Ludwig-Maximilians-Universität at München
Department “Institut für TODO”
Lehr- und Forschungseinheit TODO
Prof. Dr. PROFESSORS NAME

Bachelor’s Thesis

THESIS TITLE

Valentin Herrmann
EMAIL

Bearbeitungszeitraum:	START bis END
Betreuer:	SUPERVISOR
Verantw. Hochschullehrer:	Prof. Dr. PROFESSORS NAME

Aufgabenstellung

THESIS TITLE

Problem Statement TO BE ADDED

Scope of the Thesis TO BE ADDED

Tasks TO BE ADDED

Requirements TO BE ADDED

Keywords TO BE ADDED

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig angefertigt, alle Zitate als solche kenntlich gemacht sowie alle benutzten Quellen und Hilfsmittel angegeben habe.

München, August 25, 2025 .....

Acknowledgments

I would like to appreciate ...

Abstract

This thesis proposes ...

Contents

1 Introduction

While the hype on AI is still ongoing there are still people wondering if current AI models are fundamentally able of reasoning (TODO: Apple Paper). Many other possible models could be better fitted to the task of reasoning. Among others Spiking neural networks present a model closer to the workings of the brain. In fact, they have been called the 3rd generation of AI models, after the 2nd generation that currently drives most successful models.

While the idea behind SNNs is quite old, they have not been as much researched, since it is much more inefficient and harder to train them. Therefore there still remain a lot of open questions about them. In this paper we shall extend on the work done in [Nguyen et al., 2025]. We will add a decaying factor to the input of the neurons and allow recursive connections between neurons in a layer.

We will roughly follow the structure of [Nguyen et al., 2025]. In the second chapter we will formally introduce discrete time leaky-integrate-and-fire SNN, d.t. LIF-SNNs. In section 3 we will give theorems about approximation of continuous functions on compact domains by d.t. LIF-SNNs. The main part will be the following section in which we will see that the number of distinct values a d.t. LIF-SNN can take on only depends on the first hidden layer and grows in particular only quadratically in time. We will further support our findings with experimental data in the last section.

2 Definitions

Our type of SNN should be thought of as a composition of an initial input layer, a number of hidden spiking layers with internal state and an affine-linear layer mapping spikes activations over time, so called spike trains, to the value of the output layer. Like motivated we are adding additional structure to the definitions [Nguyen et al., 2025]. We shall first define the input $i^{[l]}(t)$, the membrane potential before spike $p^{[l]}(t)$, the membrane potential after spike $u^{[l]}(t)$ and the spike activations $s^{[l]}(t)$ of the hidden layers:

Definition 2.1. The **input vector** $i^{[l]}(t) \in \{0,1\}^{n_l}$, the **spike vector** $s^{[l]}(t) \in \{0,1\}^{n_l}$ and the **membrane potential vector** $u^{[l]}(t)$ of a hidden layer $\lambda = (W^{[l]}, b^{[l]}, u^{[l]}(0), i^{[l]}(0), \alpha^{[l]}, \beta^{[l]}, \vartheta^{[l]})$, $l \in [L]$, are recursively defined as

$$i^{[l]}(t) := \alpha^{[l]} i^{[l]}(t-1) + W^{[l]} s^{(l)-1}(t) + V^{[l]} s^{[l]}(t-1) \tag{1}$$

$$p^{[l]}(t) := \beta^{[l]} u^{[l]}(t-1) + i^{[l]}(t) + b^{[l]} \tag{2}$$

$$s^{[l]}(t) := H(p^{[l]}(t) - \vartheta 1_{n_l}) \tag{3}$$

$$u^{[l]}(t) := p^{[l]}(t) - \vartheta s^{[l]}(t) \tag{4}$$

with $s^{[l]}(0) = 0$ and given

- **initial membrane potential**: $u^{[l]}(0) \in \mathbb{R}^{n_l}$,
- **initial input**: $i^{[l]}(0) \in \mathbb{R}^{n_l}$,
- **weight matrices**: $W^{[l]} \in \mathbb{R}^{n_l \times n_{l-1}}$, $V^{[l]} \in \mathbb{R}^{n_l \times n_l}$,
- **bias vectors**: $b^{[l]} \in \mathbb{R}^{n_l}$,
- **leaky terms**: $\alpha^{[l]}, \beta^{[l]} \in [0,1]$,
- **threshold**: $\vartheta^{[l]} \in (0, \infty)$,

where $H := \mathbb{1}_{[0,\infty)}$ is a step function and $T \in \mathbb{N}$ is the number of simulated time steps.

We further define recurrent d.t. LIF-SNN and the function the network realizes:

Definition 2.2. A **recurrent discrete-time LIF-SNN** of **depth** L with **layer-widths** $(n_0, ..., n_{L+1})$ and $T \in \mathbb{N}$ time-steps is given by

$$\Phi := ((W^{[l]}, b^{[l]}, V^{[l]}, u^{[l]}(0), i^{[l]}(0), \alpha^{[l]}, \beta^{[l]}, \vartheta^{[l]})_{l \in [L]}, T, (E, D))$$

where the **input encoder** $E : \mathbb{R}^{n_0} \to \mathbb{R}^{n_0 \times T}$ maps a vector $x \in \mathbb{R}^{n_0}$ to a corresponding first layer spike activation $s^{[0]} = E(x)$ and the **output decoder** $D : \{0,1\}^{n_L \times T} \to \mathbb{R}^{n_{L+1}}$ maps the spike activations of the last hidden layer to real values.

Definition 2.3. A recurrent discrete-time LIF-SNN ϕ **realizes** the function $R(\Phi) : \mathbb{R}^{n_0} \to \mathbb{R}^{n_{L+1}}$:

$$R(\Phi)(x) = D((s^{[L]}(t))_{t \in [T]}) \quad \text{with } s^{[0]} := E(x)$$

Definition 2.4. A recurrent discrete-time LIF-SNN employs **direct encoding** if we have

$$\forall_{t \in [T]} E(x)(t) = x$$

for the input encoder and has **membrane potential outputs** if the output decoder can be written as

$$D((s(t))_{t \in [T]}) = \sum_{t=1}^{T} a_t (W^{L+1} s(t) + b^{L+1})$$

for some $(a_t)_{t \in [T]} \in \mathbb{R}^T$, $b^{L+1} \in \mathbb{R}^{n_{L+1}}$ and $W^{L+1} \in \mathbb{R}^{n_{L+1} \times n_L}$.

We will only consider recurrent discrete-time LIF-SNN with direct encoding and membrane potential outputs. In fact, we will use "recurrent discrete-time LIF-SNN" to mean "r. d.t. LIF-SNN with direct encoding and membrane potential".

For clearer construction of our networks we will additionally define neurons:

Definition 2.5. The i-th neuron of a hidden layer $\lambda = (W^{[l]}, b^{[l]}, V^{[l]}, u^{[l]}(0), i^{[l]}(0), \alpha^{[l]}, \beta^{[l]}, \vartheta^{[l]})$ of a r. d.t. LIF-SNN is a tuple (w, b, v, u_0, i_0) with $w \in \mathbb{R}^{n_{l-1}}$, $v \in \mathbb{R}^{n_l}$ and $b, u_0, i_0 \in \mathbb{R}$, such that b, u_0, i_0 are the i-th component of $b^{[l]}$, $u^{[l]}(0)$, $i^{[l]}(0)$ respectively and w, v are the i-th row vector of $W^{[l]}$, $V^{[l]}$ respectively.

The following lemma will be very helpful for the proofs in the following sections:

Lemma 2.1. *We define the following non-recursive formulas with explicit reference to the previous spikes. Let $1 \leq t \leq T$ and $(s_p^{[l]})_{l \in \{0 \dots L'\}}$ such that $s_p^{[l]} : \{0,1\}^{n_l \times t'_l}$ for $l \in \{0 \dots L'\}$. L' and $\forall_l t'_l$ have to be chosen such that the following terms are well-defined, e.g. $l' = L$ and $\forall_l t'_l = T$*

$$i^{[l]}(t; (s_p^{[l]})_l) := (\alpha^{[l]})^t i^{[l]}(0) + \sum_{k=1}^{t} (\alpha^{[l]})^{t-k} \left(W^{[l]} s_p^{[l-1]}(k) + V^{[l]} s_p^{[l]}(k-1) \right), \tag{5}$$

$$p^{[l]}(t; (s_p^{[l]})_l) := (\beta^{[l]})^t u^{[l]}(0) + \sum_{k=1}^{t} (\beta^{[l]})^{t-k} \left(i^{[l]}(k; (s_p^{[l]})_l) + b^{[l]} \right) - \vartheta \sum_{k=1}^{t-1} (\beta^{[l]})^{t-k} s_p^{[l]}(k), \tag{6}$$

$$s^{[l]}(t; (s_p^{[l]})_l) := H(p^{[l]}(t; (s_p^{[l]})_l) - \vartheta 1_{n_l}) \tag{7}$$

$$u^{[l]}(t; (s_p^{[l]})_l) := (\beta^{[l]})^t u^{[l]}(0) + \sum_{k=1}^{t} (\beta^{[l]})^{t-k} \left(i^{[l]}(k; s_p) + b^{[l]} - \vartheta s_p^{[l]}(k) \right), \tag{8}$$

These formulas are equivalent to the previous recursive definitions, given $s_p^{[l]} = s^{[l]}$ for $l \in \{0 \dots L'\}$.

Proof. Let the time $1 \leq t \leq T$ be non-zero. By induction we immediately get

$$i^{[l]}(t) = (\alpha^{[l]})^t i^{[l]}(0) + \sum_{i=1}^{t} (\alpha^{[l]})^{t-i} \left(W^{[l]} s^{[l-1]}(i) + V^{[l]} s^{[l]}(i-1) \right)$$

By definition of $u^{[l]}$ and $p^{[l]}$ we further obtain

$$u^{[l]}(t) = \beta^{[l]} u^{[l]}(t-1) + i^{[l]}(t) + b^{[l]} - \vartheta s^{[l]}(t)$$

from which

$$u^{[l]}(t) = (\beta^{[l]})^t u^{[l]}(0) + \sum_{i=1}^{t} (\beta^{[l]})^{t-i} \left(i^{[l]}(i) + b^{[l]} - \vartheta s^{[l]}(i) \right)$$

follows by induction. By substituting $u^{[l]}$ in $p^{[l]}$ we further obtain:

$$p^{[l]}(t) = (\beta^{[l]})^t u^{[l]}(0) + \sum_{i=1}^{t} (\beta^{[l]})^{t-i} \left(i^{[l]}(i) + b^{[l]} \right) - \vartheta \sum_{i=1}^{t-1} (\beta^{[l]})^{t-i} s^{[l]}(i)$$

□

Let us now take a look at some simple examples:

Example 2.1. *Let $T, L \in \mathbb{N}$. Then there exists a d.t. LIF-SNN with $\forall_{t \in [T]} s^{[L]}(t) = s^{[0]}(t)$ for any $s^{[0]} \in \{0,1\}^{n_0 \times T}$.*

We can proof this by using constant width $n_l = n$, weights $W^{[l]} = I_n$, biases $b^{[l]} = 0$, initial membrane potential $u^{[l]}(0) = 0$, leaky term $\beta^{[l]} = 0$ and threshold $\vartheta^{[l]} = 1$ for all $l \in [L]$.

We then get by definition:

$$\begin{aligned} s^{[l]}(t) &= H(s^{[l-1]}(t) - 1_{n_l}) \;=\; s^{[l-1]}(t) \\ u^{[l]}(t) &= \quad s^{[l-1]}(t) - s^{[l]}(t) \qquad\quad = 0 \end{aligned}$$

Example 2.2. *Let $T, L \in \mathbb{N}$. Then there exists a d.t. LIF-SNN with $\forall_{t \in T} s^{[L]}(t) = \max_{t' \in [t-1]}(s^{[0]}(t'))$ for any $s^{[0]} \in \{0,1\}^{n_0 \times T}$: In this network an output neuron switches on when the corresponding input neurons fires and does not switch off later.*

We can proof this by using constant width $n_l = n$, weights $W^{[l]} = T \cdot I_n$, biases $b^{[l]} = 0$, initial membrane potential $u^{[l]}(0) = 0$, leaky term $\beta^{[l]} = 1$ and threshold $\vartheta^{[l]} = 1$ for all $l \in [L]$.

We then get by definition:

$$\begin{aligned} s^{[l]}(t) &= H(\;\; u^{[l]}(t-1) + T \cdot s^{[l-1]}(t) - 1_{n_l}) \\ u^{[l]}(t) &= \quad u^{[l]}(t-1) + T \cdot s^{[l-1]}(t) - s^{[l]}(t) \end{aligned}$$

By adding over all timesteps we obtain

$$\begin{aligned} s^{[l]}(t) &= H(T \sum_{i=1}^{t} s^{[l-1]}(i) - (\sum_{t=i}^{t-1} s^{[l]}(i)) + 1_{n_l})) \\ u^{[l]}(t) &= \sum_{i=1}^{t} (T \cdot s^{[l-1]}(i) - s^{[l]}(i)) \end{aligned}$$

Since $\sum_{i=1}^{t-1} s^{[l]}(i) + 1_{n_l} \leq T$, once there is any $t_0 \in [T]$ with $s_i^{[l-1]}(t_0) = 1$ for an i, we get $\forall_{t \geq t_0} s_i^{[l]}(t) = 1$. By induction over the layers we clearly get the required property.

3 Structure of computations in r. d.t. LIF-SNN s

This section concern itself with the approximation of continuous functions by d.t. LIF-SNN.

In [Nguyen et al., 2025] the following theorem was proved:

Theorem 3.1. *Let f be a continuous function on a compact set $\Omega \subset \mathbb{R}^{n_0}$. For all $\varepsilon > 0$, there exists a d.t. LIF-SNN Φ with direct encoding, membrane potential output, $L = 2$ and $T = 1$ such that*

$$\|R(\Phi) - f\|_\infty \leq \varepsilon$$

Moreover, if f is Γ-Lipschitz, then Φ can be chosen with width parameter $n = (n_1, n_2)$ given by

$$n_1 = \left(\max\left\{ \left\lceil \frac{\operatorname{diam}_\infty(\Omega)}{\varepsilon} \Gamma \right\rceil, 1 \right\} + 1 \right) n_0$$
$$n_2 = \max\left\{ \left\lceil \frac{\operatorname{diam}_\infty(\Omega)}{\varepsilon} \Gamma \right\rceil^{n_0}, 1 \right\}$$

where $\operatorname{diam}_\infty(\Omega) = \sup_{x,y\in\Omega} \|x - y\|_\infty$.

Proof. See [Nguyen et al., 2025]. □

We will now provide an alternative version of this theorem which uses an increased latency to reduce the number of neurons:

Theorem 3.2. *Let $f \in \mathcal{C}^1(\Omega, \mathbb{R}^m)$ be a continuously differentiable function on a compact set $\Omega \subset \mathbb{R}^n$. For all $\varepsilon > 0$, there exists a d.t. LIF-SNN Φ with direct encoding, membrane potential output, $L = 2$ and*

$$T = 2K$$
$$n_1 = n + 1$$
$$n_2 = K^n \cdot (m + 1)$$

such that

$$\|R(\Phi) - f\|_\infty \leq \varepsilon$$

Lemma 3.1. *Let $\Omega \subset \mathbb{R}^n$ be compact. Then there exists a half-open cube C with width $\operatorname{diam}_\infty(\Omega)$ such that $\Omega \subset \overline{C}$.*

Proof of Lemma 3.1.. We can first define $a := (\min_{x\in\Omega} x_i)_i$ since Ω is compact We further have $b := a + \operatorname{diam}_\infty(\Omega) \cdot \mathbb{1}_n$. We now have $C := [a, b)$. Suppose now a point $y \in \Omega \setminus \overline{C}$ exists. By definition of a, we have $a \leq y$. By definition of C we further get $y \not\leq b$, and therefore $\exists_i b_i < y_i$ by definition of C. But this means $\|a - y\|_\infty > \operatorname{diam}_\infty(\Omega)$. □

Lemma 3.2. *Let $f \in \mathcal{C}^1(\Omega, \mathbb{R}^m)$ be a continuously differentiable function on a compact set $\Omega \subset \mathbb{R}^n$ and $\varepsilon > 0$. Then there exists a half-open cube C with $\Omega \subset C$ that can be composed in*

$$K^n := \min_{\substack{\xi,\theta>0 \\ \xi\theta=\varepsilon}} \left\{ \left\lceil \frac{\operatorname{diam}_\infty(\Omega)}{\frac{2}{\sqrt{n}} \min(\delta(\xi), \theta)} \right\rceil \right\}^n$$

half-open subcubes $(C_i)_{i=1..K^n}$ such that $\Omega \subset C$ and linear functions $g_i : C_i \to \mathbb{R}^m$ exists, such that $\|f - g|_\Omega\|_\infty < \varepsilon$ for $g := \sum_{i=1}^m g_i \mathbb{1}_{C_i}$. Here $\delta(\varepsilon)$ is the modulus of uniform continuity of the total derivative dF.

Proof of Lemma 3.2.. By Lemma 3.1 we have a half-open cube C with width $\operatorname{diam}_\infty(\Omega)$ and $\Omega \subset \overline{C}$. Let further $\varepsilon > 0$ be given. Since Ω is compact and $f \in \mathcal{C}^1(\Omega, \mathbb{R}^m)$, dF is uniformly continuous on Ω. Let $\delta(\varepsilon)$ be dFs modulus of continuity. We will now partition C in

$$K^n := \min_{\substack{\xi,\theta>0 \\ \xi\theta=\varepsilon}} \left\{ \left\lceil \frac{\operatorname{diam}_\infty(\Omega)}{\frac{2}{\sqrt{n}}\min(\delta(\xi),\theta)} \right\rceil \right\}^n$$

smaller half-open cubes. There are ξ, θ such that the minimum in the definition of K^n is obtained, since we take it over the set of natural numbers. The subcubes have width $w := \frac{\operatorname{diam}_\infty(\Omega)}{K} \leq \frac{2}{\sqrt{n}}\min(\delta(\xi),\theta)$. Let us further define $g_i : C_i \to \mathbb{R}^m$ by $g_i(x) := f(c_i) + df_{c_i}(x - c_i)$ where c_i is the center of C_i. It suffices now to show $\left\| f|_{C_i} - g_i \right\|_\infty < \varepsilon$. Let $x \in C_i$. We get:

$$\begin{aligned}
\left\| f(x) - f(c_i) - df_{c_i}(x-c_i) \right\|_p &= \left\| f(x) - f(c_i) - df_{c_i}(x-c_i) \right\|_p \\
&= \left\| \int_0^1 df_{c_i+(x-c_i)t}(x-c_i)dt - df_{c_i}(x-c_i) \right\|_p \\
&\leq \int_0^1 \left\| df_{c_i+(x-c_i)t}(x-c_i) - df_{c_i}(x-c_i) \right\|_p dt \\
&= \int_0^1 \left\| (df_{c_i+(x-c_i)t} - df_{c_i})(x-c_i) \right\|_p dt \\
&\leq \int_0^1 \left\| (df_{c_i+(x-c_i)t} - df_{c_i}) \right\| \left\| x - c_i \right\|_p dt \\
&\leq \int_0^1 \xi \left\| x - c_i \right\|_p dt \\
&= \xi \left\| x - c_i \right\|_p \\
&\leq \xi\theta \\
&= \varepsilon
\end{aligned}$$

□

Proof of Theorem 3.2.. Let a continuously differentiable function $f \in \mathcal{C}^1(\Omega, \mathbb{R}^m)$ on a compact set $\Omega \subset \mathbb{R}^n$ be given. Let there further be a $\varepsilon > 0$. By Lemma 3.2 we have a half-open cube $C = [y, z)$ with $\Omega \subset C$ with composition K^n half-open subcubes $(C_i)_{i=1..K^n}$ and linear functions $g_i : C_i \to \mathbb{R}^m$, such that $\left\| f - g|_\Omega \right\|_\infty < \varepsilon\tau$ for a $g := \sum_{i=1}^m g_i \mathbb{1}_{C_i}$. We will now define a r. d.t. LIF-SNN Φ with direct input encoding and membrane-potential outputs such that $\left\| R(\Phi)|_\Omega - g \right\|_\infty < \varepsilon(1-\tau)$. Before anything else we shall set the following basic parameters: $i^{[l]}(0) = 0$, $\alpha^{[l]} = 0$ and $\beta^{[l]} = \vartheta^{[l]} = 1$. We obtain the simplified equations:

$$\begin{aligned}
p^{[l]}(t) &= u^{[l]}(t-1) + W^{[l]} s^{[l-1]}(t) + V^{[l]} s^{[l]}(t-1) + b^{[l]} \\
s^{[l]}(t) &= H(p^{[l]}(t) - 1_{n_l}) \\
u^{[l]}(t) &= p^{[l]}(t) - s^{[l]}(t)
\end{aligned}$$

The intuitive idea is the following… We define the i-th neuron of the first n of the first layer by parameters $w = \frac{1}{z_i - y_i} e_i$, $b = -\frac{y_i}{z_i - y_i}$, $v = -e_{n+1}$, $u_0 = 0$, $i_0 = 0$. We obtain

$$p_i^{[1]}(t) = u_i^{[1]}(t-1) + \frac{1}{z_i - y_i} s_i^{[0]}(t) - s_{n+1}^{[1]}(t-1) - \frac{y_i}{z_i - y_i}$$

and therefore $1 = s_i^{[1]}(t) \Leftrightarrow$

We further define the $n+1$-th neuron by parameters $w = 0$, $b = 1$, $v = e_{n+1}$, $u_0 = -(K-1)$, $i_0 = 0$.

Let us now regard the second layer. □

comparison: sin mit kleiner ampitude, extrem hoher frequenz

Theorem 3.3.

The proof works by first showing that a continuous function can be arbitrarily approximated by step functions, in particular by step functions constant on hypercubes in Ω. Then the d.t. LIF-SNN is constructed by using the first layer to cut the input space by hyperplanes along the cubes and using the second layer to represent the hypercubes.

Figure 3.1 shows a sinus function getting approximated in this way.

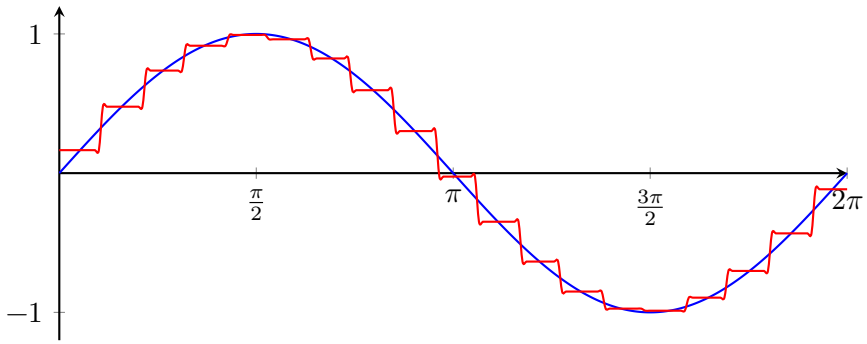

Figure 3.1: A r. d.t. LIF-SNN approximating a sinus wave

4 Complexity of input partitions

In the following we will analyze what shape the graph of a r. d.t. LIF-SNN has. Since the output of the following layers only depends on the spike trains of the first hidden layer, those layers are only able to merge different output regions of the first hidden layer. We will therefore only study the output landscape of the first hidden layer.

We can therefore simplify the notation in this section by writing $W, b, V, \beta, \vartheta, u, s$ for $W^{[1]}, b^{[1]}, V^{[1]}, \beta^{[1]}, \vartheta^{[1]}, u^{[1]}, s^{[1]}$ respectively. Since we are using direct encoding, we will write x for $s^{[0]} \in \mathbb{R}^{n_0}$:

$$i(t) = \alpha i(t-1) + Wx + Vs(t-1) \tag{9}$$
$$p(t) = \beta u(t-1) + i(t) + b \tag{10}$$
$$s(t) = H(p(t) - \vartheta 1_n) \tag{11}$$
$$u(t) = p(t) - \vartheta s(t) \tag{12}$$

By using the simplified notation, we obtain the following explicitely formulated definitions from Lemma 2.1 for $x \in \mathbb{R}^{n_0}$ and $s : \{0,1\}^{n_1 \times t'}$ for t' chosen such that the following equations are well-defined:

$$i(t; x; s_p) = \alpha^t i(0) + \sum_{k=1}^{t} \alpha^{t-k} \left(Wx + V s_p(k-1) \right), \tag{13}$$
$$u(t; x; s_p) = \beta^t u(0) + \sum_{k=1}^{t} \beta^{t-k} \left(i(k; x; s_p) + b - \vartheta s_p(k) \right), \tag{14}$$
$$p(t; x; s_p) = \beta^t u(0) + \sum_{k=1}^{t} \beta^{t-k} \left(i(k; x; s_p) + b \right) - \vartheta \sum_{k=1}^{t-1} \beta^{t-k} s_p(k), \tag{15}$$
$$s(t; x; s_p) = H(p(t; x; s_p) - \vartheta 1_{n_1}) \tag{16}$$

Remark 4.1. We will sometimes just write $i(t; x)$ instead of $i(t; x; s_p)$, with which we imply to use s (at x) for s_p. The same holds for u, p, s.

It is furthermore quite obvious that i grows linear in x, given fixed s_p. The same holds therefore for u and p. Further, the interdependence of the different components in $i/u/p/s$ is gone using a constant s_p. We can therefore split e.g. p into functions $p_{i;s_p}$ such that $p(t, x) = p_{1;s_p}(t; x_1) \times ... \times p_{n_1;s_p}(t; x_{n_1})$. The functions $p_{i;s_p}$ are not only linear in x, but also grow monotonically in x since $\alpha, \beta \geq 0$. So we in particular have that p to is growing monotonically (regarding by-component ordering). Since H is a monotonically growing function, s is also growing monotonically.

We will examine the graph by investigating the shape and location of the constant regions of a r. d.t. LIF-SNN :

Definition 4.1. The set of constant regions of a r. d.t. LIF-SNN Φ is defined as the partition

$$C_\Phi := \{R(\Phi)^{-1}(\{y\}) \mid y \in \mathrm{im}(R(\Phi))\}$$

of $\mathbb{R}^{n_0}$. The set of constant regions of the first layer of a r. d.t. LIF-SNN Φ is defined as

$$C_{\Phi,1} := \{\{x \in \mathbb{R}^{n_0} \mid \forall_t s(t; x) = s'(t)\} \mid s' \in \{0,1\}^{n_1 \times T}\}$$

Proposition 4.1. *The constant regions of the first layer of a r. d.t. LIF-SNN Φ with $W = I_{n_1}$ are half-open cuboids.*

Lemma 4.1. $\bigcap_{j \in J} \prod_{i \in [n]} M_{i,j} = \prod_{i \in [n]} \bigcap_{j \in J} M_{i,j}$ *for an index set J and sets $(M_{i,j})_{i \in [n], j \in J}$.*

Proof of Lemma 4.1. For every $x = (x_i)_{i\in[n]} \in M := \prod_{i\in[n]} \bigcup_{j\in J} M_{i,j}$ holds:

$$x \in \bigcap_{j\in J} \prod_{i\in[n]} M_{i,j} \Leftrightarrow \forall_{j\in J} x \in \prod_{i\in[n]} M_{i,j} \Leftrightarrow \forall_{j\in J} \forall_{i\in[n]} x_i \in M_{i,j}$$

On the other hand, we have

$$x \in \prod_{i\in[n]} \bigcap_{j\in I} M_{i,j} \Leftrightarrow \forall_{i\in[n]} x_i \bigcap_{j\in I} M_{i,j} \Leftrightarrow \forall_{i\in[n]} \forall_{j\in J} x_i \in M_{i,j}$$

In total we get

$$\bigcap_{j\in J} \prod_{i\in[n]} M_{i,j} = M \cap \bigcap_{j\in J} \prod_{i\in[n]} M_{i,j} = M \cap \prod_{i\in[n]} \bigcap_{j\in I} M_{i,j} = \prod_{i\in[n]} \bigcap_{j\in I} M_{i,j}$$

□

Lemma 4.2. *The intersection $C_1 \cap C_2$ of half-open cuboids $C_1, C_2 \subset \mathbb{R}^n$ is a half-open cuboid.*

Proof of Lemma 4.2. By definition of a half-open cuboid, we have half-open intervals $[c_{i,j}, d_{i,j})$ with $j \in [n]$ such that $C_i := \prod_{j\in[n]} [c_{i,j}, d_{i,j})$ for $i \in [2]$. We therefore have

$$C_1 \cap C_2 = \prod_{j\in[n]} ([c_{1,j}, d_{1,j}) \cap [c_{2,j}, d_{2,j})) = \prod_{j\in[n]} ([\max(c_{1,j}, c_{2,j}), \min(d_{1,j}, d_{2,j})))$$

by Lemma 4.1 □

Lemma 4.3. *Let $[c,d)$ be a half-open interval and $\varphi : \mathbb{R} \to \mathbb{R}$ be an affine linear function. Then $\varphi^{-1}([c,d))$ is a half-open interval.*

Proof. If φ is constant, then $\varphi^{-1}([c,d))$ is either $\mathbb{R} = [-\infty, \infty)$ or $\emptyset = [0,0)$. If φ is not constant, then $\varphi = ax + b$ with $a > 0$ and $\varphi^{-1}([c,d)) = [\frac{c-b}{a}, \frac{d-b}{a})$, if $c, d \in \mathbb{R}$; if $c = -\infty$ or $d = \infty$ the lower limit or upper limit of $\varphi^{-1}([c,d))$ is $-\infty$ or ∞ respectively. □

Proof of Proposition 4.1. Let $C \in C_{\Phi,1}$ be such a region and s_C the corresponding spike train. We then get

$$\begin{aligned}
C &= \bigcap_{i\in[n_1], t\in[T]} \{x \mid (s_C)_i(t) = s_i(t; x)\} \\
&= \bigcap_{i\in[n_1], t\in[T]} \{x \mid (s_C)_i(t) = s_i(t; x; s_C)\} \\
&= \left(\bigcap_{\substack{i\in[n_1], t\in[T] \\ (s_C)_i(t)=0}} \{x \mid p_i(t; x; s_C) < \vartheta\} \right) \bigcap \left(\bigcap_{\substack{i\in[n_1], t\in[T] \\ (s_C)_i(t)=1}} \{x \mid p_i(t; x; s_C) \geq \vartheta\} \right)
\end{aligned}$$

So C is an intersection of pre-images of half-open intervals regarding functions $p_i(t; \cdot; s_C)$. Since $p_i(t; x; s_C) = p_{i;s_C}(t; \pi_i(x); s_C)$ for all $x \in \mathbb{R}^{n_0}$, and since $p_{i;s_C}$ is linear and growing monotonically, we can apply Lemma 4.3. A pre-image of a half-open interval regarding $p_i(t; \cdot; s_C)$ is therefore a pre-image of a half-open interval regarding π_i, so a half-open cuboid.

We can now apply Lemma 4.2 and obtain that C is indeed a half-open cuboid.

□

We will add an ordering to spike trains based on lexicographical ordering. We define $s' \leq_l s''$ for $s', s'' \in \{0,1\}^{n_1 \times T}$ to mean that either $s' = s''$ or $s'(t) < s''(t)$ holds for the minimal time t such that the spike trains have different values.

Lemma 4.4. *Let us regard a r. d.t. LIF-SNN Φ with $W = I_{n_1}$. Let further $x, y \in \mathbb{R}^{n_0}$. We then have $x \leq y \Rightarrow s(\cdot; x) \leq s(\cdot; y)$.*

Proof. We first proof $x \leq y \Rightarrow s(\cdot; x) \leq s(\cdot; y)$. Suppose $s(\cdot; x) \neq s(\cdot; y)$. We then have a minimal $t \in [T]$ such $s(t; x) \neq s(t; y)$. Now due to $\forall_{t'<t} s(t'; x) = s(t'; y)$ and Remark 4.1 we have $s(t; x) \leq s(t; y)$ and therefore $s(t; x) < s(t; y)$. □

Lemma 4.5. *All (finite) vertices of the constant regions of the first layer of a r. d.t. LIF-SNN Φ with $W = I_{n_1}$ are contained in $[a, b]$, where*

$$a :=$$
$$b := \max(-\beta u(0) - \alpha i(0),) + (-b) + \vartheta \cdot \mathbb{1}_{n_1}$$

Proof. Let □

While in theory we would expect the number of constant regions to grow exponentially with time, it grows only quadratically.

Theorem 4.1. *A r. d.t. LIF-SNN with $W = I_{n_1}$, … has at a maximum $(\frac{T^2+T}{2} + 1)^n$ different constant regions.*

Proof. Since the number of constant regions of a r. d.t. LIF-SNN are just unions of the constant regions of the corresponding first layer, it suffices to compute the maximum number of constant regions of that layer.

Let us consider by how much the regions can increase going from $t-1$ to $t \in [T]$. We can categorize the regions at $t-1$ by the number of spikes they have in each component. □

Proof. □

5 Experimental results

Bibliography

References

[Nguyen et al., 2025] Nguyen, D. A., Araya, E., Fono, A., and Kutyniok, G. (2025). Time to spike? understanding the representational power of spiking neural networks in discrete time.