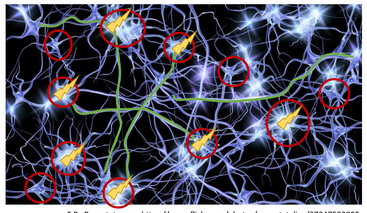
On the Algorithmic Power of Spiking Neural Networks

Chi-Ning Chou[1], Kai-Min Chung[2], Chi-Jen Lu[2]
[1]Harvard University, MA, USA; [2]Academia Sinica, Taipei, Taiwan

What is Spiking Neural Networks (SNNs)?

- Mathematical models for "biological neural networks".



Neurons: Nerve cells

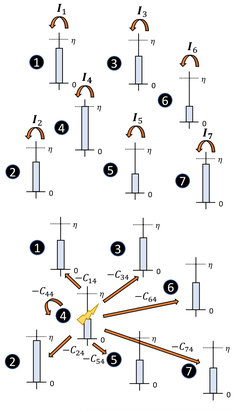
Synapse: Connection between neurons

Spikes: Instantaneous signals

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

- 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].
- [Barret-Denève-Machens 2013] empirically showed a connection between the firing rate of integrate-and-fire SNNs and an optimization problem.
- Investigate (integrate-and-fire) SNNs through the lens of algorithms?

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



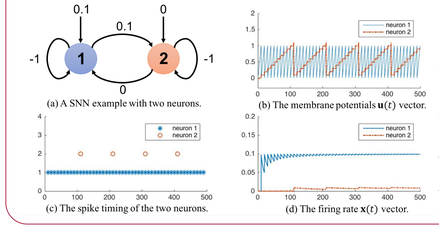
- Neurons: $[n] = \{1, 2, \dots, n\}$
- Potential: $\boldsymbol{u}(t) \in \mathbb{R}^n$
- External charging: $\boldsymbol{I} \in \mathbb{R}^n$

Firing Rule
Neuron i fires a spike at time t if $\boldsymbol{u}_i(t) \geq \eta$.

- Spikes: $\boldsymbol{s}(t) \in \{0,1\}^n$
- Connectivity: $C \in \mathbb{R}^{n \times n}$
- Firing rate: $\boldsymbol{x}(t) =$ (#spikes before time t)$/t$

Dynamics: $\boldsymbol{u}(t + \Delta t) = \boldsymbol{u}(t) - C\boldsymbol{s}(t) + \boldsymbol{I}\Delta t$

Example



- Setup:

$C = \begin{pmatrix} 1 & 0 \\ -0.1 & 1 \end{pmatrix}$,

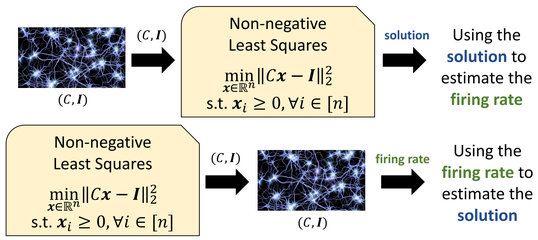
$\boldsymbol{I} = \begin{pmatrix} 0.1 \\ 0 \end{pmatrix}$,

$\boldsymbol{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$,

$\eta = 1$.

The Algorithmic Power of SNN?

[Barrett-Denève-Machens, NIPS 2013] Using a "quadratic program" to analyze the firing rate of an SNN.



SNN Solves Non-negative Least Squares

[Barret-Denève-Machens 2013] Empirical evidence, no theoretical guarantee.

[Tang-Lin-Davies 2017] Convergence in the limit using Lyapunov theory.

[This work] The first convergence rate analysis.

Theorem 1 (informal)**.** Given $A \in \mathbb{R}^{m \times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, and $\epsilon > 0$. Suppose A satisfies some *regular conditions*. Set $C = A^\top A$, $\boldsymbol{I} = A^\top \boldsymbol{b}$, and properly set the SNN. Let $\boldsymbol{x}^*$ be the optimal solution to the following quadratic program.

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

When $t \geq \Omega(\frac{\lambda_{\max}(A)\sqrt{n}}{\epsilon \cdot \|\boldsymbol{b}\|_2})$, $\|A\boldsymbol{x}(t) - A\boldsymbol{x}^*\|_2 \leq \epsilon \cdot \|\boldsymbol{b}\|_2$.

A Simple Proof for Solving Least Squares

Consider *two-sided* SNN where $C = \begin{pmatrix} A^\top A & -A^\top A \\ -A^\top A & A^\top A \end{pmatrix}$, $\boldsymbol{I} = \begin{pmatrix} A^\top \boldsymbol{b} \\ -A^\top \boldsymbol{b} \end{pmatrix}$, $\boldsymbol{u}(0) = \boldsymbol{0}$. Observe that

$$\boldsymbol{u}(t) = \int_{\tau=0}^{\tau=t} d\boldsymbol{u}(\tau) = -A^\top A\boldsymbol{x}(t)t + A^\top \boldsymbol{b}t.$$

That is, $\|A\boldsymbol{x}(t) - \boldsymbol{b}\|_2 \leq \sqrt{\lambda_{\min}^{-1}(A^\top A)} \cdot \|\boldsymbol{u}(t)\|_2$. Namely, to bound the residual error, it suffices to bound $\|\boldsymbol{u}(t)\|_2$.

Lemma. For any $t \geq 0$, $\|\boldsymbol{u}(t)\|_2 = \sqrt{\lambda_{\min}^{-1}(A^\top A)} \cdot \eta \cdot \sqrt{n}$ when $\eta \geq \lambda_{\max}$.

Proof. When there's a neuron firing a spikes, $\|\boldsymbol{u}(\tau)\|_{(A^\top A)^\dagger}$ won't increase because

$$\boldsymbol{u}(\tau)^\top \boldsymbol{s}(\tau) \geq \eta \cdot \#\text{ spikes fired at time } \tau.$$

What If There Are Many Solutions?

In many applications, non-negative least squares problem might have infinitely many solutions. Which one would SNN converge to?

The firing rate converges to the **sparse solution**!

Theorem 2 (informal)**.** Given $A \in \mathbb{R}^{m \times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, and $\epsilon > 0$. Suppose A satisfies some *regular conditions*. Set $C = A^\top A$, $\boldsymbol{I} = A^\top \boldsymbol{b}$, and properly set the SNN. Let $\boldsymbol{x}^*$ be the optimal solution to the l1 minimization problem.

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{x}\|_1 \quad \text{s.t.} \quad \|A\boldsymbol{x} - \boldsymbol{b}\|_2 = 0, \ \boldsymbol{x}_i \geq 0, \forall i \in [n]$$

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) $\|\boldsymbol{x}(t)\|_1 - \|\boldsymbol{x}^*\|_1 \leq \epsilon \cdot \|\boldsymbol{x}^*\|_1$.

Key Technique: A Dual View of SNN

Recall that the dynamic of SNN (**primal SNN**) is

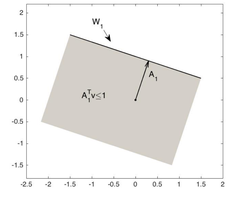
$$\boldsymbol{u}(t + \Delta t) = \boldsymbol{u}(t) - A^\top A\boldsymbol{s}(t) + A^\top \boldsymbol{b}\Delta t.$$

The **dual SNN** is defined as follows.

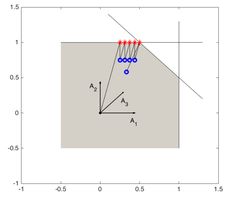
$$\boldsymbol{v}(t + \Delta t) = \boldsymbol{v}(t) - A\boldsymbol{s}(t) + \boldsymbol{b}\Delta t.$$

	Primal SNN	Dual SNN
Spiking rule	$\boldsymbol{u}_i(t) > \eta$	$A_i^\top \boldsymbol{v}(t) > \eta$
Spiking effect	$-A^\top A_i$	$-A_i$
Optimization problem	$\min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{x}\|_1$ s.t. $A\boldsymbol{x} = \boldsymbol{b}$	$\max_{\boldsymbol{v} \in \mathbb{R}^m} \boldsymbol{b}^\top \boldsymbol{v}$ s.t. $\|A^\top \boldsymbol{v}\|_\infty \leq \eta$

Geometric Interpretation of the Dual SNN



An example of one neuron where $A_1 = \begin{bmatrix} 1 \\ \frac{1}{2} \end{bmatrix}^\top$. The neuron won't fire if the **dual SNN** stays in the gray area. We call the boundary of the gray area as a ``*wall*".



An example of three neurons where $A_1 = [1\ 0]^\top$, $A_2 = [0\ 1]^\top$, $A_3 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}^\top$. The **dual SNN** is moving along the direction of $\boldsymbol{b}$ and fires a spike once it leaves the gray area and jumps back in the normal direction of the wall it touches.

Two Main Steps in the Proof

Q1: The convergence of **dual SNN** to the **dual of l1 minimization**?

Ans: Identify a contrive *potential function* that keeps track of the progress of **dual SNN**. Involving structure of the $\|A^\top \boldsymbol{v}\|_\infty \leq \eta$ polytope.

Q2: How to get from **dual solution** to the **primal solution** to the **l1 minimization**?

Ans: Using KKT theorem and Perturbation theory. Ask me for more details!

Perspectives – Natural Algorithms

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

Q: What "computational problems" can SNNs efficiently solve?

Q: How does SNNs solve these problems?

Q: What "algorithms" can SNNs efficiently implement?

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

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 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 & Future Directions

- In this work,
  - (**Algorithmic results**) The first proof for integrate-and-fire SNNs efficiently solving the non-negative least squares problem and the l1 minimization problem.
  - (**Technical results**) The discovery of a dual view of SNN.
- Next step?
  - Giving more rigorous analysis for the efficiency of other SNN algorithms!