# Dendrites decrease the synaptic weight resolution necessary to compute

Romain Cazé, Marcel Stimberg

January 2020

## Abstract

In theory, neurons can compute all threshold functions, but in practice, synaptic weight resolution limits their computing capacity. Here, we study how dendrites alleviate this practical limitation by demonstrating a computation where dendrites considerably decrease the necessary synaptic weights resolution. We show how a biophysical neuron model with two passive dendrites and a soma that can implement this computation more efficiently than a point neuron. The latter requires synaptic weight orders of magnitudes larger than the others, the former implement the computation with equivalent synaptic weights. This work paves the way for a new generation of neuromorphic chips composed of dendritic neurons. These chips will require less space and less energy to function.

## Introduction

We learn in textbooks that neurons perform a weighted sum of their inputs which triggers or not an action potential. We call this model the linear threshold unit (LTU). In this model, neuron only computes threshold functions, but it has been proven that a network such neurons function as a universal approximator. This result provided the initial momentum to the field of artificial neural networks and encouraged the design neuromorphic chips.

Neuromorphic chips face, however, a challenge. Synapses often occupy the majority of the space in the chip, up to ten times more than the space occupied by neurons themselves [8].

A solution would be to use small precision synaptic weights, but neurons with limited precision weights cannot compute all threshold functions [4]. Living creatures evolved a strategy to relax this requirement.

Brains compute more efficiently than our best super-computer. Our brain occupies $cm^3$ whereas certain super computers occupy entire buildings. In the present study, we look at how dendrites enables to compute more function with low resolution synaptic weights.

The receptive organ of neurons' -dendrites- pushed the field of single neuron computation forward [6]. Certain dendrites behave non-linearly [1, 10, 5] turning neurons into more than LTU [9]. This new generation of unit can perform original computations like the feature binding problem [3].

Here we show that an LTU implementation of a function requires a synaptic weight many times higher than the other inputs. In an implementation using dendrites, however, all synaptic weights can be equal.

# Materials and methods

## A biophysical neuron model

We use a Hodgkin-Huxley type model because it has three interesting properties:

- The neuron-like temporal dimension enabling time-varying internal states.
- The model forms a whole, undivided into hermetic subunits.
- The sub-linearity comes for "free" with the synaptic implementation producing local sub-linearities.

A current $I_c$ charges the bilipid layer of a compartment following:

$$I_c = C_m \frac{dV}{dt} \tag{1}$$

Where $I_c$ stands for the current, $C_m$ is the membrane capacitance and $V$ is the voltage within the compartment.

The current $I_m$ flowing though the membrane can be expressed as:

$$I_m = g_L(V - E_L) + g_K(V - E_K) + g_{Na}(V - E_{Na}) \tag{2}$$

Where $V$ is the voltage within the compartment. $g_L$, $g_K$ and $g_{Na}$ stand for the leak, potassium and sodium conductances respectively. $E_L$, $E_K$ and $E_{Na}$ are reversal potentials.

Tab. 1 gives the values chosen for the equilibrium potentials.

Tab. 2 describes the channels' conductance.

This model is divided into compartments, and the neighbouring compartments send a current $I_a$.

The synaptic input current $I_s$ consists of:

$$I_s = g_s * (E_s - V) \tag{3}$$

with

$$\frac{dg_s}{dt} = -\frac{g_s}{\tau_s} \tag{4}$$

In total the current flowing within a compartment is:

| $E_L$ | $E_{Na}$ | $E_s$ | $E_K$ |
|---|---|---|---|
| -75 | 50 | 0 | -90 |

Table 1: Equilibrium potentials (in mV).

| $g_L$ | $g_{Na}^{max}$ | $g_K^{max}$ | $g_s$ | $\tau$ |
|---|---|---|---|---|
| $1.10^{-4}S/cm^2$ | 0 or $65.10^{-4}S/cm^2$ | 0 or $65.10^{-4}S/cm^2$ | 2e-8$S$ | $1ms$ |

Table 2: Conductance and time constant from the biophysical model.

$$I_c = I_m + I_a + I_s \tag{5}$$

Where $I_m$ is the current flowing through the membrane, $I_a$ is the current coming from the other compartments and $I_s$ is the synaptic current.

Importantly, we have a distinct $V$ for each compartments. Contrary to a point neuron, a biophysical model is a pluripotential structure.

Note that $g_{Na}^{max} = 0$ and $g_K^{max} = 0$ for the dendritic compartments to account for their passivity.

The three compartments model consists of a soma (diam=10$\mu$m) and two dendrites (length=400$\mu$m and diam=0.4$\mu$m). The two dendrites connect to the soma at their extremity.

At the junction all compartments have the same potential.

We use Brian 2 [11] software, working under Python, to simulate the biophysical model. You can find and use the code on the github repository associated with this publication named 20_01CaSt.

## Elementary neuron model and Boolean functions

We define a computation as a Boolean functions:

**Definition 1.** *A Boolean function of $m$ variables is a function on $\{0,1\}^m$ into $\{0,1\}$, where $m$ is a positive integer.*

Note that a Boolean function can also be seen as a classification.

A neuron modelled as a linear threshold unit (LTU) can only compute threshold functions/computations/classifications.

**Definition 2.** *$f$ is a threshold function of $m$ variables if and only if there exists at least a vector $w \in \mathbb{R}^m$ and a $\Theta \in \mathbb{R}$ such that:*

$$f(X) = \begin{cases} 1 & \text{if } w.X \geq \Theta \\ 0 & \text{otherwise} \end{cases}$$

We define here neuron models that take as inputs binary variables and activate or not. The standard neuron model is defined by a LTU:

**Definition 3.** *A LTU has a set of $m$ weights $w_i$ and a threshold $\Theta$ so that:*

$$f(X) = \begin{cases} 1 & \text{if } \sum w_i X_i \geq \Theta \\ 0 & \text{otherwise} \end{cases}$$

$w_i$ and $\Theta$ belong to a finite set of numbers depending on the implementation peculiarities and noise at which these value can be stabilised. This states the main differences with Def. 2. It means that a neuron may not be able to implement all threshold functions. For instance, a neuron with non-negative weights can only compute positive threshold functions.

From a neuroscience perspective, $m$ refers to the number of group of uncorrelated active presynaptic neurons. The synaptic weight refers to the depolarisation or hyperpolarisation caused by this afferent activity.

To account for saturations occurring in dendrites, we introduce the sub-linear threshold unit (SLTU):

**Definition 4.** *A SLTU with $d$ dendrites receiving $m$ inputs has a set of $d \times m$ weights $w_{i,j} \in \{0, 1\}$:*

$$f(X) = \begin{cases} 1 & \text{if } \sum_i E(\sum_j w_{i,j} X_{i,j}) \geq d \\ 0 & \text{otherwise} \end{cases}$$

*with*

$$E(Y) = \begin{cases} 1 & \text{if } Y \geq 1 \\ Y & \text{otherwise} \end{cases}$$

A STLU differs significantly from a network of LTUs. A network of LTUs requires multiple neurons and/or multiple compartments with voltage gated channels, a SLTU however can be implemented in a single neuron with a single, active compartment containing voltage gated channels –the soma– attached to passive dendrites.

Such a neuron model can compute all positive (see Def. 5 Boolean functions provided a sufficient number of dendrites and synapses [3].

**Definition 5.** *$\forall (X, Z) \in \{0, 1\}^n$ such that $X \geq Z$ (meaning that $\forall i : x_i \geq z_i$), $f$ is* **positive** *if and only if $f(X) \geq f(Z)$*

Interestingly, we are going to see that a SLTU enables to compute a threshold function/computation/classification with binary synaptic weights whereas a LTU requires high resolution synaptic weights.

# Results

## Implementations of a function requiring n-bits synaptic weights in a LTU

Fig. 1 shows an example of the threshold function.

Fig. 1A presents the truth table entirely defining the computation. Importantly, the neuron fires when two inputs activate depending on the inputs' identity.

We call this function the feature selective coincidence detection (FSC). This building block is the simplest computation after coincidence detection.

This function looks essential from an evolutionary perspective: to recognise a predator of a given colour from an harmless object of the same colour. Imagine a fish in the sea who would want to flee from a green or deep blue object that has the shape of a predator, yet this fish might not want to flee all the time as it is surrounded by a light blue environment.

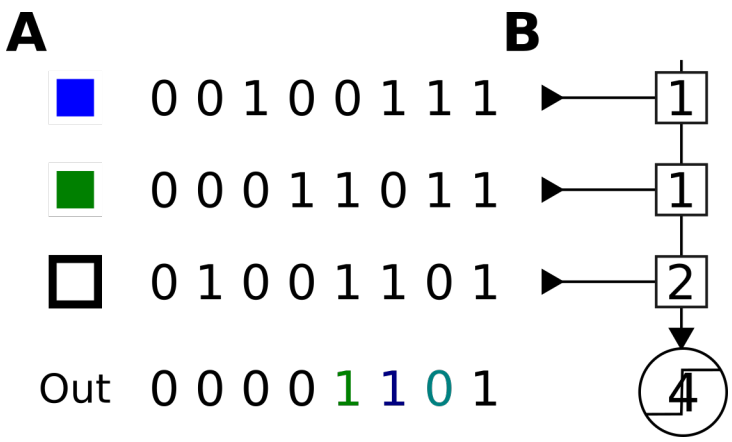


Figure 1: **The feature selective coincidence detection function and its implementation by a linear threshold unit (LTU).** A: A computation with three input variables. Note that the LTU fires only if $X_0$ is active with another specific input. It yields the coloured outputs (green and blue). The LTU remains however silent when $X_0$ is silent yielding the coloured 0. B: Its implementation in a LTU: small black squares stand for synapses (with the value of the synaptic weight within), and the weighted input sum passes through a Heaviside function with a given threshold (number in the circle).

Fig. 1B presents an implementation of this function by a linear threshold unit (LTU). This implementation uses the lowest possible synaptic weights resolution. One might want to use inhibitory synapses, but this would increase the necessary synaptic requirement.

Note here that the input $X_0$ has a synaptic weight twice as big as the others, and this difference will grow with the number of inputs.

This implementation uses a brute force strategy to compute the function: $X_0$ has a synaptic weight twice bigger than $X_1$ and $X_2$. If the threshold is sufficiently large, $X_1$ and $X_2$ active together won't produce a spike.

## Implementation of a threshold positive function with low resolution synaptic weights in a biophysical model

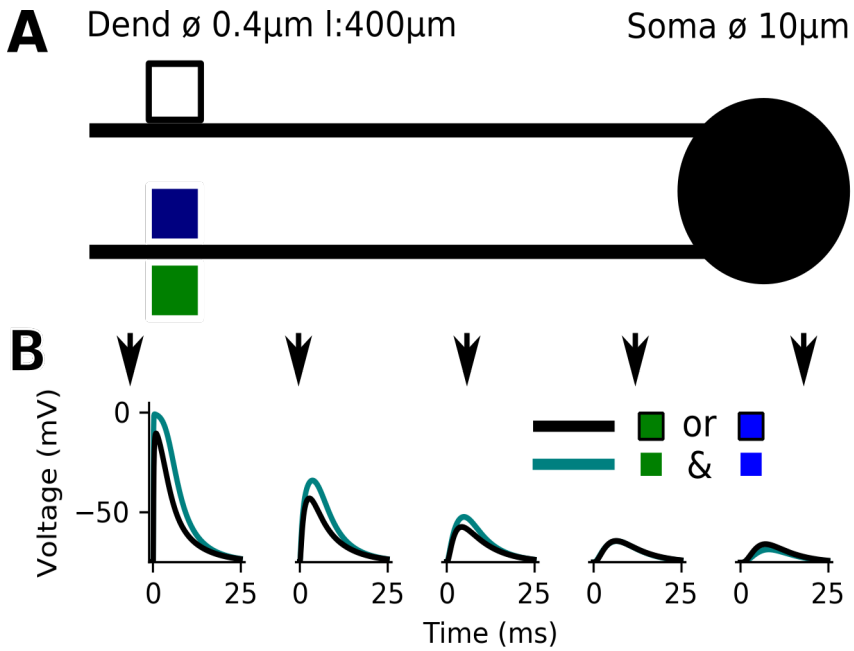


Figure 2: **A biophysical model sensitive to synapses' spatial distribution.** A: A biophysical model with two dendrites and a soma (lines:dendrites, circle:soma). Coloured squares depicts synapses. The model has three equivalent synapses. One comes from an input encoding the shape (black square), and two situated on the other dendrite encode the colour (blue or green). B: Membrane voltage traces responding to either clustered or dispersed synaptic activation (resp. aqua or black) at five distinct location (dendrite where two synapses cluster at 350 $\mu$m, 250 $\mu$m, 150 $\mu$m, 50 $\mu$m and soma). Note that at the point where two synapses cluster the clustered activation is larger than the scattered in the dispersed case the outcome changes.

We demonstrate in this section how a biophysical model can implement the threshold function described in the previous section, but this time with an equal synaptic weight for all inputs.

Fig. 2A presents a model with three synaptic inputs. One synapse comes from an input encoding for the shape of an object, and this synapse impinges on the first dendrite. The two others encode for two different colours, either green or blue, and they impinge on the second dendrite.

We studied the sub-threshold behaviour by setting $g_{Na}^{max} = 0$. We aim to

see here if the subthreshold membrane voltage enables to implement the feature selective coincidence detection function differently. Fig. 2B plots the voltage response at distinct locations in response to the activation of shape and colour (blue or green) synapses or to all the combination of the blue and green synapses. All the synapses, taken individually, produce the exact same depolarisation at the soma because we place them at the same distance (350 $\mu$m) and they all have the same conductance (20 nS). The depolarisation differs however in the two cases. At the tip of the second dendrite a dispersed synaptic activation shape + colour on two distinct dendrites) generates a depolarisation lower than the clustered synaptic activation (the two colours on the recorded dendrite). We make the opposite observation at the soma. We record a 9.3 mV depolarisation at the soma when the shape and a colour activate together, whereas we record a 6.2mV depolarisation when both colours activate at the same time.

This observation becomes self-evident when you take into account the synaptic driving force [6]. Indeed the synaptic current depends on the difference between the internal and external potential, as the number of excitatory inputs increases this driving force diminishes leading to sublinear summation.

A larger difference in depolarisation at the soma (a 100% instead of 50%) would have been expected if the compartments were fully independents. That is the reason why a rigorous simulation was necessary.

This mean that even in the case where all synapses have the same impact on the soma; even if we have a complete synaptic democracy [7] this does not change that the important factor is the position of synapses relative to each other.

To see if this model implement the FSC we submitted it to a series of input shown on top of Fig. 3. Each input activates exactly 25 times following a poisson law. We used a random activation to mimic in vivo neuronal inputs and display multiple possible scenarios.

We let neuron model fire by resetting $g_{Na}^{max} = 65e-4\ S/cm^2$. It is crucial to look at the supra-threshold behaviour as it is how the neuron communicate with the rest of the network. Moreover, back propagated action potential might indeed undermine the dendritic non-linearity disrupting the implementation [2].

The Fig. 2 displays the model's responses in four different situations:

- A single input activates, in this case the neuron remains silent. We obtain the same outcome whatever the chosen input.
- Two inputs activate (black + green or black + blue), in these two cases the neuron fires.
- The two clustered inputs activate, in the later case the neuron remains silent as expected from our observation in Fig. 2B.
- All inputs activate, in this last case the neuron does not overly fire notably because of the refractory period.

This figure thus presents the reaction of the neuron model in 5 over 8 cases, but for the case where no input activates the neuron obviously remains silent

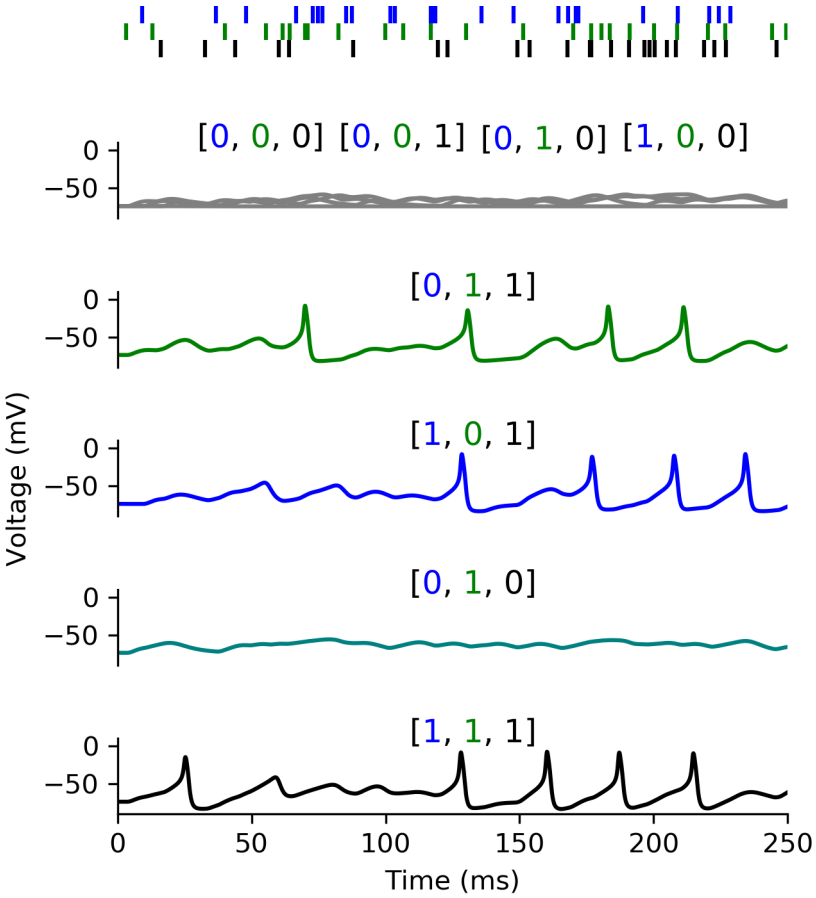


Figure 3: **A biophysical model implementing the feature selective coincidence detection.** Top: activity of three synapses, the two first synapses impinge on the same dendrite while the black one impinges on another. Bottom: Eight somatic membrane response depending on the active inputs. (gray: no synapse/only black/green/blue, green: black + green, blue: black + blue, aquamarine: green + blue, black: all inputs active). Note that we reproduced the truth table depicted in the first figure.

and the two other variants of the first case yield the same outcome.

All in all, the biophysical model implements the threshold function described in the previous result section using a different strategy. Each input has the same synaptic weight producing the same depolarisation at the soma. The biophysical model uses space to implement the FSC. The gating input (black) has its own

dendrite, while the two other inputs cluster on the same dendrite. Contrary to the LTU implementation, synaptic weights all stay equal. In the next section, we will enlarge this gap and scale the computation to an arbitrary $m$.

## Extension to an arbitrary number of input variables

We can extend the definition of the function to $m = n + 1$ variables with the following equation:

$$f(X) = \left( \bigvee_{i=1}^{n} X_i \right) \wedge X_0 \tag{6}$$

As you can observe this computation is very similar to a logic AND (the wedge notation in logic).

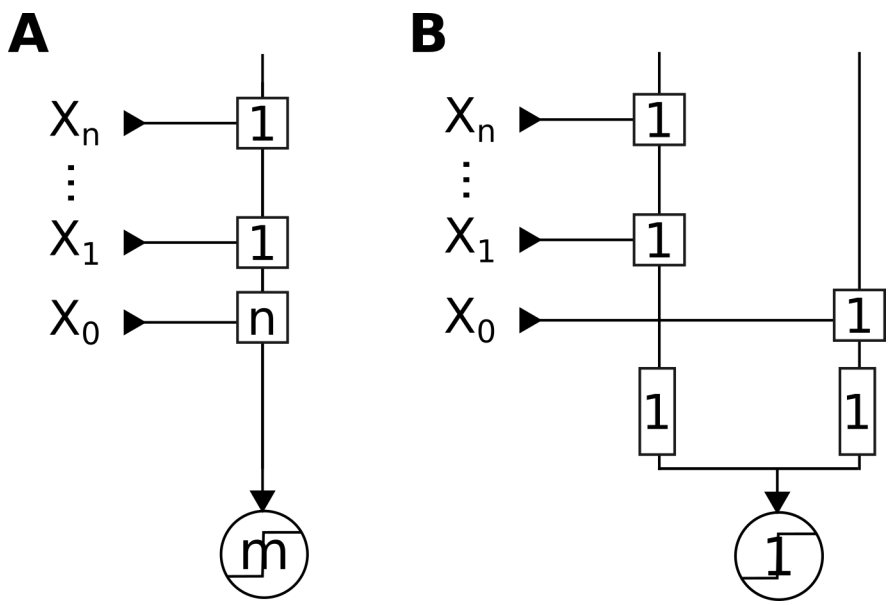


Figure 4: **Extending the implementation of the feature selective coincidence detection function to $m$ inputs** Synaptic weights are in squares, the somatic threshold is in a circle, and the saturation thresholds are in the two rectangles. A) Minimal implementation in a LTU. B) Implementation in a SLTU. Note that the implementation in a LTU requires a synaptic weight $n$ times bigger than in a SLTU.

On the one hand, Fig. 4A provides a constructive method to compute this classification with a LTU. Note that such a computation with $m$ variables requires that an input has a synaptic weight $n$ times bigger than the other inputs.

We can summarise this observation within a proposition.

**Proposition 1.** *To implement the FSC, a LTU requires that an input has a synaptic weight $n$ times larger than the others.*

*Proof.* The LTU must stay silent when $X_1, X_2, ..., X_n$ are active. Therefore $w_1 + w_2 + ... + w_n < \Theta$ if we pick a minimum resolution step for the synaptic weights $\alpha \geq 0$ we obtain from the previous inequality $n * \alpha \leq \Theta$, so $\Theta = n\alpha + \alpha$ at minimum because $\alpha$ is the minimal step. The neuron fires for every couple where $X_0 = 1$ and $X_i = 1$ so for instance $w_0 + w_1 \geq \Theta$, thus $w_0 + \alpha \geq \Theta$ therefore $w_0 + \alpha \geq n * \alpha + \alpha$ leading to $w_0 \geq n * \alpha$ □

Importantly, the proof assumes that synaptic weights are positive, because inhibitory synapses will only increase the synaptic requirements.

On the other hand, Fig. 4B provides a constructive proof that a SLTU can implement this computation with synaptic weights which are all equal.

# Discussion

In this work, we demonstrated how dendrites can alleviate the requirement in synaptic weights to implement threshold functions. We make two observations asserting this proposition:

- A LTU requires that an input has a synaptic weight $n$ times larger than the other inputs to implement a computation slightly more complex than an AND.
- A biophysical model with two dendrites can implement the same computation with all synaptic weights being equal.

The two first result sections make this observation for three inputs.

In the last section, we define the function for an arbitrary number of inputs. We prove in this section that a LTU needs a synaptic weight $n$ times bigger than the other inputs.

One might argue that a SLTU is nothing more than a network of LTUs. We focus, however, here on space efficiency and adding two passive dendrites waste less space than adding two whole neurons.

We could also say that even if dendrites alleviate the limitation they do not cancel it completely. For instance there are computations requiring an exponentially growing number of sublinear dendrites (e.g. a function described in [3]).

Multiple articles support, however, our results. Studies in computer science assert that even problems solvable by a LTU might not have a solution when weights have a limited precision [4]. Experimental studies in neuroscience demonstrate examples of sublinear summation in dendrites [12, 1], notably in interneurons.

This work advances works implying dendrites in computation. It shows that dendrites not only add new computing capacities but also improve the computing efficiency.

In conclusion, passive dendrites bring additional computing capacities for a modest spatial cost. Thus, passive dendrites should direct the design of a new generation of neuromorphic chips.

## Acknowledgements

This project started a long time ago we therefore first want to acknowledge B. Teleńczuk, PhD we also want to thank F. Danneville, PhD and C. Loyez, PhD for the stimulant discussions; Ms Marini-Audouard for the english proof-reading. Thank you Prof. Alain Cappy for his advices. A. Foust, PhD for correcting my English.

## References

[1] Therese Abrahamsson, Laurence Cathala, Ko Matsui, Ryuichi Shigemoto, and David A. DiGregorio. Thin Dendrites of Cerebellar Interneurons Confer Sublinear Synaptic Integration and a Gradient of Short-Term Plasticity. *Neuron*, 73(6):1159–1172, March 2012.

[2] Bardia F. Behabadi and Bartlett W. Mel. Mechanisms underlying subunit independence in pyramidal neuron dendrites. *Proceedings of the National Academy of Sciences*, 111(1):498–503, January 2014.

[3] Romain D. Cazé, Mark D. Humphries, and Boris S. Gutkin. Spiking and saturating dendrites differentially expand single neuron computation capacity. *Nips*, pages 1–9, 2012.

[4] Sorin Draghici. On the capabilities of neural networks using limited precision weights. *Neural networks : the official journal of the International Neural Network Society*, 15(3):395–414, April 2002.

[5] Albert Gidon, Timothy A. Zolnik, Pawel Fidzinski, Felix Bolduan, Athanasia Papoutsi, Panayiota Poirazi, Martin Holtkamp, Imre Vida, and Matthew Evan Larkum. Dendritic action potentials and computation in human layer 2/3 cortical neurons. *Science*, 87(January):83–87, 2020.

[6] Cristof Koch and Idan Segev. The role of single neurons in information processing. *Nature neuroscience*, 3 Suppl(suppl 1):1171–1177, November 2000.

[7] Jeffrey C. Magee and Erik P. Cook. Somatic EPSP amplitude is independent of synapse location in hippocampal pyramidal neurons. *Nature Neuroscience*, 3(9):895–903, September 2000.

[8] Thomas Pfeil, Tobias C. Potjans, Sven Schrader, Wiebke Potjans, Johannes Schemmel, Markus Diesmann, and Karlheinz Meier. Is a 4-Bit Synaptic

Weight Resolution Enough? Constraints on Enabling Spike-Timing Dependent Plasticity in Neuromorphic Hardware. *Frontiers in Neuroscience*, 6, 2012.

[9] Panayiota Poirazi, Terrence Brannon, and Bartlett W. Mel. Pyramidal Neuron as Two-Layer Neural Network. *Neuron*, 37(6):989–999, March 2003.

[10] Alon Polsky, Bartlett W. Mel, and Jackie Schiller. Computational subunits in thin dendrites of pyramidal cells. *Nature neuroscience*, 7(6):621–7, June 2004.

[11] Marcel Stimberg, Romain Brette, and Dan F M Goodman. Brian 2 , an intuitive and efficient neural simulator. *eLife*, pages 1–41, 2019.

[12] Alexandra Tzilivaki, George Kastellakis, and Panayiota Poirazi. Challenging the point neuron dogma: FS basket cells as 2-stage nonlinear integrators. *Nature Communications*, 10(1):1–14, August 2019.