
Simulating the Izhikevich spiking neuron model using the Brian2 software

Julen Etxaniz and Ibon Urbina

Abstract

In this document we report our proposal for the Izhikevich spiking neuron model representation in Brian2 Python's library, which is a simulator for spiking neural networks. We manage to successfully implement most of the proposed neuron types and features. We also make a simulation with a network of 1000 neurons. The original code was proposed by the author in the MATLAB programming language, but taking into account the importance Python programming language is taking in Machine Learning task implementations, we believe that it is convenient to also have an implementation with Python.

Contents

1	Description of the problem	2
2	Understanding the Izhikevich spiking neuron model	2
2.1	A new paradigm	2
2.2	Biological neurons	3
2.3	Biological neurons and spiking models	4
2.4	Why Izhikevich spiking neuron model?	4
2.5	Cortical neurons	4
2.6	Neuro-computational features and information processing	4
2.7	Neuron Types	7
2.8	The Izhikevich model	9
3	Implementation	9
3.1	Importing the libraries	10
3.2	Defining the model	10
3.3	Interacting with the model	10
3.4	Neuron Types	10
3.5	Neuron Features	10
3.6	Defining the simulation	10
3.7	Running the simulation	10
4	Results	11

4.1 Neuron Types	11
4.2 Neuron Features	12
4.3 Simulation	12
5 Conclusions	12

1 Description of the problem

The study of spike-timing dynamics in the brain is of interest to neuroscientists and the artificial network community. Investigation of the relative timing of spikes of multiple neurons [1] and the role they play in temporal coding in the brain is an important issue. Similarly, spiking networks can serve as powerful supervised learning and memory representation tools with several potential applications in the machine learning domain [2].

The Izhikevich spiking neuron model [3] [4] is a two-dimensional system of ordinary differential equations. In the equations shown in Figure 1, variable v represents the membrane potential of the neuron and u represents a membrane recovery variable, which accounts for the activation of K^+ ionic currents and the inactivation of Na ionic currents, and provides negative feedback to v .

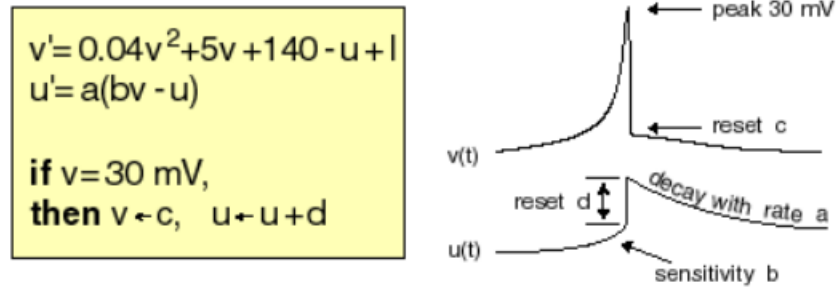


Figure 1: Izhikevich's spiking neuron model. Electronic version of the figure and reproduction permissions are freely available at www.izhikevich.com

The goal of the project is to implement the Izhikevich's model using the Brian2 Python library <https://brian2.readthedocs.io/en/stable/>. Then, test that for different parameters of the model it is possible to reproduce the spiking patterns described in [3] [4].

2 Understanding the Izhikevich spiking neuron model

2.1 A new paradigm

The fact that recent physiological experiments demonstrate that neural code and thus, neural information, is founded on the time of individual action potentials (individual spikes) changes the governing paradigm over the last years regarding to neural encoding: neural information is encoded in the firing rate of neurons.

This new paradigm does not exclude the previous one, but yes, give us a new perspective. Moreover, it has carried out a new class of neural models called Spiking Neural Networks (SNN).

Due to the fact that these type of networks reproduce our biological neurons better than other artificial neural networks (ANN), not only do they provide powerful tools for analysis of elementary processes in the brain, but also are able to process substantial amount of data using relatively small number of spikes. Furthermore, it has been scientifically demonstrated that they are capable of resolving all problems solved by non-spiking neural networks plus the ones that those are not capable of.

2.2 Biological neurons

The neuron is the basic building block of the brain and central nervous system. Moreover, they are specialized cells that transmit chemical and electrical signals. Thus, in essence, they are cells with common cell components (such as nucleus, organelles, cytoplasm, mitochondria, etc); but, with exclusive structures for receiving and sending the electrical signals that make neuronal communication possible. Which are these special structures? The most important ones, behind:

- **Dendrite:** branch-like structures extending away from the cell body that receive messages from other neurons and allow them to travel to the cell body.
- **Axon:** tube-like structure that carries an electrical impulse from the cell body to the opposite end of the neuron, the axon terminals, which pass the impulse to another neuron.
- **Synapse:** chemical junction between the axon terminals of one neuron and the dendrites of the next.

As common cells, neurons are surrounded by a cell membrane which completely separates the cell from its external environment. In neurons, this is the membrane that allows positive or negative ions to flow into and out of it. The inside of the cell is usually more negative than the outside; scientific studies talk about a negative difference of 70 *mv* with respect to the outside. This difference is known as the membrane potential, so, it defines the difference in electric potential between the interior and the exterior of the neuron. This term is not a static variable and changes constantly depending on the inputs coming from the axons of other neurons or from the exterior: some inputs make the membrane potential being more positive (less negative) or less positive (more negative).

But, how do biological neurons communicate with each other?

As we mentioned above, depending on the input a neuron receives, its membrane potential can be modified. This occurs because when an action potential reaches the presynaptic terminal, that are the axon terminals of the neuron, different neurotransmitters (chemicals) are released to the postsynaptic dendrite. Depending on what chemical is released, positive or negative ions will be attached to the membrane potential, and therefore, the potential will be modified. When the membrane potential reaches a potential value called action potential threshold, an action potential (or spike) is released; sometimes it is said that a neuron has fired a spike.

Let's resume the communication process:

- An electrical input reaches the presynaptic neuron.
- The electrical signal is converted into a chemical signal in the form of neurotransmitter release.
- Postsynaptic neuron receptors catch the chemicals released by the presynaptic neuron.
- Postsynaptic neuron membrane potential is modified due to the ion flow of the received neurotransmitters.

No matter what type of signal is transmitted (chemical or electric) but both need a stimuli (electrical activity) in the presynaptic neuron. This stimuli is usually given in two ways: by an external current input (in scientific experiments for example) or as a consequence of another presynaptic current induction.

To finish with this brief explanation of how biological neurons are and behave and to understand future concepts, it is convenient to introduce what excitatory and inhibitory neurotransmitters are. For now, let's refer to neurotransmitters as the body's chemical messengers. As mentioned before, they are the molecules used by the nervous system to transmit messages between neurons, or from neurons to muscles. A neurotransmitter can influence a neuron in two ways:

- **Excitatory:** a neurotransmitter that promotes the generation of the electrical signal (action potential) in the receiving neuron.
- **Inhibitory:** a neurotransmitter that prevents the generation of the electrical signal (action potential) in the receiving neuron.

2.3 Biological neurons and spiking models

3 properties are shared in both biological and spiking models:

- **Output signals:** The coming information is delivered via many inputs, but a single spiking signal is produced as an output.
- **Firing:** The probability of generating a spike is increased by excitatory inputs and decreased by inhibitory inputs.
- **Dynamic:** The spiking flow or spiking generation is characterized by at least one state variable. When the internal variables of the model reach a certain state (a certain threshold), the model generates one or more spikes (a burst of spikes).

2.4 Why Izhikevich spiking neuron model?

Neuroscience, the scientific study of the nervous system which goal is to understand the fundamental and emergent properties of neurons and neural circuits, not only requires experimental studies with animals and humans, but also models that are able to simulate with numerical means a large-scale of brain models.

However, until the emergence of the Izhikevich model, it was not easy to find an adequate balance between good biological neurons representation and a low computational cost. For instance, coexisted models with very rich biological representation but computationally prohibitive (Hodgkin-Huxley-type model) and simple and unrealistic biological neuron characterization but computationally exceptional.

Two were the requirements demanded to artificial brain models:

- Computationally simple.
- Capable of producing rich firing patterns exhibited by real biological neurons.

That's why the arrival of the Izhikevich model has been a fascinating improvement in this field. Apart from being biologically as plausible as the Hodgkin-Huxley model, it is also as computationally efficient as the integrate-and-fire model. For instance, changing four parameters of the model, one can reproduce the spiking and bursting dynamics of known types of cortical neurons.

2.5 Cortical neurons

Cortical neurons correspond to the ones living in the cerebral cortex; which is the outer layer of the neural tissue of the brain in humans and other mammals. This cortex is the largest site of neural integration in the central nervous system and plays a key role in attention, perception, awareness, thought, memory, language and consciousness.

The cerebrum, the largest part of the brain containing the cerebral cortex and the uppermost region of the central nervous system, is the most highly developed part of the human brain and is responsible for thinking, perceiving, producing and understanding language. Moreover, it is directly or indirectly involved in several functions of the body, such as determining intelligence, determining personality, motor functions, planning and organization, touch sensation, processing sensory information, language processing...

As we may notice, most information processing occurs in the cerebral cortex. Thus, we may make an idea of the importance of analyzing and representing the cortical neurons in an artificial way in order to reproduce human processing and learning. As mentioned before, it is exactly the type of neurons the Izhikevich model can simulate. Furthermore, it can also simulate neural features and behaviours mentioned in the next section.

2.6 Neuro-computational features and information processing

Neurons are not limited to a single spike or action potential; they may release more than one and with different patterns depending on two variables: time and input current. Analyzing this firing-

behaviour has an utmost importance regarding to the new paradigm: spike-timing information processing.

Below, 20 of the most prominent features of biological spiking neurons mentioned in [4]. See Figure 2.

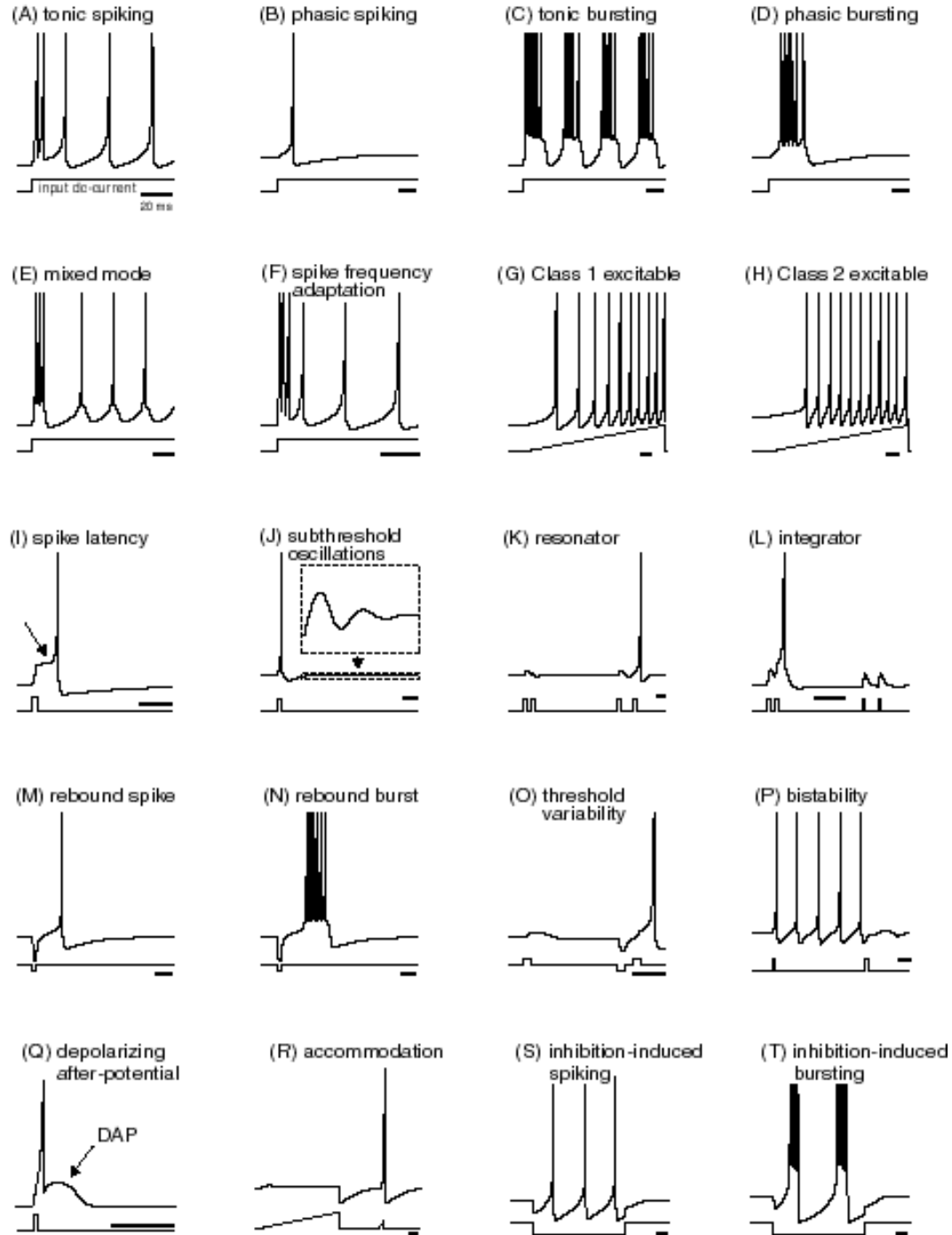


Figure 2: Electronic version of the figure and reproduction permissions are freely available at www.izhikevich.com

- **Tonic Spiking:**

- Neurons are quiescent (inactive) until they are stimulated.
- When the input is on, the neuron continues to fire a train of spikes.
- Information processing: can indicate that there is a persistent input.
- **Phasic Spiking:**
 - Neurons fire a single spike at the onset of the input and remain quiescent afterwards.
 - Information processing: can detect the beginning of the stimulation.
- **Tonic Bursting:**
 - Neurons fire periodic bursts of spikes when stimulated.
 - Contribute to the gamma-frequency oscillations in the brain.
- **Phasic Bursting:**
 - Neurons fire phasic bursts of spikes at the onset of the input.
 - Information processing: can report the beginning of the stimulation.
 - Information processing: can recognize the channel of communication between neurons because the interspike frequency within the bursts encodes that channel.
- **Mixed Model (Bursting Then Spiking):**
 - Neurons fire a phasic burst at the onset of the stimulation and then switch to the tonic spiking mode.
 - Information processing: can detect the onset and report the extent of the stimulation.
- **Spike Frequency Adaptation:**
 - Neurons fire tonic spikes with decreasing frequency.
 - The frequency is relatively high at the onset of stimulation, and then it adapts.
 - Information processing: can encode the time elapsed since the onset of the input thanks to the interspike frequency.
- **Class 1 Excitatory:**
 - The frequency of tonic spiking depends on the strength of the input.
 - Neurons fire low-frequency spikes when the input is weak.
 - Information processing: can encode the strength of the input.
- **Class 2 Excitability:**
 - Neurons are either quiescent or fire a train of spikes with a certain relatively large frequency.
 - Cannot fire low-frequency spike trains.
 - Their firing rate is a poor predictor of the strength of stimulation.
- **Spike Latency:**
 - Neurons fire spikes with a delay that depends on the strength of the input signal.
 - For weak but high-threshold input, the delay (spike latency) can be quite large.
 - Information processing: can encode the strength of the input thanks to the latencies.
- **Subthreshold Oscillations:**
 - Neurons capable of exhibiting oscillatory potentials.
 - Oscillation frequencies play an important role and act as bandpass filters.
- **Frequency Preference and Resonance:**
 - Neurons having oscillatory potentials can respond selectively to the inputs having frequency content similar to the frequency of subthreshold oscillations.
- **Integration and Coincidence Detection:**
 - Neurons without oscillatory potentials.
 - The higher the frequency of the input, the more likely they fire.
 - Information processing: can be useful for detecting coincident or nearly coincident spikes.
- **Rebound Spike:**

- When a neuron receives and then is released from an inhibitory input, it may fire a post-inhibitory spike.
- **Rebound Burst:**
 - When a neuron receives and then is released from an inhibitory input, it may fire post-inhibitory bursts.
- **Threshold Variability:**
 - Depending on the prior activity of the neurons, biological neurons have a variable threshold.
- **Bistability of Resting and Spiking States:**
 - Neurons that exhibit two stable modes of operation: resting and tonic or bursting spiking.
 - To switch from the tonic spiking to resting mode, the input must arrive at an appropriate phase of the oscillation.
 - Information processing: can create a short-term memory.
- **Depolarizing After-Potentials:**
 - After firing a spike, the membrane potential of a neuron may exhibit a prolonged after-hyperpolarization or a prolonged depolarized after-potential.
- **Accommodation:**
 - Neurons are extremely sensitive to brief coincident inputs, but may not fire in response to a strong but slowly increasing input.
- **Inhibition-Induced Spiking:**
 - Neurons are quiescent (inactive) when there is no input, but fire when hyperpolarized by an inhibitory input or an injected current.
- **Inhibition-Induced Bursting:**
 - Neurons fire tonic bursts of spikes in response to a prolonged hyperpolarization.
 - It is believed that this bursting plays an important role in sleep rhythms.

2.7 Neuron Types

Before we start analyzing and understanding the long-awaited Izhikevich model, it is convenient to have a subtle idea of the different cortical neurons we can encounter with according to [3]. See Figure 3.

In the case of the neocortical neurons in the mammalian brain, depending on the pattern of spiking and bursting, excitatory cells are divided into three classes:

- **Regular spiking (RS):**
 - The most typical neurons in the cortex.
 - When a prolonged stimulus is injected, the neurons fire a few spikes with short inter-spike period and then the period increases.
- **Intrinsically bursting (IB):**
 - Fire a stereotypical burst of spikes followed by repetitive single spikes.
- **Chattering (CH):**
 - The stereotypical bursts of closely spaced spikes.

In addition, depending also on spiking and bursting patterns, inhibitory cells are divided in two classes:

- **Fast spiking (FS):**
 - Fire periodic trains of action potentials with extremely high frequency practically without any adaptation; that is, without practically a slowing down in the frequency.

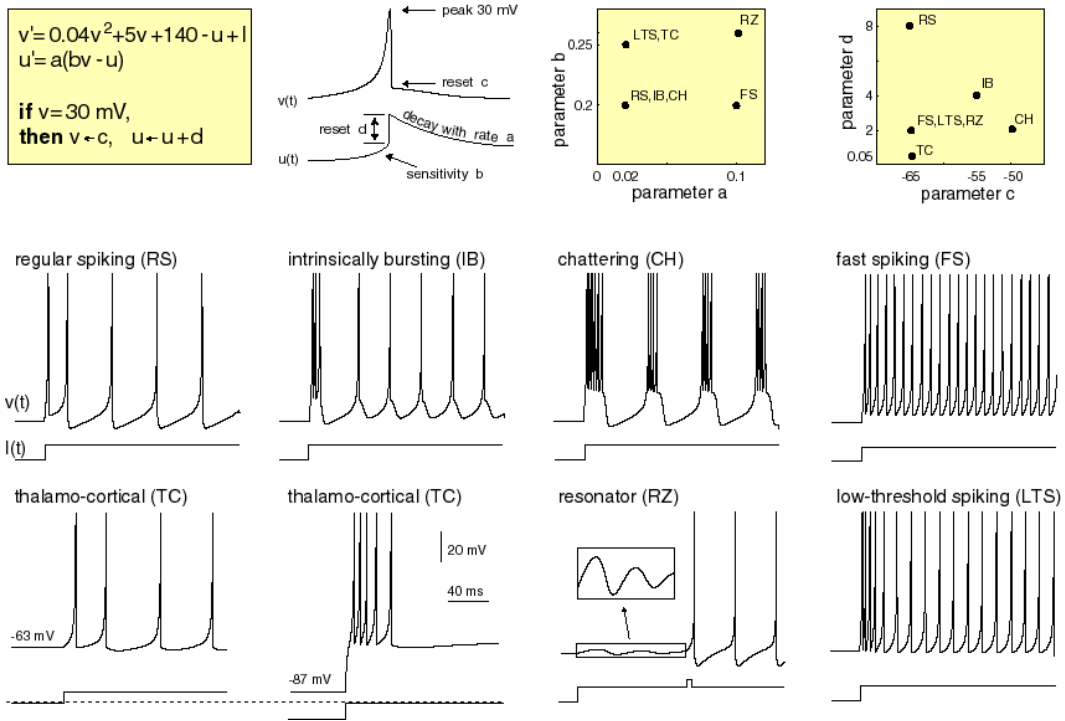


Figure 3: Neuron Types. Electronic version of the figure and reproduction permissions are freely available at www.izhikevich.com

- **Low-threshold spiking (LTS):**

- Fire periodic trains of action potentials with extremely high frequency with noticeable spike frequency adaptation; that is, with a noticeable slowing down in the frequency.

However, this model is not only limited to representing cerebral cortex neurons. For instance, it can reproduce thalamo-cortical cells; those that convey sensory information to the neocortex via thalamo-cortical synapses. We have to take into account that the thalamus is the door which introduces cells to the neocortex. Such is the importance of the thalamus, that every single sensory information except the olfactory one is processed in the neocortex thanks to it. How that sensory information reaches the thalamus is out of the scope.

Two are the possible dynamics of the thalamo-cortical cells depending on the firing regimes:

- **At rest v is around -60 mV and then depolarized:**

- They exhibit tonic firing; which refers to a sustained response that activates during the course of the stimulus.

- **A negative current step is delivered and the membrane potential is hyperpolarized:**

- Neurons fire a rebound burst of action potentials.

Finally, to finish with the dynamics the model can represent, there is another neuron behaviour called resonator that is convenient to mention. Apart from this one, others with minor importance in the research can be represented either: brainstem, hippocampus, basal ganglia and olfactory bulb.

- **Resonator (RZ) :**

- Neurons have damped or sustained subthreshold oscillations.
- They resonate to rhythmic inputs having appropriate frequency.

2.8 The Izhikevich model

In the Izhikevich model proposed in [3], the author reduced the model to a two-dimensional system of ordinary differential equations plus an auxiliary after-spike resetting behaviour. The model can exhibit firing patterns of cortical neurons with the choice of parameters a , b , c and d , taking only 13 floating point operations to simulate 1 *ms* of the model (at least in MATLAB).

The author recalls that the $+30mV$ is not a threshold, but the peak of the spike. The real value of the threshold is between -70 and -50 ; being a dynamic value as in biological neurons.

- **The differential equations:**

$$\begin{aligned} - \frac{dv}{dt} &= 0.04v^2 + 5v + 140 - u + I \\ - \frac{du}{dt} &= a(bv - u) \end{aligned}$$

- **The after-spike resetting function:**

$$- \text{if } v \geq 30mV, \text{ then } \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases}$$

Where,

- **Dimensionless variables:**

- v , the membrane potential of the neuron.
- u , the membrane recovery variable, which accounts for the activation of K^+ ionic currents and inactivation of Na^+ ionic currents. Thus, it provides a negative feedback to v .

- **Dimensionless parameters:**

- a , the time scale of the recovery variable u .
- b , the sensitivity of the recovery variable u to the subthreshold fluctuations of the membrane potential v .
- c , the after-spike reset value of the membrane potential v caused by the fast high-threshold K^+ conductances.
- d , the after-spike reset of the recovery variable u caused by slow high-threshold Na^+ and K^+ .
- t , the time.

- **Scale:**

- v has mV scale.
- t , has ms scale.

3 Implementation

Firstly, we completed Jupyter Notebooks from Brian2 tutorial available at <https://brian2.readthedocs.io/en/stable/resources/tutorials>. They helped us understand how SNNs and Brian2 work. Then we read the 2 papers [3] [4] that describe the Izhikevich model. Finally, we implemented the model and simulated the proposed neuron types, features and simulation. To help with the implementation we used resources available at <https://www.izhikevich.org/publications/spikes.htm> and <https://www.izhikevich.org/publications/whichmod.htm>. For example, the MATLAB code was very helpful.

All the project steps were implemented in a Jupyter Notebook with Python. All the resulting images can be visualized there. We mainly used Brian2 library. We also used ipywidgets for interaction and matplotlib for visualization. The project description, the written report, the notebooks and relevant papers are available at <https://github.com/juletx/spiking-neural-network>.

We organized the implementation of the project according to the following tasks.

3.1 Importing the libraries

We mainly used `Brian2` library. We also used `ipywidgets` for interaction and `matplotlib` for visualization.

3.2 Defining the model

We defined the model as a python function that receives the parameters of the model as input. Firstly, we define the neuron with the equations. Then, we create a monitor to measure the values of the neuron. Next, we run the simulation for the specified duration. Finally plot the result using `matplotlib`.

We tried to make it flexible enough to adjust to all the neuron types and features. As we advanced through neuron types and features we had to make some adjustments to increase flexibility. For example, we decided to pass `I` as a function instead of a fixed value. Some simulations required changing `I` and this avoids having to copy the model at those. However, there were some cases that it wasn't possible because bigger changes had to be done. For instance, `Class 1 Excitable` required changing the differential equation for `v`. In those cases we copied the model and changed the necessary parts.

3.3 Interacting with the model

We used `ipywidgets` to create sliders that allow interacting with the model by changing the parameters. By playing a bit with the sliders it is clear that the types of spikes are very diverse. In the next sections we define and simulate those types.

3.4 Neuron Types

We defined the neuron types using the parameters from [3]. The duration wasn't specified so we chose duration that was reasonable for all the types. That way, we didn't have to specify the duration and `fi` function for each neuron type. We decided to change the time step from 0.1 to 0.25 so that the simulation was faster. Otherwise, the duration had to be quite long to see enough spikes.

3.5 Neuron Features

We defined the neuron features using the parameters from <https://www.izhikevich.org/publications/whichmod.htm>. In most of the cases the default parameters weren't working as expected. We realised that sometimes increasing the times was enough. In most of the cases after some tries we achieved a good result. However, there were some cases where we couldn't find the correct parameters.

3.6 Defining the simulation

We define the simulation that is proposed in [3] and implemented in <https://www.izhikevich.org/publications/spikes.htm>. See Figure 4 for an illustration of the simulation. This simulation creates a network of 1000 neurons, 800 excitatory and 200 inhibitory. The simulation is run for 1000 ms. It doesn't use any of the previously mentioned types. Instead, some values are calculated randomly to add some variability similar to biological neurons. Moreover, we decided to do something similar to the model to allow some flexibility. We added the number of neurons, time step and duration as parameters.

First, we define the equations and excitatory and inhibitory neuron groups. Next, we define the synapses between neurons, connecting all the neurons with random weights. Then, we create the monitor to record the values and run the simulation. Finally, we plot the a graph with the times of all the spikes and another one with the values of the first neuron.

3.7 Running the simulation

We used `ipywidgets` to create sliders that allow interacting with the model by changing the parameters. The number of neurons can be increased up to 10000. The number of excitatory and

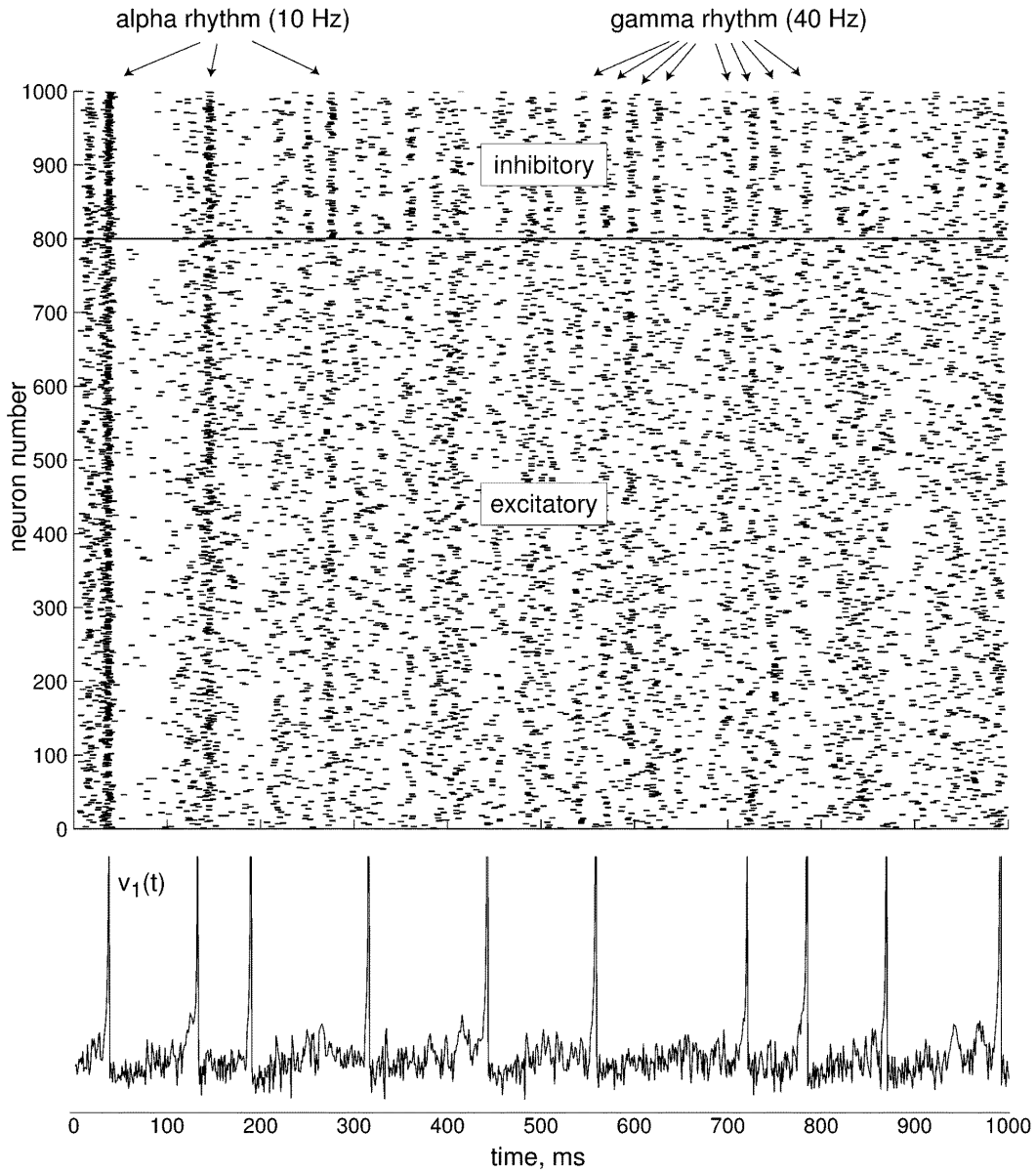


Figure 4: Simulation of a network of 1000 neurons. At the top we can see all the spikes of the neurons though time. At the bottom, we can see the behaviour of one of those neurons.

inhibitory neurons can be changed separately to allow different rates than 4 to 1. The time resolution and duration can also be adjusted.

4 Results

4.1 Neuron Types

We managed to simulate all the types mentioned in [3] correctly except for the Resonator. We tried changing the parameters but we couldn't find the correct combination. We decided to change the time step from 0.1 to 0.25 so that the simulation was faster. The duration of each simulation wasn't specified so we chose 1000 ms so that it was long enough for all the types. Figures of all the neuron types can be seen at the Jupyter Notebook.

4.2 Neuron Features

We achieved a correct result in most of the features that are introduced in [4] by making some adjustments. We have also included the simulations with the proposed parameters so that they can be compared with ours. In general, increasing the duration and the times in the I function was enough. In some cases other adjustments had to be done with different parameters. And there were also some features where we didn't achieve the expected result even if we tried many combinations. Figures of all the neuron features can be seen at the Jupyter Notebook.

4.3 Simulation

The result obtained with the same parameters is a bit different to [3]. However, similar rhythmic behaviour can be achieved by changing the parameters. This can be obtained by increasing the number of neurons and/or increasing tau. For instance, combining 2500 excitatory and 1000 inhibitory neurons with the default tau produces short rhythmic behaviour. The simulation can be run in the Jupyter Notebook to obtain the figures.

5 Conclusions

The Izhikevich model is a powerful model that can simulate many neuron types and features. We haven't managed to simulate all the behaviours, but we achieved most of them. It is also notable that it is computationally efficient and therefore the simulations take few seconds. Even when simulating with a big network of neurons the needed time is reasonable.

We have experimented with Brian2 library and we think that it is a great tool for simulating Spiking Neural Networks. It makes defining the models, simulating and visualization much easier. However, we have clearly seen that replicating the results proposed at [3] [4] isn't that easy. In most of the cases some parameters had to be changed, and there were some cases where we didn't find the correct parameters. This is a common problem with simulations, replicating the results can be challenging even when the code is the same. When we take into account that we were using a different language and library, it isn't surprising that we had some difficulties.

References

- [1] E. Izhikevich and G. Edelman. Large-scale model of mammalian thalamocortical systems. *Proceedings of the National Academy of Sciences*, 105:3593 – 3598, 2008.
- [2] H. Paugam-Moisy, R. Martinez, and S. Bengio. Delay learning and polychronization for reservoir computing. *Neurocomputing*, 71:1143–1158, 2008.
- [3] E. M. Izhikevich. Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6):1569–1572, 2003.
- [4] E. M. Izhikevich. Which model to use for cortical spiking neurons? *IEEE Transactions on Neural Networks*, 15(5):1063–1070, 2004.