random projection [ADMPSV18, LPVM18, PV19].
  - The efficiency of SNNs in solving computational problems.
    - Solving Winner-Take-All (WTA) problem, similarity testing, and neural coding [LMP17a, LMP17b, LMP17c, LM18].

Conclusions

Very few works provide provable analysis on the efficiency of SNN algorithms!

- In this work, we give the first proof for the firing rate of integrate-and-fire SNNs efficiently solving non-negative least squares problem and ℓ_1 minimization problem.
- Related works.
  - Universality and computational complexity.
    - SNN is able to simulate Turing machines, random access machines (RAM), and threshold circuits etc. [Maa96, Maa97b, Maa99, MB01].
  - Using SNNs to solve computational/optimization problems.
    - Sparse coding [ZMD11, Tan16, TLD17], dictionary learning [LT18], pattern recognition [DC15, KGM16, BMF+17], and non-negative least squares[BDM13].
    - Implementing MCMC to solve traveling salesman problem (TSP) and constraint satisfaction problem (CSP) [BBNM11, JHM14, Maa15, JHM16].
    - Assemblies of neurons and random projection [ADMPSV18, LPVM18, PV19].
  - The efficiency of SNNs in solving computational problems.
    - Solving Winner-Take-All (WTA) problem, similarity testing, and neural coding [LMP17a, LMP17b, LMP17c, LM18].

Conclusions

Very few works provide provable analysis on the efficiency of SNN algorithms!

- In this work, we give the first proof for the firing rate of integrate-and-fire SNNs efficiently solving non-negative least squares problem and ℓ_1 minimization problem.
- Related works.
  - Universality and computational complexity.
    - SNN is able to simulate Turing machines, random access machines (RAM), and threshold circuits etc. [Maa96, Maa97b, Maa99, MB01].
  - Using SNNs to solve computational/optimization problems.
    - Sparse coding [ZMD11, Tan16, TLD17], dictionary learning [LT18], pattern recognition [DC15, KGM16, BMF+17], and non-negative least squares[BDM13].
    - Implementing MCMC to solve traveling salesman problem (TSP) and constraint satisfaction problem (CSP) [BBNM11, JHM14, Maa15, JHM16].
    - Assemblies of neurons and random projection [ADMPSV18, LPVM18, PV19].
  - The efficiency of SNNs in solving computational problems.
    - Solving Winner-Take-All (WTA) problem, similarity testing, and neural coding [LMP17a, LMP17b, LMP17c, LM18].
- Next step?
  - Giving more rigorous analysis for the efficiency of other SNN algorithms!

Very few works provide provable analysis on the efficiency of SNN algorithms!

Conclusions

- In this work, we give the first proof for the firing rate of integrate-and-fire SNNs efficiently solving non-negative least squares problem and ℓ_1 minimization problem.
- Related works.
  - Universality and computational complexity.
    - SNN is able to simulate Turing machines, random access machines (RAM), and threshold circuits etc. [Maa96, Maa97b, Maa99, MB01].
  - Using SNNs to solve computational/optimization problems.
    - Sparse coding [ZMD11, Tan16, TLD17], dictionary learning [LT18], pattern recognition [DC15, KGM16, BMF+17], and non-negative least squares[BDM13].
    - Implementing MCMC to solve traveling salesman problem (TSP) and constraint satisfaction problem (CSP) [BBNM11, JHM14, Maa15, JHM16].
    - Assemblies of neurons and random projection [ADMPSV18, LPVM18, PV19].
  - The efficiency of SNNs in solving computational problems.
    - Solving Winner-Take-All (WTA) problem, similarity testing, and neural coding [LMP17a, LMP17b, LMP17c, LM18].
- Next step?
  - Giving more rigorous analysis for the efficiency of other SNN algorithms!

Thanks for your attention!