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 equal 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, people call this model the linear threshold unit (LTU). A LTU can only implement certain computations, but it has been proven that a network of such units can implement any of them. This result provided the initial momentum to the field of artificial neural networks and encouraged the design neuromorphic chips based on this model.

These chips face, however, a practical challenge. Synapses often occupy the majority of the space in the chip, up to ten times more than the space occupied by neurons themselves [9]. A solution would be to use synaptic weights with a small number of stables states, but LTUs with such a limited precision weights can implement an even smaller number of computations than before [5]. We demonstrate here that neurons found a way to overcome this challenge.

Neurons possess a large receptive organ called dendrites, which do more than receiving, and provide new computing capacities [6]. Interestingly, all dendrites behave non-linearly [1, 11] turning all neurons into more than LTUs [10]. This new generation of neuron model can also perform computations impossible for a LTU like the feature binding problem [3].

We took a complementary perspective on dendrites here. Instead of searching for original computations, we searched for an original way to implement a computation reachable even by a LTU. In the present study, we look at how dendrites enables to compute with smaller means than for LTU - with synaptic weights taking a smaller number of distinct values than for a LTU.

Materials and methods

A biophysical neuron model

We performed simulations in a spatially extended neuron model, consisting a spherical soma (diam=10 µm) and two cylindrical dendrites (length=400 µm and diam=0.4 µm). The two dendrites are each divided into four compartments and connect to the soma at their extremity.

The membrane potential dynamics of the individual compartments follow the Hodgkin-Huxley formalism with:

$$C_m \frac{dV_{\text{soma}}}{dt} = g_L(V - E_L) + \bar{g}_K n^4 (V - E_K) + \bar{g}_{Na} m^3 h (V - E_{Na}) + I_a + I_s \quad (1)$$

for the somatic compartment and

$$C_m \frac{dV_{\text{dend}}}{dt} = g_L(V - E_L) + I_a + I_s \quad (2)$$

for the dendritic compartments.

Here, V_{soma} and V_{dend} are the respective membrane potentials, C_m is the membrane capacitance, g_L, $\bar{g}_K$, and $\bar{g}_{Na}$ stand for the leak, the maximum potassium and sodium conductances respectively, and E_L, E_K, and E_{Na} stand for the corresponding reversal potentials. The currents I_a represent the axial currents due to the membrane potential difference between connected compartments. The synaptic current I_s arises from a synapse placed at the respective compartment. It is described by

$$I_s = g_s(E_s - V) \quad (3)$$

with E_s being the synaptic reversal potential and g_s the synaptic conductance. This conductance jumps up instantaneously for each incoming spike and decays exponentially with time constant τ_s otherwise:

$$\frac{dg_s}{dt} = -\frac{g_s}{\tau_s} \quad (4)$$

The dynamics of the gating variables n, m, and h are adapted from [13] and omitted here for brevity.

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

Tab. 2 describes the channels' conductance.

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	τ
$10\,\mathrm{mS\,cm^{-2}}$	0 or $650\,\mathrm{mS\,cm^{-2}}$	0 or $650\,\mathrm{mS\,cm^{-2}}$	20 nS	1 mS

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

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

The synaptic input current I_s consists of:

$$I_s = g_s * (E_s - V) \tag{5}$$

with

$$\frac{dg_s}{dt} = -\frac{g_s}{\tau_s} \tag{6}$$

In total the current flowing within a compartment is:

$$I_c = I_m + I_a + I_s \tag{7}$$

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

Note that due to the absence of sodium and potassium channels in the dendrites, the dendrites are passive and cannot generate action potentials.

We use Brian 2 [12] software, working under Python, to simulate the biophysical model. You can find the code and an associated README on the github repository associated with this publication named 20_01CaSt. This code will allow to reproduce figure similar to Fig. 2 and Fig. 3 for two types of models: either a Hodgkin-Huxley model or a passive model with a leaky integrate fire mechanism to enable spiking.

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 $w \in \mathbb{R}^m$, a vector $X \in \{0,1\}^n$ 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 Θ so that:*

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

w_i and Θ 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.

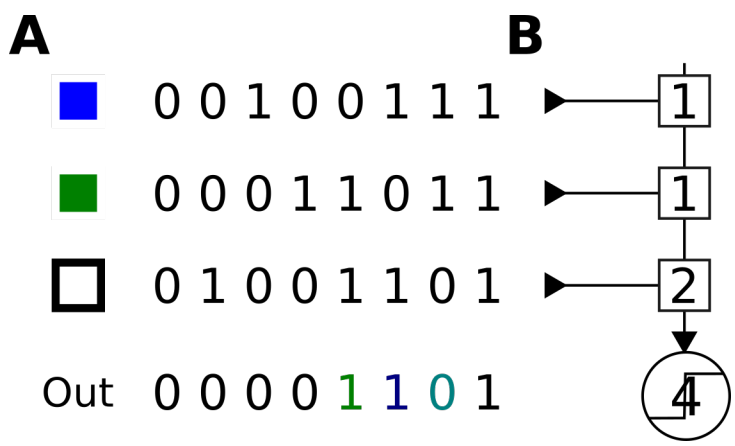

Figure 1: **The feature selective coincidence detection function and its implementation by a linear threshold unit (LTU).** A: A computation with three input variables. Note that the neuron must only fire if the last input (encoding the shape) is active with another specific input. It yields the coloured outputs (green and blue). The neuron stays however silent when the last input equates 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). We employed positive integer valued synaptic weights to reflect the finite precision. Even if synaptic weights can take real values, a finite precision means a finite number of values, and each integer represent the index of this

value. This implementation uses the lowest possible integer valued synaptic weights. This means that it requires the lowest resolution synaptic weights possible. 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 (see last part for proof).

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

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 µm) 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 (the two synapses activate on distinct dendrites) synaptic activation shape + colour1/2 generates a depolarisation lower than the clustered (the two synapses activate on same dendrite) synaptic activation colour1 + colour2. 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 6.2 mV depolarisation when both colours activate at the same time.

This observation becomes self-evident when you take into account the synaptic driving force [7]. 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 [8] this does not change that the important factor is the position of synapses relative to each other.

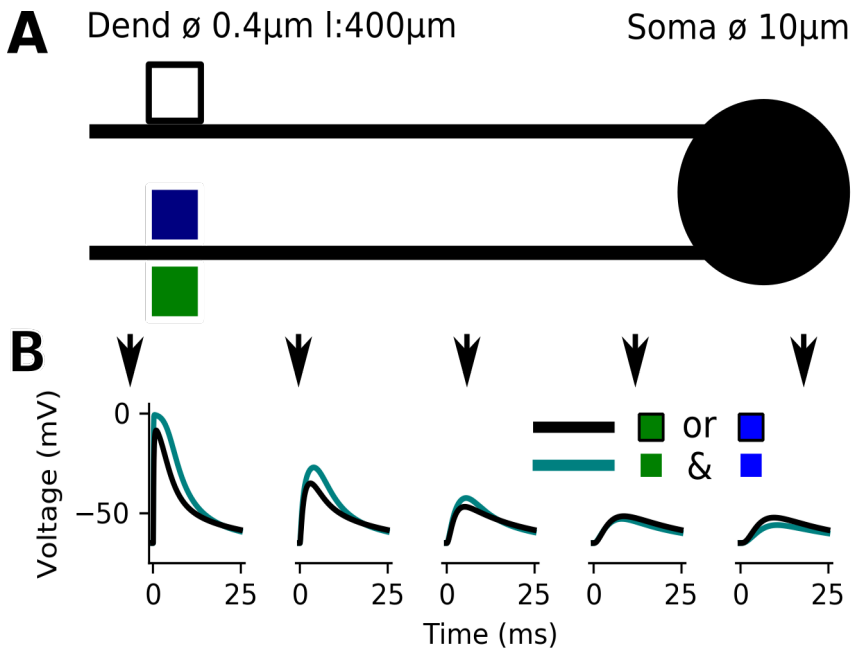

Figure 2: **A biophysical model sensitive to synapses' spatial distribution.** A: A biophysical model with two dendrites and a soma (lines:dendrites, circle:soma). Coloured squares depicts synapses. The model has three equivalent synapses. One comes from an input encoding the shape (black square), and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

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 $g_{Na}^{max} = 650\,\mathrm{mS\,cm^{-2}}$. 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

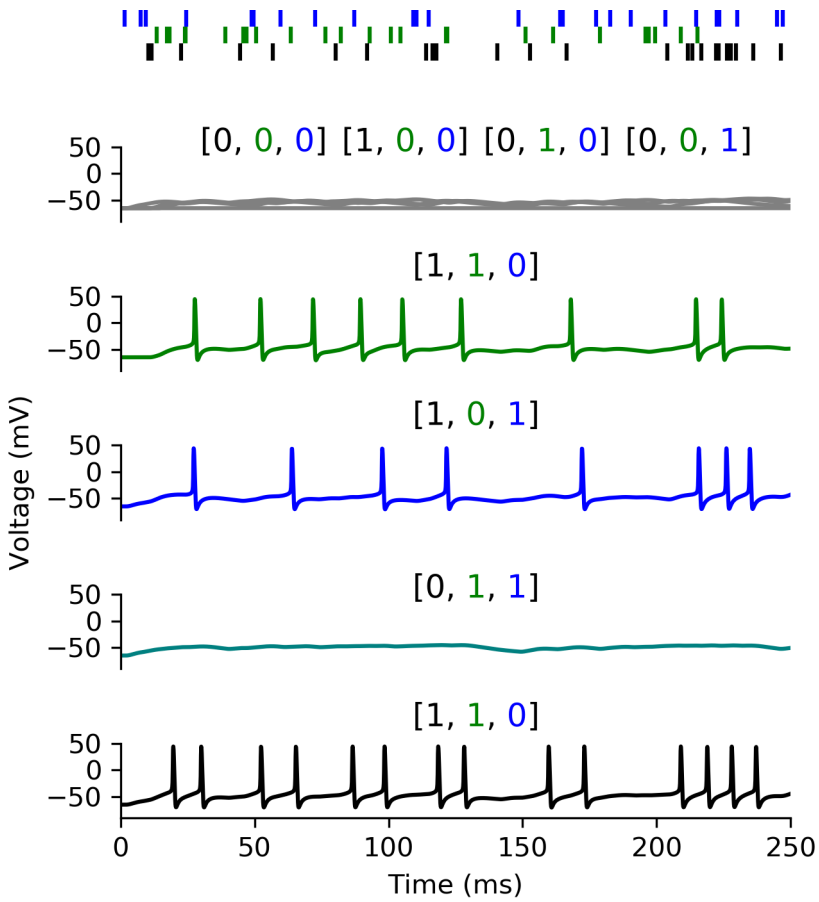

Figure 3: **A biophysical model implementing the feature selective coincidence detection.** Top: activity of three synapses, the two first synapses impinge on the same dendrite while the black one impinges on another. Bottom: Eight somatic membrane response depending on the active inputs. (gray: no synapse/only 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.

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 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:

$$f(X) = \left(\bigvee_{i=1}^{n} X_i \right) \wedge X_0 \tag{8}$$

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

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 at least n times bigger than the other inputs. Note also the massive growth of the threshold in a LTU.

We can summarise these observations in two propositions.

Proposition 1. *To implement the FSC with m variables, a LTU requires that an input has a threshold with at least m distinct states.*

Proof. The LTU must stay silent whatever $X_1, X_2, ..., X_n$ are active. Therefore $w_1 < \Theta$, $w_1 + w_2 < \Theta$ up to $w_1 + w_2 + ... + w_n < \Theta$ thus Θ must have at least $n + 1 = m$ distinct states. □

Proposition 2. *To implement the FSC, a LTU requires that an input has a synaptic weight that can take at least n times more possible states than the others.*

Proof. We saw that for a LTU to implement the FSC we need that at minimum $\Theta = n + 1$, and that the neuron fires for every couple where $X_0 = 1$ and $X_i = 1$ so for instance $w_0 + w_1 \geq \Theta$. Therefore at minimum $w_0 + 1 \geq n + 1$. It means that w_0 can at least have n states. □

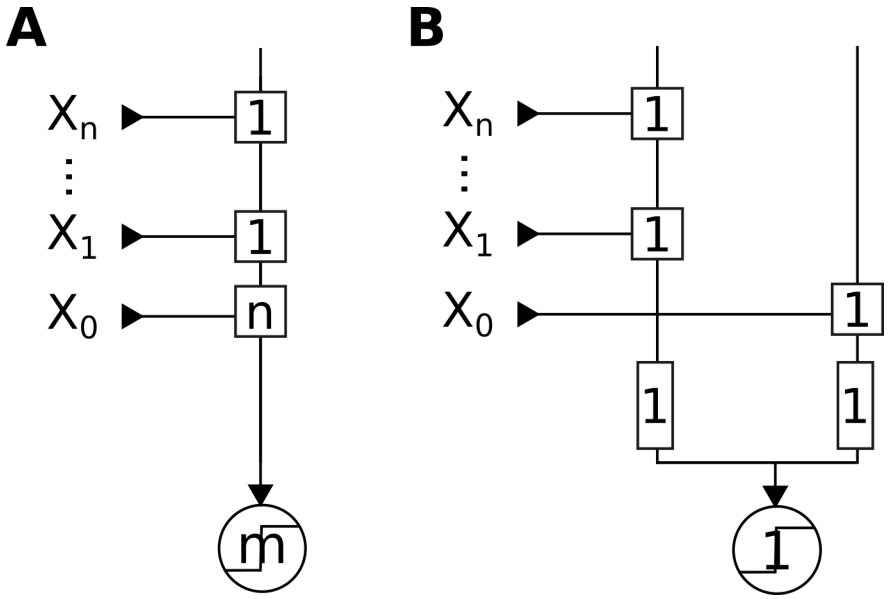

Figure 4: **Extending the implementation of the feature selective coincidence detection function to m inputs** Synaptic weights are in squares, the somatic threshold is in a circle, and the saturation thresholds are in the two rectangles. A) Minimal implementation in a LTU. B) Implementation in a SLTU.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

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 the present work, we demonstrate a computation for which an LTU implementation requires a synaptic weight many times higher than the other. In an implementation using dendrites, however, all synaptic weights can remain equal.

Two observations support this conclusion:

- 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 compu-

tation with all synaptic weights being equal.

The two first result sections explicit this observation for three inputs. In the last section, we define the function for an arbitrary number of inputs, and we prove that a LTU needs a synaptic weight with n times more distinct values than the other.

One might argue that a SLTU is nothing more than a network of LTUs. We focus, however, here on efficiency and adding two passive dendrites waste less space than adding two LTUs in an artificial neural network or in a neuromorphic chip.

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 [5]. Experimental studies in neuroscience demonstrate examples of sublinear summation in dendrites [14, 1], notably in interneurons.

We thoroughly reports here an example of where synaptic placement can replace a wide variation in weights. Our results relate to another study [4] where we also employ synaptic placement to compute. It shows that dendrites not only add new computing capacities, but also improve the computing efficiency. They provide new ways to compute.

In conclusion, dendrites bring additional computing capacities for a modest spatial cost that can direct the design of new neural networks and compact neuromorphic chips.

Acknowledgements

This project started a long time ago within a team directed by Alain Destexhe, PhD, it stems from multiple discussions with B. Teleńczuk, PhD. We also want to thank F. Danneville, PhD and C. Loyez, PhD for the stimulant discussions. Thank you Prof. A. Cappy, PhD for his remarks on the method. We also want to thank M., Humphries, PhD who provided numerous valuable comments. Finally we want to thank A. Foust, PhD for making our scientific English easier to read and Ms Marini-Audouard for the English proof-reading.

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] Romain D. Cazé, Sarah Jarvis, Amanda Joy Foust, and Simon R. Schultz. Dendrites Enable a Robust Mechanism for neuronal stimulus selectivity. *Neural Computation*, 29(1-17), 2017.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

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

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

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

[13] Roger D. Traub and Richard Miles. *Neuronal Networks of the Hippocampus*. Cambridge University Press, May 1991.

[14] 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.