Spiking neural models

Single neuron model and introduction to a simple neural network to conduct a working memory experiment

ABSTRACT

Computational modeling in neuroscience is used to test specific hypotheses about theoretical predictions, to control and manipulate parameters in simulations that are not possible to control in the real world, and to help interpret or explore empirical data. [1] The basic functional unit in the nervous system is the neuron and its activity can be described by multiple models.

The **Hodgkin-Huxley** model is a point neuron model or Integrate-and-fire (IAF) neural model. Point neuron models are only concerned with how the neuron handles input voltage to produce, or not produce, an action potential. They are not concerned with more complex features of neurons that can affect the buildup and dissemination of the action potential. [2]. In the present project, this model is going to be introduced, only to later work with a much simplified and computationally efficient derived model, the **Izhikevich´s model**. This second type will be used to study different spiking patterns of single neurons and to build a neural network. To go a step further, this network will be defined to model a cortical set of connected neurons to simulate a short-term memory experiment with a monkey.

Brief overview of the biological neuron

The Hodgkin-Huxley model is an example of an explicit representation of the neuron. This model is concerned with how the movements of ions produce the changes in the voltage of the neuron. Therefore, to understand what this model mimics, a basic knowledge of the ionic changes is important.

When a neuron is at rest, the intracellular fluid is negatively charged, and the extracellular fluid is positively charged (Figure 1) Both compartments are separated by the cell membrane whose permeability will depend on the specific ion type. There are two forces that control the ionic flow across the membrane:

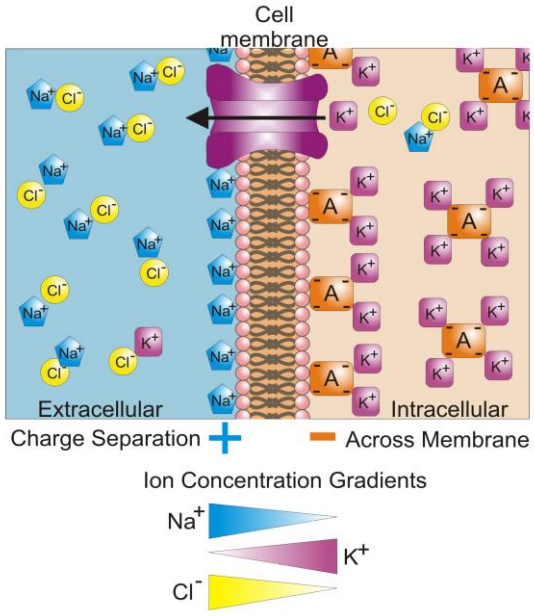

Figure 1: The ionic basis of the resting potential. The extracellular fluid (positively charged) has a high concentration of K^+ and organic ions (A^-) with a smaller concentration of Cl^- and Na^+. The extracellular medium (negatively charged) has a high concentration of Cl^- and Na^+ but a low concentration of K^+ [2].

- **Diffusion**: moves ions following a concentration gradient. For example, because there is a high concentration of K^+ in the intracellular fluid, diffusion exerts pressure to move some of the ions to the extracellular fluid. The opposite happens with Na^+ ions, which move from the extracellular fluid to the intracellular fluid.
- **Electrostatic force**: causes ions of the same charge to be repulsed by each other while making ions of opposite charges attracted to each other. Because the extracellular fluid is positively charged, it repulses positively charged ions, such as K^+, while attracting negatively charged ions such as Cl^-. Similarly, intracellular fluid is negatively charged therefore repulsing Cl^- while attracting K^+. [2]

To achieve stability, the cell membrane contains sodium-potassium pumps which pushes three Na+ ions out of the cell in exchange for pumping two potassium ions into the cell. Overall, these pumps and the cell membrane permeability keep the balance of Na^+ and K^+

stable in the intra- and extracellular fluid despite extra pressure on the Na^+ to enter the intracellular fluid.

Cells also have different ion channels which, when open, allow ions of a particular type to flow though the cell membrane. The opening and subsequent closing of these channels affect, and are affected by, the voltage of the cell membrane and are the cause of action potentials.

Hodgkin-Huxley model

The Hodgkin-Huxley model is one of the simplest biological models, uses four differential equations to compute the membrane potential. These four differential equations model the ionic flow of the neuron. [2]

This model can be understood with the help of Figure 3. The semipermeable cell membrane separates the interior of the cell from the extracellular medium and acts as a capacitor. If an input current I(t) is injected into the cell, it may add further charge on the capacitor, or leak through the channels in the cell membrane. Because of active ion transport through the cell membrane, the ion concentration inside the cell is different from that in the extracellular liquid. The Nernst potential (equilibrium potential of an ionic species) generated by the difference in ion concentration is represented by a battery. [3]

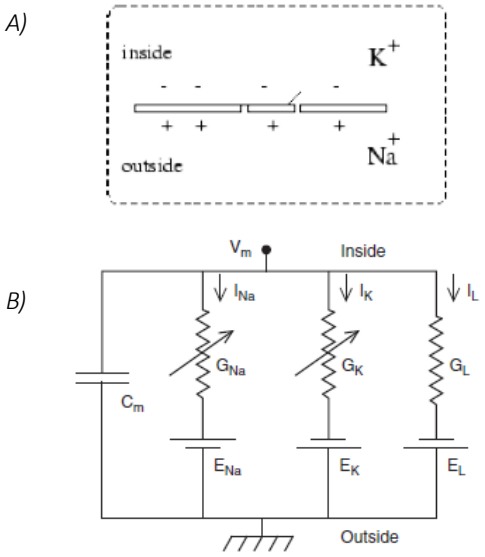

Figure 3: A) Schematic diagram of the Hodgkin-Huxley model. [3]
B) An electrical circuit diagram of a single axonal compartment of a neuron. [1]

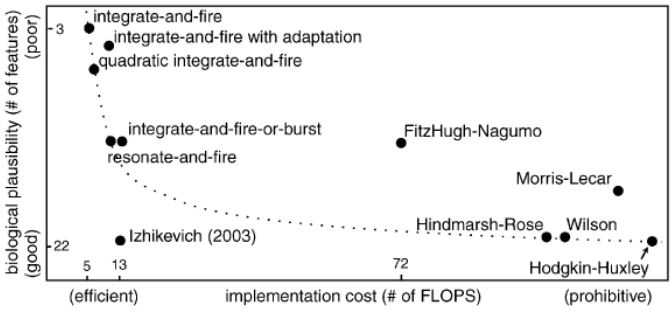

Figure 4: biological plausibility vs implementation costs of different neural models. [4]

Although the Hodgkin-Huxley model is only described by four differential equations, when it is planned to use in more complex neural compositions, like a network of cortical neurons, it has high computational costs. For this reason and because of the high parametrization, the Izhikevich's model has been chosen to carry out the simulations that concern this project. This is a modesl as biologically plausible as the Hodgkin-Huxley. [4]

Izhikevich's model

Izhikevich's model is based on the bifurcation and normal form reduction of the Hodgkin-Huxley model. In other words, it is a simpler version of the Hodgkin-Huxley model which only consists of two differential equations, four parameters and one reset condition. The mathematical derivation of this model is beyond the scope of this report, and we ill only focus on its meaning and implementation.

Although the Izhikevich's model is biologically realistic, its parameters are abstract and do not individually map onto specific biophysical processes but interact in a dynamical model to reproduce behaviors of different classes of neurons (bursting, fast spiking, adapting, etc.). [1]. This ability of showing the different firing dynamics, while still being a simple two-dimensional model is the most important characteristic of the Izhikevich's model.

The equations that define this model are:

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I, \text{(1)}$$

$$\frac{du}{dt} = a(bv - u), \text{(2)}$$

and the reset condition is:

$$\text{If: } v \geq 30, \text{ then } \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases} \quad \text{(3)}$$

The parameters in these equations represent:

- **v:** membrane potential of the neuron. It has mV units and time, t, ms units.
- **u:** generic recovery variable which accounts for the activation of K^+ ionic currents and inactivation of Na^+ ionic currents, and it provides negative feedback to v. After the spike reaches its apex (+30 mV), the membrane voltage and the recovery variable are reset according to the Eq. 3.
- **I:** external input to the neuron. This input could be thought of as external input to the neuron from outside the network or even synaptic input from a neuron within the network.
- **a:** controls the rate of recovery of u. Smaller values result in slower recovery.
- **b:** controls the sensitivity of recovery to subthreshold fluctuations of the membrane potential. Greater values couple v and u more strongly, resulting in possible subthreshold oscillations and low-threshold spiking dynamics.
- **c:** control the after-spike reset values for v caused by the fast high-threshold K^+ conductances.
- **d:** control the after-spike reset values for u caused by slow high-threshold Na^+ and K^+ conductances.

The resting potential in the model is between -70 and -60 mV depending on the value of b. As most real neurons, the model does not have a fixed threshold; Depending on values of the membrane potential prior to the spike, the threshold potential can be as low as -55 mV or as high as -40 mV. [5]

Different types of dynamics - mammalian cortex

The parameters selected for use in the equations can have a huge effect on the final results. For example, in Izhikevich's model, the values of the parameters a, b, c, and d produce drastically different spike patterns.

One logical question would be: *which are the values of a, b c and d that define the different spiking dynamics of neurons? Can this model describe the activity of a typical neuron in, for example, the mammalian cortex?*

To answer this question, the different types of neurons in the mammalian cortex will be introduced next. Two main types can be identified: pyramidal and interneurons neurons. As we know, a neuron receives input, and it emits an action potential when the sum of those inputs exceeds a threshold. Pyramidal neurons are excitatory, which means that they will emit signals that will bring other neurons closer to its threshold Interneurons are inhibitory, they will drag it further away from its threshold. [1]. All excitatory cortical cells can be divided into for classes:

- **Regular spiking neurons (RS):** the most typical neurons in the cortex. When presented with a prolonged stimulus, the neurons fire a few spikes with short inter-spike period and then the period increases. This is called the spike frequency adaptation. Increasing the strength of the injected dc-current increases the inter-spike frequency, though it never becomes too fast because of large spike-afterhyperpolarizations. According to this description, it is expected a deep voltage reset (c) and a large after-spike jump of u (d).
- **Intrinsically bursting neurons (IB):** they fire a stereotypical burst of spikes followed by repetitive single spikes. Therefore, they will have a high voltage reset (high c) and a large after-spike jump of u (d).
- **Chattering neurons (CH):** they can fire stereotypical bursts of closely spaced spikes. The inter-burst frequency can be as high as 40 Hz. In the model, this is expected to correspond a very high voltage reset (c) and moderate after-spike jump of u (d).

All inhibitory cortical cells can be divided into the following two classes

- **Fast spiking (FS)** neurons can fire periodic trains of action potentials with extremely high frequency practically without any adaptation (slowing down). Translated to our model, this corresponds to a fast recovery (a).
- **Low-threshold spiking (LTS)** neurons can also fire high-frequency trains of action potentials, but with a noticeable spike frequency adaptation. These neurons have low firing thresholds (small b). [5]

To determine the values of the parameters that would result in firing patterns corresponding to the ones of the neurons described above, we have simulated the activity of a Izhikevich ´s neuron with a step of DC current I=10mA as input. The code can be found in the MATLAB file *Izhikevich_model.mat*. After twitching the parameter´s values, the matches to the different

spiking dynamics were gathered in Table 1 and the resulting voltages have been plotted and are shown in Figure 5. These values were in concordance with the ones found in the scientific literature.

	RS	IB	CH	FS	LTS
A	0.02	0.02	0.02	0.1	0.1
B	0.2	0.2	0.2	0.2	0.25
C	-65	-55	-50	-65	-65
D	8	4	2	2	2

Table 1: model´s parameters characteristic of the different spiking neural dynamics. Values in light gray are the typical parameters for excitatory or inhibitory cortical neurons regardless their activity.

One must be careful when twitching the parameters because not all possible results are useful. For example, what happens if the parameter d is set as negative or the reset, c, is set to? As seen in the bottom right plot of Figure 5, this result is not useful and would make the simulation meaningless.

When we review Eq. 1 which defines our model, the part "$0.04v^2+5v+140$" is chosen the same for every neuron dynamic. However, it is true that for certain types like RS neurons, other expressions might be more suitable (e.g., function "0.04v2+4.1v+ 108" with b = -0.1). In the next section the single model of the Izhikevich ´s neuron will be introduced in a network, where Eq. 1 will be used as first described.

Network of cortical neurons

Neurons are never found alone in the cortex but connected to others. To obtain a more realistic model of how real cortical neurons work we will create a network of Ne=800 pyramidal neurons and Ni=200 Interneurons neurons, each one connected to the rest of neurons in the network. This is ratio 4:1 choice was motivated by the typical anatomy of the mammalian cortex. [5]

The network represents a cortical module of an area receiving information about the identity of objects (inferotemporal cortex or ventral prefrontal cortex).

Neurons

Both pyramidal cells and interneurons are described by Izhikevich ´s neurons.

To express the heterogeneity of neuron types we will consider all excitatory neurons to be RS and all inhibitory neurons, FS and then, a variability in the parameters will be introduced as follows to reflect different spiking dynamics:

Pyramidal cells:

$$(a_i; b_i) = (0.02; 0.2)$$
$$(c_i; d_i) = (-65; 8) + (15; -6)\cdot r^2_i$$

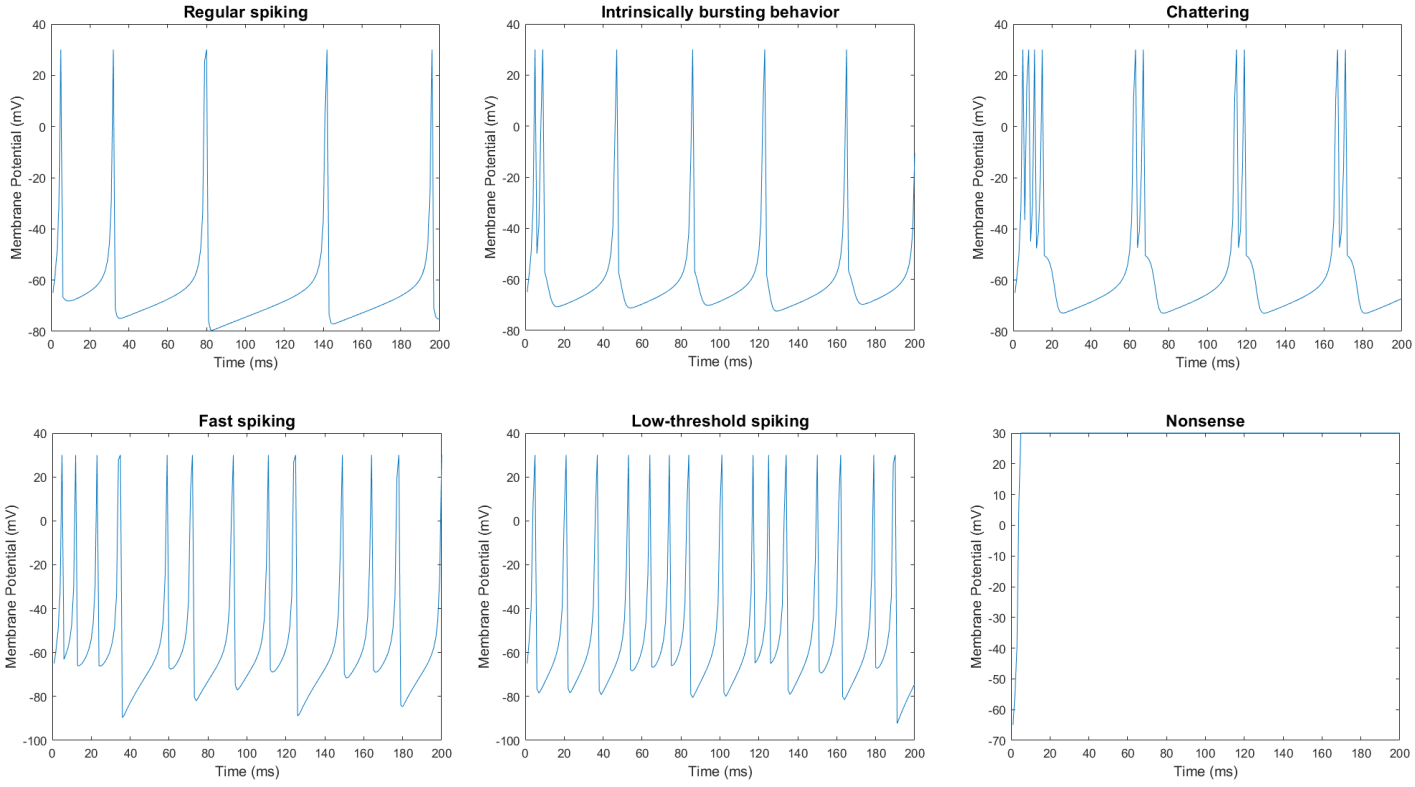

Figure 5: Membrane voltage for different parameter values. Top left) RS. Top middle) IB. Top right) CH. Bottom left) FS. Bottom middle) LTS Bottom right) Nonsense selection of parameter values. The plots have been built using the file Izhikevich_model.mat with the parameters shown in Table 1.

Interneurons cells:

$$(a_i; b_i) = (0.02; 0.25) + (0.08; -0.05)\cdot r_i$$
$$(c_i; d_i) = (-65; 2)$$

where r_i is a random variable uniformly distributed on the interval [0,1], and i is the neuron index. Thus, $r_i = 0$ corresponds to regular spiking (RS) cell, and $r_i = 1$ corresponds to the chattering (CH) cell. We use r^2_i to bias the distribution toward RS cells, which are the most abundant type. [5]

In addition, the network is assumed to encode the identities of p object stimuli. For this reason we will divide the pyramidal cells into p specific populations of f·Ne neurons each (f=0.1) which respond each of them to a to specific stimuli; and one non-specific subpopulation of (1 – f·p)·Ne neurons that do not respond to any of the stimulus that will get stimulated by any type of stimulus.

Synapses:

Besides the synaptic input from the rest of the cells ($I_{Synaptic}$), each neuron receives a noisy thalamic input ($I_{Thalamic}$). They can also be externally excited by a stimulus (I_{ext}).

$$I = I_{Thalamic} + I_{Synaptic} + I_{ext} \qquad (4)$$

Do not forget this model only considers ionic synapsis, not chemical synapses (mediated by neurotransmisors).

External Stimuli

The network is assumed to encode the identities of p object stimuli. Each of them activates a distinct and small subpopulation of f·Ne excitatory cells, with f·p <1. Thus, external stimuli define p functional assemblies of f·Ne neurons, each labelled by its preferred stimulus, and one population of (1 – f·p)·N neurons that do not respond to any of the stimuli.

This subpopulation classification based on the selectivity of the neurons was based on experimental data. This information was extracted from neurophysiological studies where monkey was presented with different visual stimuli. In these experiments, cells that are responsive to at least one of the shown pictures can be classified according to their best stimulus. Cells that do not show any significant activation for any of the shown pictures can be classified in the nonselective group.[6]

Structure of Connections Between Neurons

The model belongs to the class of pulse-coupled neural networks (PCNN), where the synaptic connection weights between the neurons are given by the matrix S = (s_{ij}), so that firing of the j^{th} neuron instantaneously changes variable vi by s_{ij}.[5]

The coupling strength between a pair of neurons is prescribed according to a Hebbian rule: the synapse is strong (or weak) if in the past the two cells tended to be active in a correlated (or anticorrelated) manner.

- Connections between neurons from the same specific group have a coupling strength s_{ii}=win, where win is a number between 0 and 0.5.
- Connections between excitatory neurons of different selective groups and between ones from selective groups with non-selective ones

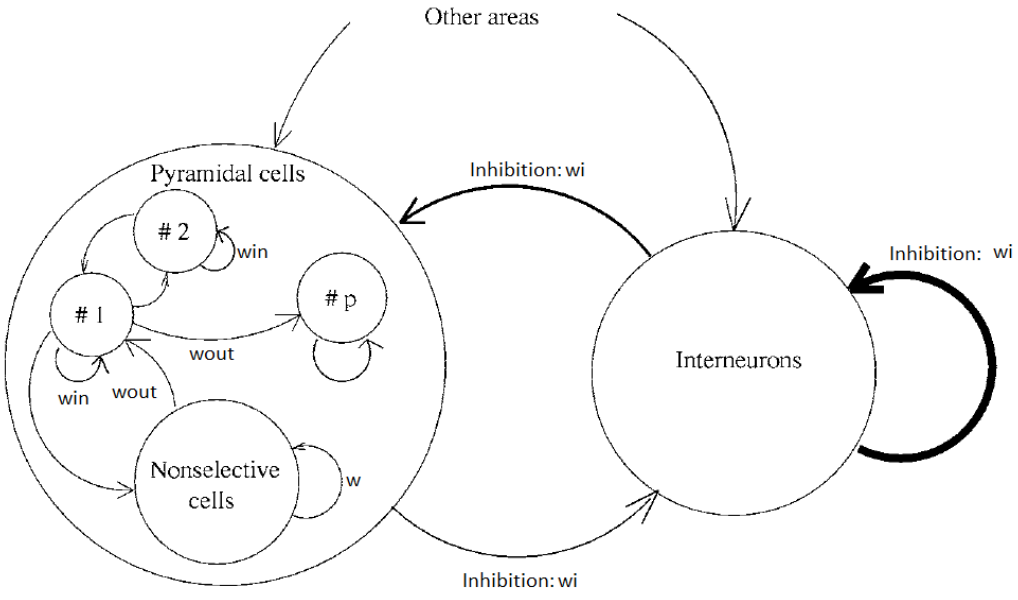

Figure 6: The cortical network model.

have strengths s_{ij}=wout. We will choose wout = 1 – f·(win − 1)/(1 – f).

- Other connections between pyramidal cells have s_{ij}=w, with w equal to a number between 0 and 0.25.
- Connections involving inhibitory cells, have s_{ij}=wi=-1.

The connection matrix S will have very little members equal to zero and therefore, almost every neuron is connected to the rest.

Synaptic efficacies remain fixed through the simulation. [6]

The cortical network is illustrated in Figure 6.

Simulations: Working memory experiment

Now that we have built a model of a cortical neural network, we decided to conduct a simulation of a working memory experiment. In this trial a trained animal (for example, a monkey) would be presented with a visual stimulus. After a few seconds have passed, it is presented with a set of stimuli from which it needs to identify the previously shown picture. To carry out this task, the animal would need to store the first stimulus in its short-term memory.

We will divide the simulation into different stages. During the first 1.5 seconds the variables will be initialized, and spontaneous thalamic activity will happen. At t=2s the first stimulus is presented during 500ms. After 4 seconds, at t=6 all populations are excited, and the monkey needs to select the known image 500 seconds after. In total, the simulation lasts 6.5 seconds.

If we consider our network a realistic model, *could we determine if our monkey is able to remember the picture it saw first?*

Results

Figure 7 shows the basic behavior of the network during a particular simulated trial. In this plot, a random selection of neurons has been plotted. Each dot represents a firing of that neuron at a given time. As expected, after presenting the first stimulus a higher concentration of spikes occurs. This doesn't happen only for the neurons sensible to the specific stimulus, but to all. This is proof that the network is highly interconnected.

In figure 8, the firing rates for each type of neuron has been plotted. With the first stimulus, the firing rate of the selective subpopulation sensible to it rapidly increases.

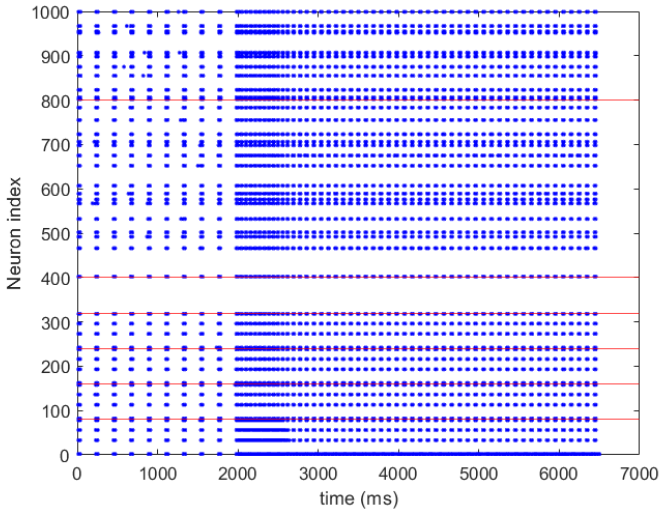

Figure 7: Firings of neurons across the experiment time. Red horizontal lines represent the group separation of neuron types. In this trial, only the first 80 neurons were stimulated.

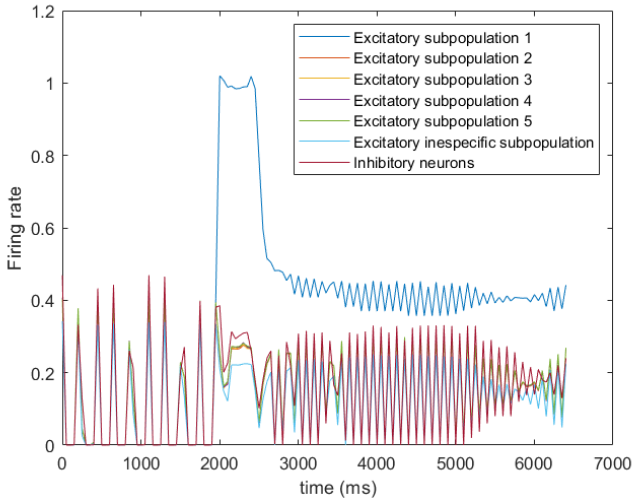

Figure 8: Firings rates of the different types of neurons across the experiment time. A stimulus that excites the selective subpopulation of neurons 1 (1-80) was presented at time t=2s.

When we observe the following seconds, we detect a persistent high activity in the neurons selective to the previous stimulus. This is because they excite cells inside their own subpopulation stronger than the rest of neurons from other groups.

After 4 seconds, the monkey is presented with five photos, therefore, the second stimulus will excite all selective subpopulations. However, due this persistent activity, the excitation of the first selective group will be much higher and allowing the monkey to recognize he familiar visual stimulus.

Conclusions and discussions

In this report we have presented a much simpler and computationally cheaper model than the Hodgkin-Huxley model but with the same biological plausibility. This model is the Izhikevich's model of a neuron and is mathematically derived from the previous one.

Two questions have been stated:

Can this model simulate the different spiking patterns of the typical cortical neurons? Using the MATLAB code Izhikevich_model.mat that can be found in the ANNEX we have found the values for the model´s parameters that describe the spiking of pyramidal and interneurons.

Next, this single neuron model has been integrated into a network that represents a cortical neural network of a mammalian animal. This network is sensible to five different types of stimulus. This model can be found in the MATLAB file Izhikevich_cortex_network.mat and in the ANNEX.

Just as we said at the beginning of this report, computational modeling in neuroscience is used to test specific hypotheses about theoretical predictions. We wanted to test the following hypothesis:

When presented with a set of visual stimuli, **is a monkey able to identify if and which of the images has been shown to it previously?** The answer seems to be positive. The definition of the neural connections following the Hebbian rule enables the network to store the memory of an object by persistent activity of a selective subpopulation of neurons.

However, is this model realistic? It is true that the definition of our connections might be limited. There are two lines of thought that would explain a persistent activity such as this one in a real cortical network:

- Cells that are selective to a particular object would have no spatial relationship. In this scenario, the connectivity structure would be due to Hebbian learning. Two cells firing together during the stimulus presentation would increase the strength of their connections, while long-term depression mechanisms would lead to weaker connections between cells selective to different stimuli.
- In the second scenario, the cells that are selective to the same stimulus would be close together, as perhaps in the same column. The connectivity structure would reflect the fact that the average distance between two cells selective for different objects is larger than between two cells selective for the same object, and thus the connection probability is smaller.

Of course, the situation in the real cortex might be an intermediate one, with cells selective to a particular object tending to cluster in space, and connectivity structure is sharpened by learning processes. More experimental data are needed to distinguish between these scenarios. [6]

Another limitation of the model is that it only considers interactions between neurons through ionic channels.

References:

1. Mike X. *MATLAB for Brain and Cognitive Scientists*. The MIT Press. 2017. ISBN 9780262035828
2. Melissa G. et al. *Spike neural models Part I: The Hodgkin-Huxley model*. The Quantitative Methods for Psychology. 2017. DOI:10.20982/tqmp.13.2.p105
3. Wulfram G. et al. *Spiking Neuron Models Single Neurons, Populations, Plasticity*. Cambridge University Press. 2002. ISBN 0 521 89079 9 : 40 USD
4. Eugene I. *Which Model to Use for Cortical Spiking Neurons?* 2004. EEE Trans Neural Net. DOI: 10.1109/TNN.2004.832719
5. Eugene I. *Simple Model of Spiking Neurons.* 2003. EEE Trans Neural Net. DOI: 1045-9227/03$17.00
6. Nicolas B. et al. Effects of Neuromodulation in a Cortical Network Model of Object Working Memory Dominated by Recurrent Inhibition. Journal of Computational Neuroscience. 2001. DOI: https://doi.org/10.1023/A:1011204814320

ANNEX:

Code for the file: Izhikevich_model.mat

```
%%SPIKING NEURONS FOLLOWING THE Izhikevich MODEL
close all
clear all
clf
%These are some default parameter values
I=10;
spiketype=5;
spikeName={'Regular spiking','Intrinsically bursting behavior','Chattering','Fast spiking','Low-
threshold spiking'};

RS=[0.02 0.2 -65 8]; %regular spiking
IB = [0.02, 0.2, -55, 4]; %intrinsically bursting behavior
CH = [0.02, 0.2, -50, 2]; %CH
FS = [0.1, 0.2, -65, 2]; %fast spiking
LTS = [0.1, 0.25, -65, 2];%Low-threshold spiking
abcd=[RS;IB;CH;FS;LTS]; %Rows: type of dynamics; Columns: [a,b,c,d]

a=abcd(spiketype,1);
b=abcd(spiketype,2);
c=abcd(spiketype,3);
d=abcd(spiketype,4);

%The initial values for v and u
v=-65;
u=b*v;

%Initialize the vector that will contain the membrane potential time series.
v_tot=zeros(1000, 1);

for t=1:1000
    %set v_tot at this time point to the current value of v
    v_tot(t)=v;
    %Reset v and u if v has crossed threshold. See Eq. 3 above.
    if (v>= 30)
    v=c;
    u=u+d;
    end;
    %Use Euler's method to integrate Eqs. 1 and 2 from above. Here v is calculated in
    %2 steps in order to keep the time step small (0.5 ms step in the line below).
    v=v+0.5*(0.04*v^2+5*v+140-u+I);
    v=v+0.5*(0.04*v^2+5*v+140-u+I);
    u=u+a*(b*v-u);
end;
%This line uses the function find to locate the indices of v_tot that hold elements with
%values greater than or equal to 30 and then sets these elements to 30.
%This normalizes to heights of the action potential peaks to 30.
v_tot(find(v_tot >= 30))=30;
%Plot the neuron's membrane potential.
time=1:1000;
figure(1)
plot(time(1:200), v_tot(1:200));
xlabel('Time (ms)', 'fontsize', 12);
ylabel('Membrane Potential (mV)', 'fontsize', 12);
title(spikeName(spiketype), 'fontsize', 14);
```

Code for the file: Izhikevich_cortex_network.mat

```
%%NETWORK MODEL OF SPIKING NEURONS FOLLOWING THE Izhikevich MODEL
close all
clear
clf

% The number of excitatory neurons in the network. The mammalian cortex has about 4 times
%as many excitatory neurons as inhibitory ones.
Ne = 800;
%The number of inhibitory neurons in the network.
Ni = 200;

%Subpopulations of excitatory neurons:
p = 5;
f = 0.1;
n_esp = f*Ne; %number of neurons in each specific subpopulation
n_inesp = (1-f*p)*Ne; %number of neurons in the unspecific subpopulation

%Random numbers
re=rand(Ne, 1);
ri=rand(Ni, 1);

%This will set the value of 'a' for all excitatory neurons to 0.02 and the value of 'a'
%for inhibitory neurons to a random number between 0.02 and 0.1
a = [0.02*ones(Ne, 1); 0.02+0.08*ri];
%This will allow b to range from 0.2-0.25
b = [0.2*ones(Ne, 1); 0.25-0.05*ri];
%This will allow the spike reset membrane potential to range between -65 and -50
c = [-65+15*re.^2; -65*ones(Ni,1)];
%This will allow the recovery reset value to range between 2 and 8
d = [8-6*re.^2; 2*ones(Ni, 1)];

%S: weight matrix, holds info about the strength of connections: (it is a fully connected net)
%HEBBIAN RULE
win = 0.5;%2.1; %between excitatory of the same specific group.
wout = 1-(f*(win-1)/(1-f)); %between excitatory from different groups.
wns = 0.25;% between non selective;
S = ones(Ne+Ni,Ne+Ni); %square matrix of Ne+Ni *Ne+Ni

spi=0;
spj=0;
for i=1:(Ne)
    for j=1:(Ne)
        for k=1:5
            if i<=k*80
                spi=k;
                break
            else
                spi=6;
            end

        end
        for k=1:5
            if j<=k*80
                spj=k;
                break
            else
                spj=6;
            end
        end
        if spi==spj && spi<6
            S(i,j)=win*rand(1);
        elseif spi==spj && spi==6
            S(i,j)=wns*rand(1);
        else
            S(i,j)=wout*rand(1);
        end
    end
end
for i=1:Ne
    for j=Ne+1:(Ne+Ni)
        %S(i,j)=wout*rand(1);
        S(i,j)=-rand(1);
```

```
    end
end
for j=1:Ne+Ni
    for i=Ne+1:(Ne+Ni)
        S(i,j)=-rand(1);
    end
end

% Very few elements of S will be exactly 0, so in this model almost every neuron has
% synaptic contacts with all other neurons in the network.

%The initial values for v and u
v = -65*ones(Ne+Ni,1);
u = b.*v;

%Firings will be a two-column matrix. The first column will indicate the time that a
%neuron's membrane potential crossed 30, and the second column will be a number
%between 1 and Ne+Ni that identifies which neuron fired at that time.
firings=[];

%stimulus:
stimulus1=1; %0-5; 0 means no stimulus.
N_range=[1,80; 81,160; 161,240; 241,320; 321,400; 401,800; 801,1000]; %index ranges for each
populaton type.
time_s1= 2000:2500;%time when the stimulus is first applied.
stimulus2=0; %0-5; 0 means no stimulus.
time_s2= 6000:6500;%time when the stimulus is reapplied.
end_experiment=6500;
for t=1:end_experiment
    %Create some random input external to the network
    I=[5*randn(Ne, 1); 2*randn(Ni,1)]; %thalamic input
    %First Stimulus to a specific supbopulation
    if stimulus1 ==1
        if ismember(t,time_s1)==1
            I_ext=zeros(Ne+Ni,1);
            I_ext(N_range(stimulus1,1):N_range(stimulus1,2))=500;
            I=I+I_ext;%total input
        end
    elseif stimulus1 ==2
        if ismember(t,time_s1)==1
            I_ext=zeros(Ne+Ni,1);
            I_ext(N_range(stimulus1,1):N_range(stimulus1,2))=500;
            I=I+I_ext;%total input
        end
    elseif stimulus1 ==3
        if ismember(t,time_s1)==1
            I_ext=zeros(Ne+Ni,1);
            I_ext(N_range(stimulus1,1):N_range(stimulus1,2))=500;
            I=I+I_ext;%total input
        end
    elseif stimulus1 ==4
        if ismember(t,time_s1)==1
            I_ext=zeros(Ne+Ni,1);
            I_ext(N_range(stimulus1,1):N_range(stimulus1,2))=500;
            I=I+I_ext;%total input
        end
    elseif stimulus1 ==5
        if ismember(t,time_s1)==1
            I_ext=zeros(Ne+Ni,1);
            I_ext(N_range(stimulus1,1):N_range(stimulus1,2))=500;
            I=I+I_ext;%total input
        end
    end
    %Second stimulus to a specific supbopulation
    if stimulus2~=0
        if ismember(t,time_s2)==1
            I_ext=zeros(Ne+Ni,1);
            I_ext(1:400)=500;
            I=I+I_ext;%total input
        end
    end


    %Determine which neurons crossed threshold at the current time step t.
```

```
    fired=find(v>=30); % indices of spikeS
    %!!!fired=find((v+65)>=30); % indices of spikeS
    %Add the times of firing and the neuron number to firings.
    times= t*ones(1, length(fired));
    tn= horzcat(times',fired);
    if isempty(tn)
        firings = firings;
    else
        firings=cat(1,firings,tn);
    end
    %Reset the neurons that fired to the spike reset membrane potential and
    %recovery variable.
    v(fired)=c(fired);
    u(fired)=u(fired)+d(fired);
    %Add to the input, I, for each neuron a value equal to the sum of the synaptic
    %strengths of all other neurons that fired in the last time step connected to that
    %neuron.
    I=I+sum(S(:,fired), 2);

    %Move the simulation forward using Euler's method. Step=0.5 for numerical stability
    v=v+0.5*(0.04*v.^2+5*v+140-u+I);
    v=v+0.5*(0.04*v.^2+5*v+140-u+I);
    u=u+a.*(b.*v-u);

end

%% Plot the raster plot of the network activity.
% We will randomly select 4#N neurons of each type to represent
U=unique(firings(:,2));
%First specific excitatory group: U(1-80)
E1_1 = firings(find(firings(:,2)==U(2)),:);
E1_2 = firings(find(firings(:,2)==U(33)),:);
E1_3 = firings(find(firings(:,2)==U(56)),:);
E1_4 = firings(find(firings(:,2)==U(78)),:);
%Second specific excitatory group: U(81-160)
E2_1 = firings(find(firings(:,2)==U(2+80)),:);
E2_2 = firings(find(firings(:,2)==U(33+80)),:);
E2_3 = firings(find(firings(:,2)==U(56+80)),:);
E2_4 = firings(find(firings(:,2)==U(78+80)),:);
%Third specific excitatory group:
E3_1 = firings(find(firings(:,2)==U(2+80*2)),:);
E3_2 = firings(find(firings(:,2)==U(33+80*2)),:);
E3_3 = firings(find(firings(:,2)==U(56+80*2)),:);
E3_4 = firings(find(firings(:,2)==U(78+80*2)),:);
%Forth specific excitatory group:
E4_1 = firings(find(firings(:,2)==U(2+80*3)),:);
E4_2 = firings(find(firings(:,2)==U(33+80*3)),:);
E4_3 = firings(find(firings(:,2)==U(56+80*3)),:);
E4_4 = firings(find(firings(:,2)==U(78+80*3)),:);
%Fifth specific excitatory group:
E5_1 = firings(find(firings(:,2)==U(2+80*4)),:);
E5_2 = firings(find(firings(:,2)==U(33+80*4)),:);
E5_3 = firings(find(firings(:,2)==U(56+80*4)),:);
E5_4 = firings(find(firings(:,2)==U(78+80*4)),:);
%Inspecific excitatory group:
Eis_1 = firings(find(firings(:,2)==U(402)),:);
Eis_2 = firings(find(firings(:,2)==U(466)),:);
Eis_3 = firings(find(firings(:,2)==U(492)),:);
Eis_4 = firings(find(firings(:,2)==U(502)),:);
Eis_5 = firings(find(firings(:,2)==U(567)),:);
Eis_6 = firings(find(firings(:,2)==U(589)),:);
Eis_7 = firings(find(firings(:,2)==U(607)),:);
Eis_8 = firings(find(firings(:,2)==U(652)),:);
Eis_9 = firings(find(firings(:,2)==U(698)),:);
Eis_10 = firings(find(firings(:,2)==U(706)),:);
Eis_11 = firings(find(firings(:,2)==U(755)),:);
Eis_12 = firings(find(firings(:,2)==U(783)),:);
Eis_13 = firings(find(firings(:,2)==U(532)),:);
Eis_14 = firings(find(firings(:,2)==U(576)),:);
Eis_15 = firings(find(firings(:,2)==U(675)),:);
Eis_16 = firings(find(firings(:,2)==U(723)),:);

%Inhibitory  group:
I_1 = firings(find(firings(:,2)==U(802)),:);
```

```
I_2 = firings(find(firings(:,2)==U(806)),:);
I_3 = firings(find(firings(:,2)==U(875)),:);
I_4 = firings(find(firings(:,2)==U(823)),:);
I_5 = firings(find(firings(:,2)==U(967)),:);
I_6 = firings(find(firings(:,2)==U(999)),:);
I_7 = firings(find(firings(:,2)==U(907)),:);
I_8 = firings(find(firings(:,2)==U(952)),:);
I_9 = firings(find(firings(:,2)==U(898)),:);
I_10 = firings(find(firings(:,2)==U(906)),:);
I_11 = firings(find(firings(:,2)==U(955)),:);
I_12 = firings(find(firings(:,2)==U(855)),:);

figure(1)
plot(E1_1(:,1),E1_1(:,2),'b.',E1_2(:,1),E1_2(:,2),'b.',E1_3(:,1),E1_3(:,2),'b.',E1_4(:,1),E1_4(:
,2),'b.');
hold on
plot(E2_1(:,1),E2_1(:,2),'b.',E2_2(:,1),E2_2(:,2),'b.',E2_3(:,1),E2_3(:,2),'b.',E2_4(:,1),E2_4(:
,2),'b.');
plot(E3_1(:,1),E3_1(:,2),'b.',E3_2(:,1),E3_2(:,2),'b.',E3_3(:,1),E3_3(:,2),'b.',E3_4(:,1),E3_4(:
,2),'b.');
plot(E4_1(:,1),E4_1(:,2),'b.',E4_2(:,1),E4_2(:,2),'b.',E4_3(:,1),E4_3(:,2),'b.',E4_4(:,1),E4_4(:
,2),'b.');
plot(Eis_1(:,1),Eis_1(:,2),'b.',Eis_2(:,1),Eis_2(:,2),'b.',Eis_3(:,1),Eis_3(:,2),'b.',Eis_4(:,1)
,Eis_4(:,2),'b.',Eis_5(:,1),Eis_5(:,2),'b.',Eis_6(:,1),Eis_6(:,2),'b.',Eis_7(:,1),Eis_7(:,2),'b.
',Eis_8(:,1),Eis_8(:,2),'b.',Eis_9(:,1),Eis_9(:,2),'b.',Eis_10(:,1),Eis_10(:,2),'b.',Eis_11(:,1)
,Eis_11(:,2),'b.',Eis_12(:,1),Eis_12(:,2),'b.',Eis_13(:,1),Eis_13(:,2),'b.',Eis_14(:,1),Eis_14(:
,2),'b.',Eis_15(:,1),Eis_15(:,2),'b.',Eis_16(:,1),Eis_16(:,2),'b.');
plot(I_1(:,1),I_1(:,2),'b.',I_2(:,1),I_2(:,2),'b.',I_3(:,1),I_3(:,2),'b.',I_4(:,1),I_4(:,2),'b.'
,I_5(:,1),I_5(:,2),'b.',I_6(:,1),I_6(:,2),'b.',I_7(:,1),I_7(:,2),'b.',I_8(:,1),I_8(:,2),'b.',I_9
(:,1),I_9(:,2),'b.',I_10(:,1),I_10(:,2),'b.',I_11(:,1),I_11(:,2),'b.',I_12(:,1),I_12(:,2),'b.');
yline(80,'r')
yline(160,'r')
yline(240,'r')
yline(320,'r')
yline(400,'r')
yline(800,'r')

xlabel('time (ms)')
ylabel('Neuron index')
hold off

figure(2)
plot(firings(:,1), firings(:,2),'.');
xlabel('time (ms)')
ylabel('Neuron index')

%One can appreciate cortical-like asynchronous dynamics.




%% Calculation of firing rates for each type of neuron:
spiketimes = firings(:,1);
spiketimes = unique(spiketimes);

is_active=zeros(Ne+Ni,end_experiment);

for j=1:end_experiment
    if ismember(j, spiketimes)
        id=find(firings(:,1)==j);
        %for id=find(firings(:,1)==j)
        is_active(firings(id,2),j)=1;
        %end
    end
end

step=50;
%initializing firing rates variables.
n1=zeros(1,length(1:step:(end_experiment-step))); %#active neurons/10msec in the first specific
exciting subpopulation of neurons
n2=zeros(1,length(1:step:(end_experiment-step)));%#active neurons/10msec in the second specific
exciting subpopulation of neurons
```

```
n3=zeros(1,length(1:step:(end_experiment-step))); %#active neurons/10msec in the third specific
exciting subpopulation of neurons
n4=zeros(1,length(1:step:(end_experiment-step))); %#active neurons/10msec in the forth specific
exciting subpopulation of neurons
n5=zeros(1,length(1:step:(end_experiment-step))); %#active neurons/10msec in the fifth specific
exciting subpopulation of neurons
n6=zeros(1,length(1:step:(end_experiment-step)));%#active neurons/10msec in the non specific
exciting subpopulation of neurons
n7=zeros(1,length(1:step:(end_experiment-step))); %#active neurons/10msec in the inhibitor
subpopulation of neurons

i=1;
 for k=1:step:(end_experiment-step)
    n1(i)=(1/step)*sum(sum(is_active(1:80,k:(k+step))));
    n2(i)=(1/step)*sum(sum(is_active(81:160,k:(k+step))));
    n3(i)=(1/step)*sum(sum(is_active(161:240,k:(k+step))));
    n4(i)=(1/step)*sum(sum(is_active(241:320,k:(k+step))));
    n5(i)=(1/step)*sum(sum(is_active(321:400,k:(k+step))));
    n6(i)=(1/step)*sum(sum(is_active(401:800,k:(k+step))));
    n7(i)=(1/step)*sum(sum(is_active(801:1000,k:(k+step))));
    i=i+1;
 end

n = [n1./n_esp; n2./n_esp; n3./n_esp; n4./n_esp; n5./n_esp; n6./n_inesp; n7./Ni];

figure(3)
%Plot the firing rates for the different neuron types in the network.
time=1:step:end_experiment-step;
plot(time,n(1,:), time,n(2,:), time,n(3,:), time,n(4,:), time,n(5,:), time,n(6,:), time,n(7,:))
xlabel('time (ms)')
ylabel('Firing rate')
legend('Excitatory subpopulation 1', 'Excitatory subpopulation 2', 'Excitatory subpopulation
3','Excitatory subpopulation 4', 'Excitatory subpopulation 5','Excitatory inespecific
subpopulation', 'Inhibitory neurons');
```