Mathematical modelling of neural impulses

Desislava Mineva, Spasimir Nonev, Zdravka Nedyalkova

June 13, 2017

Sofia University St.Kliment Ohridski, Bulgaria

Abstract

The impact of the nerve system on the behavior and the cognitive abilities of the human being are of high importance for the unsolved problems in the research of the human brain.
In this article we shall review the function of a nerve impulse and the process of transmitting a signal from one cell to another. We shall specifically analyze the Hodgkin-Huxley model, which although obtained for a specific nerve in a specific species — a squid, gives a very good description of the propagation of neural impulses. The model is derived using the charge conservation principle and the Ohm's law. It consists of 4 PDEs — 1 of second order in terms of a space variable x and 3 of first order in terms of time. The system is solved with different boundary and initial conditions (Dirichlet and Neumann) using finite difference method. We shall give a short introduction to machine learning and neural networks.

1 Introduction

1.1 Neuron structure

A neuron (also known as a nerve cell) is a cell that process and transmits information through electrical and chemical signals. A typical neuron(see Figure 1) consists of:

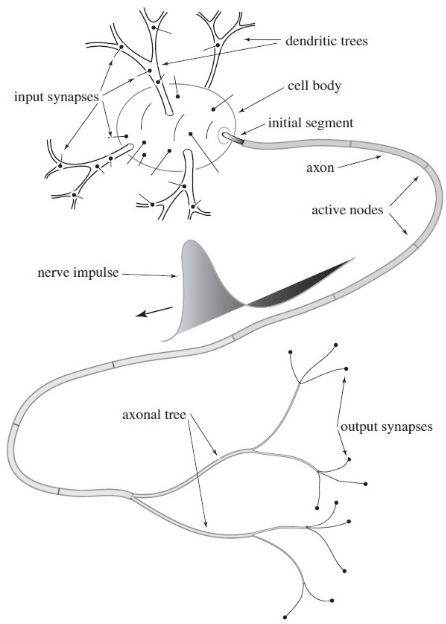


Figure 1: Structure of the neuron

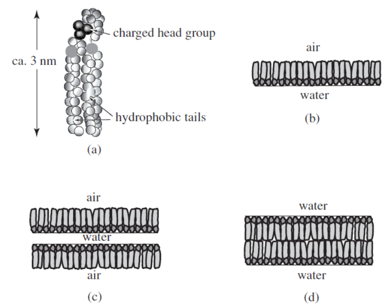
- cell body — It appears as a star shaped structure and there is located neuron's nucleus and synthesized a lot of proteins.
- axon (nerve fibre) — It is responsible for carrying electrical impulse away from the cell body.
- active nodes — In them nerve membrane is exposed and the switching action occurs.
- initial segment — The region of the axon which is connected to the cell body. The electrical signal is generated there and through the axonal tree is transmitted to the next neuron.

- axonal tree — The axon of a neuron eventually branches (or bifurcates) in a tree-like manner, allowing the impulse train on the trunk to be directed toward a variety of locations on muscle cells or other neurons.
- dendritic trees — Structures on the neuron that receive and process electrical messages; single neuron may have more than one set of dendrites, and may receive many thousands of input signals.
- neuron membrane — a very thin membrane, composed of lipids and protein, that surrounds the cytoplasm of a cell and controls the passage of substances into and out of the cell.
- Na+/Ka+ channels — integral membrane proteins that form ion channels, conducting Na+/Ka+ ions through a cell's plasma membrane.

A signal transmitted along a nerve fibre is called nerve impulse. There is an electrical difference between the inside of the axon and its surroundings. When the nerve is activated, there is a sudden change in the voltage across the wall of the axon, caused by the movement of ions in and out of the neuron. This triggers a wave of electrical activity that passes from the cell body along the length of the axon to the synapse.

1.2 Neural Membrane

The neural membrane serves as a barrier to enclose the cytoplasm inside the neuron, and to exclude certain substances that float in the fluid that bathes the neuron. The membrane is made of lipids and proteins — fats and chains of aminoacids(See Figure 2). The basic structure of this membrane is a bilayer of phospholipids, organized in such a way that the polar (charged) regions face outward and the non-polar inward.
The membrane has very important electrical property. It is a capacitor. Capacitor is a charge-storing component consisting of two conductors separated by a dielectric (insulator). The current is carried in the wires by electrons, in biological systems such as the neuron it is carried by ions. The current flow outside the nerve cell and inward through the membrane is primarily carried by



(a) A lipid (fatty) molecule (redrawn from Goodsell). (b) A monomolecular lipid layer on water surface. (c) A bimolecular soap film. (d) A lipid bilayer.

Figure 2: Structure of the membrane

Na^+ ions. Similarly, intracellular and outward currents are primarily carried by K^+ ions because of its relatively high intracellular and low extracellular concentrations.

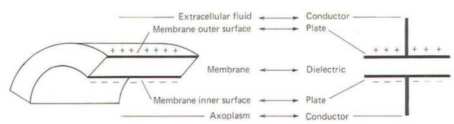


Figure 3: Neuron membrane as a capacitor

A capacitor is characterized by its capacitance — a number that shows how good is it at storing charge. Capacitor with a large capacitance will store a lot of charge.
The capacitance of a capacitor could be obtained from the following formula showing the dependancy between storage and voltage [1]:

$$\text{Capacitance} = \frac{\text{charge stored on a capacitor}}{\text{voltage of that capacitor}}$$

The unit measurement of the capacitance is Farad(F):

$$[\frac{\text{Coloumbs}}{\text{Vold}}] = [\text{Farad(F)}]$$

The inside and the outside of the cell are both solutions of various salts in water. As opposed to the cell membrane, salt water

[1] *voltage – difference in electrical potencial between two pieces of metal; electrical potential is high near positive charges and low near negative charges.

constitutes quite a good conductor because there are free ions that can transport electrical charges. There are two conductors (the inside and the outside of the cell), which are separated by an insulator (the membrane, see Figure 2). This makes it possible to have different amounts of electrical charges inside and outside the cell. The membrane is so thin (only two molecules thick, with a total thickness of about

$$6 * 10^{-9} m$$

that there is no need of much voltage to separate the charges and therefore the membrane capacitance is quite high; per unit area, it is around:

$$10^2 F m^2.$$

The membrane posses sections which together with the diffusion layer of ions from the surrounding water environment form peripherally electrically charged layers. The lipophilic sections of membrane molecules directed toward the interior of the it do not have such groups and form a central dielectric layer. Together electrically charged layers (from both sides of the membrane) and dielectric layer form system which represents a capacitor and is characterized by electrical capability.

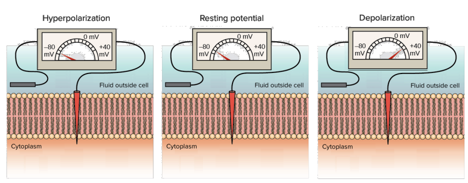


Figure 4: Neuron membrane as a capacitor

1.3 Nerve impulse transmission

The nerve impulses are transmitted as a domino effect. Each neuron gets an impulse and pass it to the next. Through a series of chemical processes the dendrites receive an impulse located in the axon's of other neurons and transmit it to the dendrits of the next axon. A neural impulse passes through the neuron for about 17 mileseconds — faster than lightning!

The process occurs in the following way:

- Neuron cell membrane polarization.

  More of the Na^+ are located outside of the neuron and more of the K^+ are located inside it. When the neuron is not stimulated(it is at rest), the membrane is said to be polarized. The polarization of the neuron cell membrance means that the electrical charge in its internal environment is negative. Its external environment is rich in sodium ions and the interior — of potassium. Besides K^+ in the inner environment of the neuron there are also negatively charged protein molecules as well as those of the nucleic acids, for this reason the electrical charge is negative. In fact K^+ and Na^+ ions move in both directions - in and out of the membrane. But nature has taken care of everything: on the membrane for the so-called K^+/Na^+ channels which are responsible for "pumping" Na^+ outwards and K^+ back inward.

- Rest potential:

  When the neuron is inactive and polarized, it is said to be in the rest potential. It remains until it is stimulated again.

- Active potential:

  When stimulated the closed neural ion channels of the neuron being at rest suddenly open and allow the sodium ions outside the cell to enter it. When this occurs the neuron passes from a polarized to a depolarized state. When more positive ions enter the cell the internal environment also becomes positive. The neuron goes into a state of depolarization and a threshold is reached. Each neuron has a threshold level which is a time of irreversibility. Once the stimulation passes the threshold more closed channels open and allow more Na^+ to enter. This causes complete depolarization of the neuron and creates an active potential. In this state the neuron continues to open more channels. This is the so-called "all or nothing" phenomenon. "All or nothing" means that if the signal is not strong enough and the threshold is not reached, no more ion channels will be opened and the signal will not be transmitted.

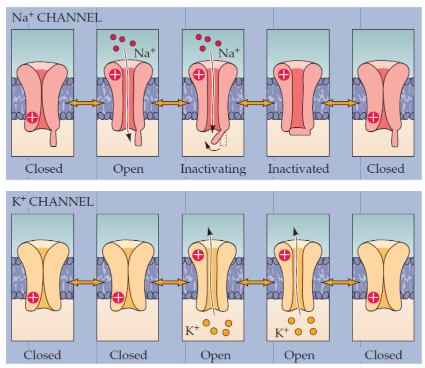


Figure 5: K^+/Na^+ Channels

- Repolarisation:

  Na^+ ions are moved from outside and K^+ remains in the neuron. Once the interior of the neuron becomes saturated with Na^+, closure channels on the inside of the membrane are opened to allow the K^+ to move out of the cell. In this way the electrical balance is restored, although it is the opposite of the initial polarization of the neuron when Na^+ are located outside and K^+ inside.

- Reflection:

  This period occurs when K^+ and Na^+ ions are returned to their original locations: K^+ in the call, Na^+ outside. While the neuron is working to restore its normal state, it does not receive signals. Once the potassium-sodium pump restores the balance, the neuron is again in its normal state of polarization and resting potential until another impulse comes.

- Na^+/Ka^+ Pump

  In the cell membrane of each neuron are ion pumps, which are protein molecules that span the membrane and use metabolic energy to transport some ions inside the cell and others outside. The Na^+/K^+ pump moves two potassium ions into the cell and, at the same time, three sodium ions out of the cell. After some period of time the concentration of potassium inside the cell becomes larger than that outside, and the concentration of sodium becomes larger outside than inside. Running the pump requires energy, which is provided to the pump in the usual energy currency of the cell, the ATP-ADP process. By having these pumps located throughout the cell membrane, sodium is brought out and potassium brought in, in order to bring the cell back to rest. If the neuron does not maintain this gradient, it will not be able to participate in impulse conduction.

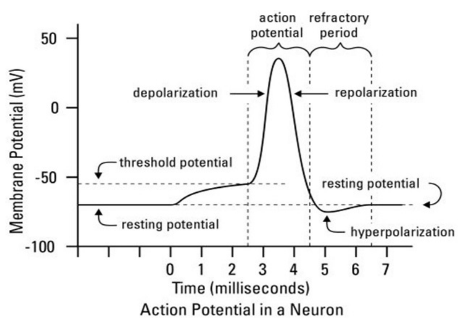


Figure 6: Function of the neuron impulse transmission.

1.4 Membrane and ionic currents

The total transmembrane current comprises three components:

- Capacitance current. A cell's capacitance determines how quickly the membrane potential can respond to a change in current. When there is a voltage difference (such as the resting membrane potential) across an insulator, charge will build up at the interface because current can't flow directly across the insulator. When this built up charge becomes large enough, an induced (capacitive) current is produced, which can change the membrane voltage. In other words, when current is injected into the cell, most of this current is used initially to charge the membrane capacitance, which basically amounts to a certain amount of current being required to change the distribution of ions near the cell membrane. As the capacitance becomes charged and current continues to be injected, the ion concentration gradients on the intracellular side of the membrane change and therefore the membrane potential changes.
- Conduction current. It is responding to the voltage difference across the membrane. Voltage, which is synonymous with difference in electrical potential, is the ability to drive an electric current across a resistance. The same principle applies to voltage in cell biology. In electrically active tissue, the potential difference between any two points can be measured by inserting an electrode at each point, for example one inside and one outside the cell, and connecting both electrodes to the leads of what is in essence a specialized voltmeter. By convention, the zero potential value is assigned to the outside of the cell and the sign of the potential difference between the outside and the inside is determined by the potential of the inside relative to the outside zero.
- Diffusion current. Diffusion current is a current in a semiconductor caused by the diffusion of charge carriers. It is responding to the difference in ionic concentrations across the membrane.

2 Hodgkin-Huxley model

One of the most important models describing the dynamics of the nerve impulse is the one of Hodgkin-Huxley[1] which although obtained for a specific nerve (of a squid) contains the main concepts for the research in this field.
In this chapter we shall derive the model that describes the transmission of a nerve impulse by using two of the most essential laws in the nature — the law of conservation of charge and the Ohm's law.

2.1 Charge conservation principle

The model of Hodgkin-Huxley is a model which describes the propagation of a neural impulse in an axon of a squid. Because the axon is a long thin projection of the cell we shall consider all functions in the model to be functions of one space variable x and the time t. Let us denote the transmembrane voltage at position x and time t with $V(x,t)$, the electric charge — with $i(x,t)$ and the current — with $j(x,t)$.
Most of the laws in physics origin from laws of conservation. In order to derive a model describing the transmission of current we shall use the charge conservation principle i.e. charge cannot be destroyed nor created but can only be transferred from one place to another.
Let us consider two points ξ and $\xi + \Delta\xi$ on the axon where $\Delta\xi$ is very small (see Figure 7). The electric charge between them at time t is exactly equal to the difference between the current that flows in at point ξ and flows out at point $\xi + \Delta\xi$.

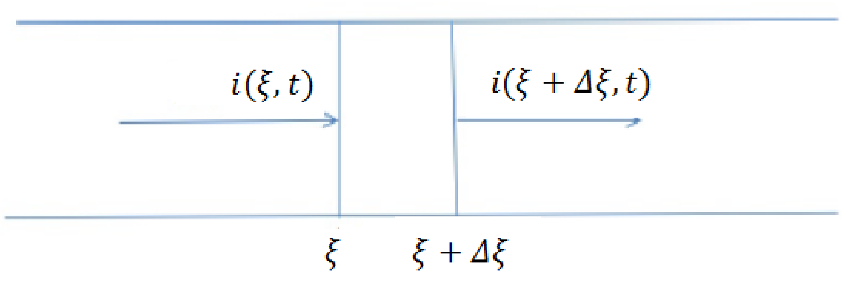


Figure 7: Flow of current between points ξ and $\Delta\xi$

If we assume that $\Delta\xi$ is infinitely small we obtain

$$\int_{\xi}^{\xi+\Delta\xi} j(x,t)dx = i(\xi,t) - i(\xi+\Delta\xi,t).$$

We approximate the integral on the left side using the approximation formula of the left rectangles and we obtain

$$j\Big(\xi + \frac{\Delta\xi}{2}, t\Big)\Delta\xi = i(\xi,t) - i(\xi+\Delta\xi,t).$$

After dividing both sides of the latter by $\Delta\xi$ we get

$$j\Big(\xi + \frac{\Delta\xi}{2}, t\Big) = \frac{i(\xi,t) - i(\xi+\Delta\xi,t)}{\Delta\xi}.$$

Now we let $\Delta\xi \to 0$ and we obtain the differential form of the equation of the charge conservation principle:

$$j = -\frac{\partial i}{\partial x}. \tag{2.1}$$

In the latter we have two unknown functions — $i(x,t)$ and $j(x,t)$. In the next section we shall use Ohm's law in order to define a relation between $i(x,t)$ and $V(x,t)$.

2.2 Ohm's law

One of the most important and basic laws of electrical circuits is the Ohm's law. It states that the current passing through a conductor is proportional to the voltage over the resistance i.e.

$$V(x,t) = i(x,t).R. \tag{2.2}$$

Let us consider the same interval $[\xi, \xi+\Delta\xi]$. The membrane voltage at point ξ is equal to the difference in the electric potential inside the axon and the electric potential outside i.e for the voltage at point ξ and $\xi+\Delta\xi$ we have:

$$V(\xi,t) = V_{in}(\xi,t) - V_{out},$$

$V(\xi + \Delta\xi, t) = V_{in}(\xi + \Delta\xi, t) - V_{out}.$

In the last expression, we assume that the potential outside the axon at every two points is constant. Subtracting the second equation from the first one we obtain:

$$V(\xi, t) - V(\xi + \Delta\xi, t) = V_{in}(\xi, t) - V_{in}(\xi + \Delta\xi, t). \tag{2.3}$$

Taking into consideration the Ohm's law, (2.2) and (2.3) we obtain

$$\int_{\xi}^{\xi+\Delta\xi} R.i(x,t)dx = V(\xi, t) - V(\xi + \Delta\xi, t).$$

Assuming that the resistance R is a constant we have

$$R. \int_{\xi}^{\xi+\Delta\xi} i(x,t)dx = V(\xi, t) - V(\xi + \Delta\xi, t).$$

We again approximate the integral with the approximation formula of the left rectangles and obtain

$$R.i(x + \frac{\Delta\xi}{2}, t) = \frac{V(\xi, t) - V(\xi + \Delta\xi, t)}{\Delta\xi}.$$

Taking the limit $\Delta\xi \to 0$ and we conclude

$$R.i(x,t) = -\frac{\partial V}{\partial x}. \tag{2.4}$$

2.3 The "Cable" equation

Substituting (2.1) in (2.4) obtained in the previous subsections we obtain the following partial differential equation of second order:

$$\frac{1}{R}\frac{\partial^2 V}{\partial x^2} = j(x,t). \tag{2.5}$$

In the following subsection we shall formulate Hodgkin and Huxley's "cable" equation based on (2.5). We shall analyze the processes of transmitting neuron's impulse.

Considering that the total current is a sum of the transmembrane current j_{ion}, the displacement current j_{cap} and the current-voltage relation in a capacitor we obtain

$$\frac{1}{R}\frac{\partial^2 V}{\partial x^2} = j_{ion} + C\frac{\partial V}{\partial t} \tag{2.6}$$

where C is the capacitance per unit of the bilayer.
Using two main techniques — space and voltage clamping by Cole and others [2] Hodgkin and Huxley empirically derived the expression for the transmembrane current

$$j_{ion} = G_{Na}m^3h(V - V_{Na}) + G_K n^4(V - V_K) + G_L(V - V_L) \tag{2.7}$$

where

- G_{Na}, G_K, G_L are the potassium, sodium and leak conductance per unit area
- V_{Na}, V_K, V_L are the potassium, sodium and leak potentials
- $m(x, t)$ is a "sodium turn-on" variable
- $h(x, t)$ is a "sodium turn-off" variable
- $n(x, t)$ is a "potassium turn-on" variable.

The latter three variables describe the dynamics of the sodium and potassium channels, which we shall model in the next section. The variable $m(x, t)$ describes the opening of the Na^+ channels during the transfer of the neural impulse signal, $n(x, t)$ — their closure and the variable $h(x, t)$ — the opening and closing of the channels responsible for the flow of the K^+ ions across the membrane. These variables vary in the interval $[0, 1]$. If $m(x, t) = 1$ that means that the Na^+ channels are fully open. Analogically, if $h(x, t) = 1$ the Na^+ channels are fully open.

We then substitute (2.7) in (2.6) to formulate the "cable" equation.

$$\frac{1}{R}\frac{\partial^2 V}{\partial x^2} = G_{Na}m^3h(V - V_{Na}) + G_K n^4(V - V_K) + G_L(V - V_L) + C\frac{\partial V}{\partial t}.$$

2.4 Description of the potassium and sodium channels

In the "cable" equation there are 4 unknowns $V(x,t)$, $m(x,t)$, $n(x,t)$ and $h(x,t)$. Because of that we need 3 other relations between the unknown functions to formulate a system which can be solved. Let us consider the variables $m(x,t), n(x,t)$ and $h(x,t)$ and fix $V = V_0 = const$. The variables $m(x,t), n(x,t)$ and $h(x,t)$ after time will converge to equilibrium state which we denote with m_0, n_0 and h_0. For different values of V we obtain different values for these equilibrium states m_0, n_0 and h_0 which depend on V i.e. $m_0(V)$, $n_0(V)$, $h_0(V)$.
Therefore, $m(x,t), n(x,t)$ and $h(x,t)$ are the solutions of following partial differential equations:

$$\left|\begin{array}{lcl} \frac{\partial m}{\partial t} & = & -\frac{m-m_0(V)}{\tau_m(V)}, \\ \frac{\partial h}{\partial t} & = & -\frac{h-h_0(V)}{\tau_h(V)}, \\ \frac{\partial n}{\partial t} & = & -\frac{n-n_0(V)}{\tau_n(V)}, \end{array}\right.$$

where $\tau_m(V)$, $\tau_n(V)$ and $\tau_h(V)$ is the the time for reaching the equilibrium value.
Combining the latter system and (2.7) we obtain the Hodgkin-Huxley model

$$\left|\begin{array}{lcl} \frac{1}{R}\frac{\partial^2 V}{\partial x^2} & = & C\frac{\partial V}{\partial t} + G_{Na}m^3h(V - V_{Na}) + G_K n^4(V - V_K) + G_L(V - V_L) \\ \frac{\partial m}{\partial t} & = & -\frac{m-m_0(V)}{\tau_m(V)}, \\ \frac{\partial n}{\partial t} & = & -\frac{n-n_0(V)}{\tau_n(V)}, \\ \frac{\partial h}{\partial t} & = & -\frac{h-h_0(V)}{\tau_h(V)}. \end{array}\right. \tag{2.8}$$

This system consists of 4 partial differential equations. In order to solve them we need 4 initial and 2 boundary conditions. The solution of the system (2.8) is obtained in the next section.

3 Numerical solution of the system (2.8)

In this section we shall solve the system (2.8) with appropriate initial and boundary conditions. It cannot be solved analytically, because of that we shall first approximate it and then solve it numerically.
The system (2.8) consists of three first order partial differential equations with respect to the time variable and one second order partial differential equation with respect to the space variable.

3.1 Euler method

In order to solve system (2.8) we have to be familiar with the numerical methods that can be used for solving a first order ODE. Let $u(t)$ be a function of a single variable. We consider the Cauchy's problem:

$$\left| \begin{array}{rcl} \frac{du}{dt} & = & f(t, u(t)), \qquad t \in (t_0, T], \\ u(t_0) & = & u_0. \end{array} \right.$$

Firstly, let us define a grid:

$$\omega_h = \{t_i = t_0 + ih,\ i = \overline{0, n},\ n = (T - t_0)/h\}.$$

We denote the exact solution at points t_i of the grid by $u(t_i) = u_i$. Let us consider the definition of a first order derivative

$$u'(t) = \lim_{h \to 0} \frac{u(t+h) - u(t)}{h} = \frac{u(t+h) - u(t)}{h} + O(h). \tag{3.1}$$

Since the term O(h) is considerably small (if h is small) we can neglect it and using the notation $u_i \approx y_i$ (where y_i is the approximate solution in t_i) we obtain the following approximated problem

$$\left| \begin{array}{rcl} \frac{y_{i+1} - y_i}{h} & = & f(t_i, y_i), \qquad i = \overline{0, n-1}, \\ y_0 & = & u_0 \end{array} \right.$$

or equivalently

$$\left| \begin{array}{rcl} y_{i+1} & = & y_i + h.f(t_i, y_i), \qquad i = \overline{0, n-1}, \\ y_0 & = & u_0. \end{array} \right.$$

The Cauchy problem can be solved iteratively i.e. we obtain the approximation in the next point of time by using the previous one. If we fixed $V = V_0 = \text{const}$ (considering $m(x,t)$ as a function of t only) we can apply the Euler method to the problem

$$\left|\begin{array}{lcl} \frac{dm}{dt} & = & -\frac{m-m_0(V_0)}{\tau_m(V_0)}, \qquad 0 < t \leq T, \\ m(0) & = & m_0^*. \end{array}\right. \tag{3.2}$$

We choose the initial value m_0^* to be equal to the value of m_0 at resting potential i.e. $m_0^* = m_0(-65)$. Approximating the latter we obtain

$$\left|\begin{array}{lcl} \frac{m_{i+1}-m_i}{h} & = & -\frac{m_i-m_0(V_0)}{\tau_m(V_0)}, \qquad i = \overline{0, n-1}, \\ m(0) & = & m_0(-65). \end{array}\right.$$

We express m_{i+1} in terms of m_i and obtain

$$\left|\begin{array}{lcl} m_{i+1} & = & \left(1 - \frac{h}{\tau_m(V_0)}\right) m_i + \frac{h}{\tau_m(V_0)} m_0(V_0), \qquad i = \overline{0, n-1}, \\ m(0) & = & m_0(-65). \end{array}\right.$$

Using Mathematica we implemented similar algorithms for $m(x,t)$, $n(x,t)$ and $h(x,t)$:

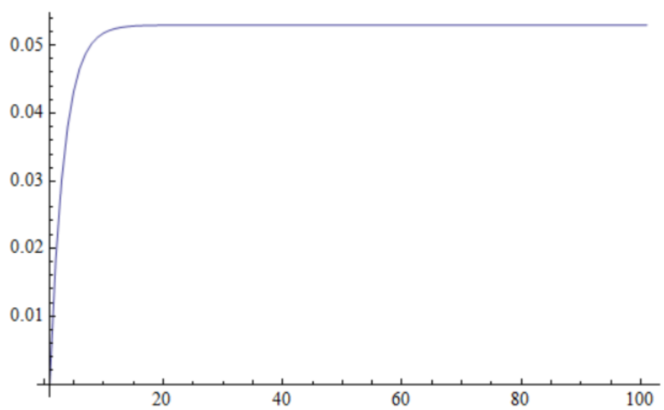


Figure 8: Solution of (3.2) with $V_0 = 0$

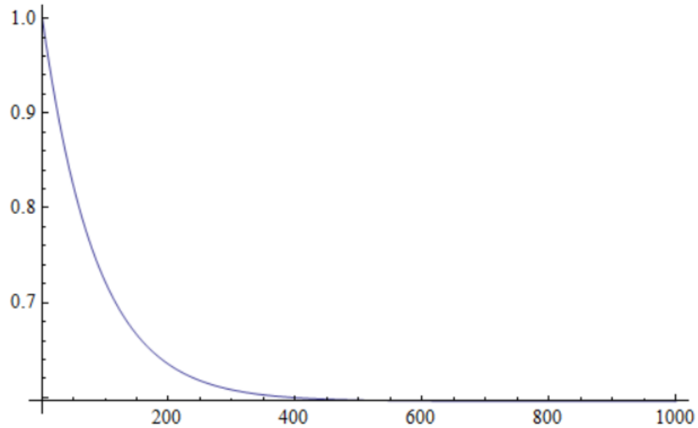


Figure 9: Solution of (3.2) for h with $V_0 = 0$

The results are shown on Figure 8 and Figure 9. On Figure 8 it can be seen that the value of $m(x,t)$ converges to an equilibrium state which is close to 0.05. The results describe well the real world since once open the channels will stay open in time. Analogically, the value of $h(x,t)$ will converge to 0, which would mean closing of the sodium channels.
We shall use the approximation (3.1) for the first order partial differential equations in system (2.8). Because they are not functions of one variable, but on two, we shall have to define another grid to approximate them.

3.2 Finite difference method for second order partial differential equations

Before approximating the system (2.8), we shall consider a simple example of second order partial differential equation.
Let us consider the following differential problem containing the homogeneous diffusion equation

$$\left| \begin{array}{ll} \dfrac{\partial u}{\partial t} - \dfrac{\partial^2 u}{\partial x^2} = 0, & 0 < x < l, \quad 0 < t \leq T, \\ & \\ u(x,0) = u_0(x), & 0 \leq x \leq l, \\ u(0,t) = u_l(t), & 0 \leq t \leq T, \\ y(l,t) = u_R(t), & 0 \leq t \leq T. \end{array} \right.$$

In this problem $u(x,t)$ (the unknown function) is a function of two variables, space variable x and time variable t. Because of that we define the following two-dimensional rectangular uniform grid:

$$\omega_{h,\tau} = \{(x_i, t_j) : x_i = ih, t_j = j\tau, i = \overline{0,n}, j = \overline{0,m}, n = l/h, m = T/\tau\}.$$

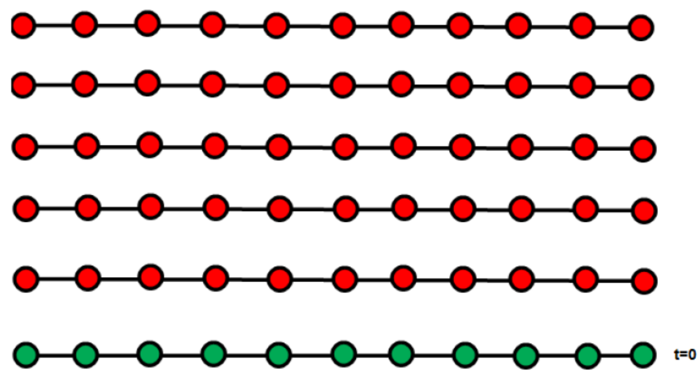


Figure 10: Grid for second order differential equation approximation

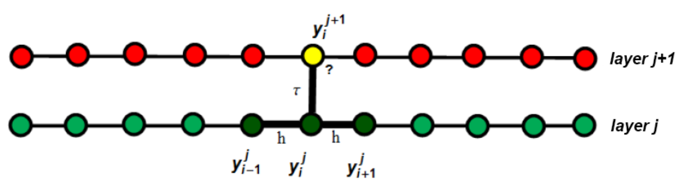


Figure 11: The stencil for the explicit method for the heat equation

The steps that we shall take until the final approximated system are as follows. We shall solve the problem layer by layer. It is clear to see that the initial condition on time gives the values on the first layer or that is the layer $t = 0$ (see Figure 10). Moreover using the boundary conditions we can compute the most left and the most right points on every layer. If the values on the previous layer and those on the borders are computed we can find each point on the next layer of the grid. More specifically, we only need the values y_{i-1}^j, y_i^j, $y_{i+1}^j$and y_i^{j+1} (see Figure 11).

We use the following formula for approximating a second order derivative:

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_j) = \frac{u(x_i + h, t_j) - 2u(x_i, t_j) + u(x_i - h, t_j)}{h^2} + O(h^2)$$

and approximating the diffusion equation, we obtain

$$\left|\begin{array}{ll} \frac{y_i^{j+1} - y_i^j}{\tau} - \frac{y_{i+1}^j - 2y_i^j + y_{i-1}^j}{h^2} = 0, & i = \overline{1, n-1}, \quad j = \overline{0, m-1}, \\ y_i^0 = u_0(x_i), & i = \overline{0, n}, \\ y_0^j = u_l(t_j), & j = \overline{0, m}, \\ y_n^j = u_R(t_j), & j = \overline{0, m}. \end{array}\right.$$

3.3 Finite difference method for the system (2.8)

Using the finite difference method for the system (2.8), we obtain the following approximation:

$$\left|\begin{array}{ll} \frac{1}{R}\frac{y_{i+1}^j - 2y_i^j + y_{i-1}^j}{h^2} = C\frac{y_i^{j+1} - y_i^j}{\tau} + j_{ion}(x_i, t_j), & i = \overline{1, n-1}, \quad j = \overline{0, m-1}, \\ \frac{m_i^{j+1} - m_i^j}{\tau} = -\frac{m_i^j - m_0(V_i^j)}{\tau_m(V_i^j)}, & i = \overline{0, n}, \quad j = \overline{0, m-1}, \\ \frac{h_i^{j+1} - h_i^j}{\tau} = -\frac{h_i^j - h_0(V_i^j)}{\tau_h(V_i^j)}, & i = \overline{0, n}, \quad j = \overline{0, m-1}, \\ \frac{n_i^{j+1} - n_i^j}{\tau} = -\frac{n_i^j - n_0(V_i^j)}{\tau_n(V_i^j)}, & i = \overline{0, n}, \quad j = \overline{0, m-1}, \end{array}\right.$$

where
$j_{ion}(x_i, t_j) =$
$G_{Na}(m_i^j)^3 h_i^j (V_i^j - V_{Na}) + G_k(n_i^j)^4(V_i^j - V_k) + G_L(V_i^j - V_L),$
$i = \overline{1, n-1}, \quad j = \overline{0, m-1}.$
We express the next time layer values y_i^{j+1}, m_i^{j+1}, n_i^{j+1}, h_i^{j+1} using the values on the previous time layer

$$\left|\begin{array}{ll}
y_i^{j+1} = y_i^j + \frac{\tau}{C}(\frac{1}{Rh^2}(y_{i+1}^j - 2y_i^j + y_i^{j-1}) - j_{ion}(x_i, t_j)), & i = \overline{1, n-1}, \quad j = \overline{0, m-1}, \\
m_i^{j+1} = m_i^j - \frac{\tau}{\tau_m(V_i^j)}(m_i^j - m_0(V_i^j)), & i = \overline{0, n}, \quad j = \overline{0, m-1}, \\
h_i^{j+1} = h_i^j - \frac{\tau}{\tau_h(V_i^j)}(h_i^j - h_0(V_i^j)), & i = \overline{0, n}, \quad j = \overline{0, m-1}, \\
n_i^{j+1} = n_i^j - \frac{\tau}{\tau_n(V_i^j)}(n_i^j - n_0(V_i^j)), & i = \overline{0, n}, \quad j = \overline{0, m-1}.
\end{array}\right.$$

3.4 Numerical results

In order to close the system we need one initial condition for each of $m(x,t)$, $n(x,t)$ and $h(x,t)$, two boundary and one initial conditions for $V(x,t)$. For the initial conditions of $m(x,t)$, $n(x,t)$ and $h(x,t)$ are used their values at the resting potential.
During our work we applied different initial and boundary conditions for system (2.8) in order to investigate its behaviour. Our first implementation uses constant voltage for the first few seconds, big enough to cause an impulse, then we fixed $V|_{x=0} = 0$. We also assume that the voltage is fixed to 0 at the right boundary (i.e. the Dirichlet boundary conditions).
The approximations of the initial and boundary conditions are:

$$\left|\begin{array}{ll}
m_i^j = m_0(V_0) = 0.052932, & i = \overline{0, n}, \\
h_i^j = h_0(V_0) = 0.596124, & i = \overline{0, n}, \\
n_i^j = n_0(V_0) = 0.317677, & i = \overline{0, n}, \\
y_1^j = 100, & j = \overline{0, m/6}, \\
y_{n+1}^j = y_n^j, & j = \overline{0, m}.
\end{array}\right.$$

The results of the algorithm are presented in Figure 12. The solution is of type "traveling wave", i.e. after the forming of the impulse it travels through the axon. These boundary conditions do not describe well the real world, because the voltage can not momentarily drop from a 100 to 0. That leads to a non-smoothness of the solution (see Figure 13 and Figure 14). On

Figure 13 we can see the last moment of time where $V|_{x=0} = 100$ (the constant voltage we are applying) and on Figure 14 which represents the next point of time, that is when $V|_{x=0} = 0$. Also in the real process of propagation of a neural impulse the voltage at the ends of the axons is not fixed for every moment of time.
Due to this problems, we solve the system using Neumann's boundary conditions for the left and right boundary i.e. $\frac{\partial u}{\partial x}|_{x=0} = \frac{\partial u}{\partial x}|_{x=L} = 0$. In order to cause an impulse we must choose a different initial condition for $V(x,t)$. The one we chose is an approximation of the Dirac delta function, which is a function with value of zero everywhere except at zero where it is infinite. The boundary conditions are approximated using forward differences:

$$\left| \begin{array}{ll} m_i^j = m_0(V_0) = 0.052932, & i = \overline{0,n}, \\ h_i^j = h_0(V_0) = 0.596124, & i = \overline{0,n}, \\ n_i^j = n_0(V_0) = 0.317677, & i = \overline{0,n}, \\ y_i^1 = \frac{1}{a\sqrt{\pi}} e^{-a(ih)^2}, & j = \overline{0,m/6}, \\ y_{n+1}^j = y_n^j, & j = \overline{0,m}, \\ y_2^j = y_1^j, & j = \overline{0,m}. \end{array} \right.$$

Now we have a smooth function when $V|_{x=0} = 0$. The results of this experiment are good (see Figure 15).

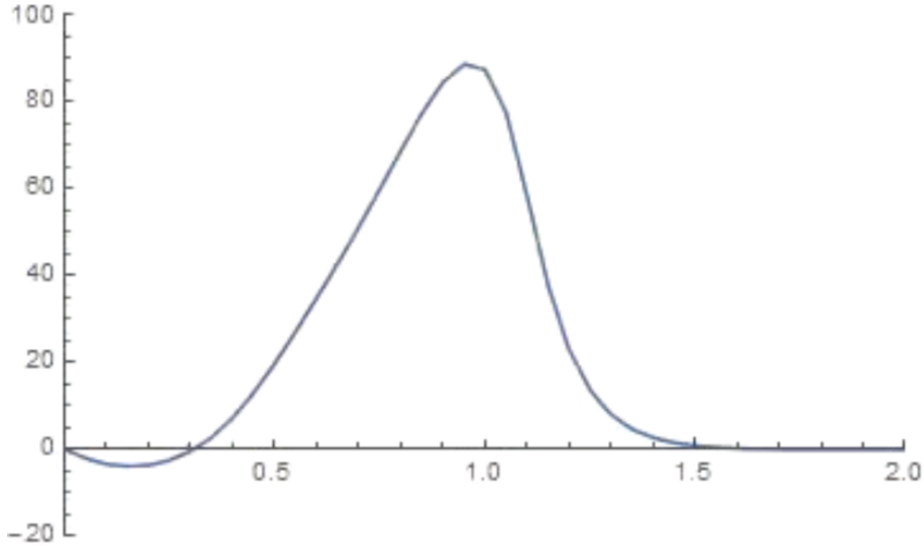


Figure 12: First implementation of neural impulse

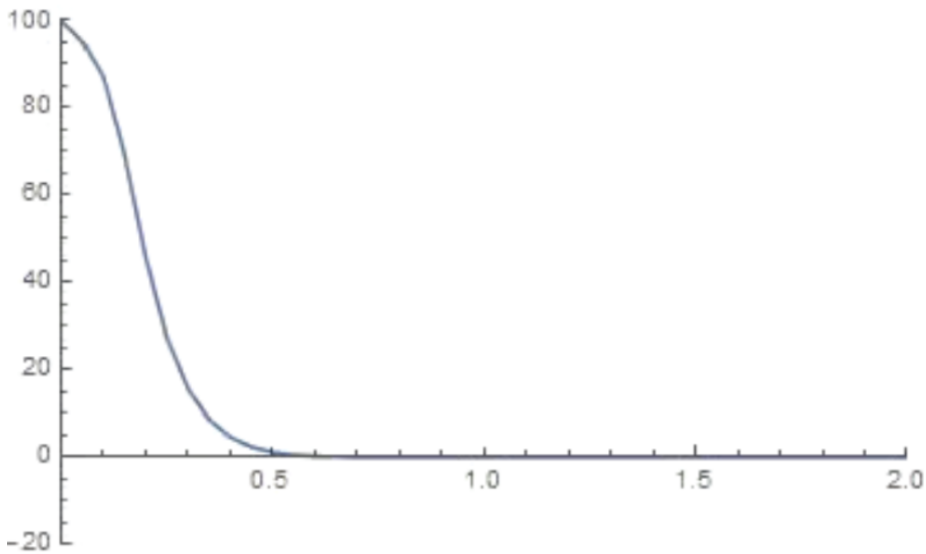


Figure 13: Last moment of time when $V|_{x=0} = 100$ for the first implementation of neural impulse

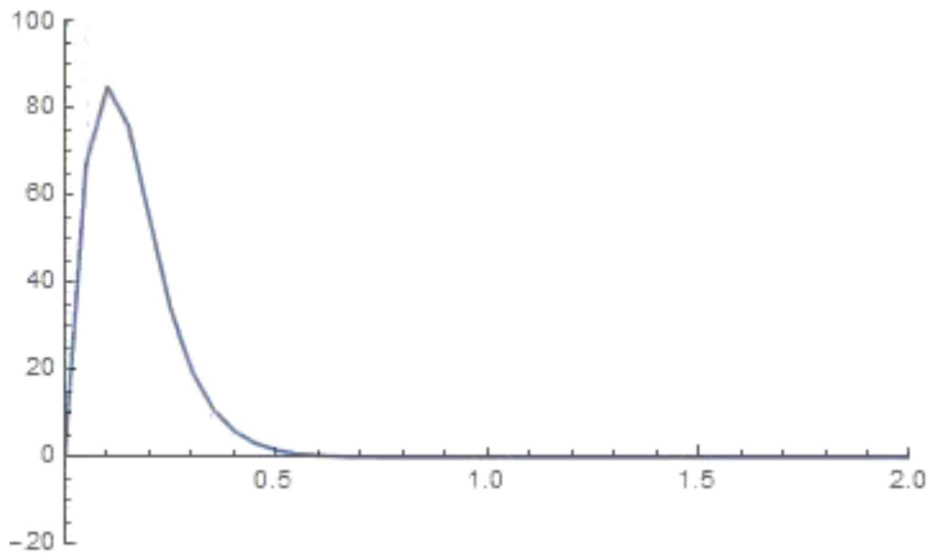


Figure 14: First moment of time when $V|_{x=0} = 100$ for the first implementation of neural impulse

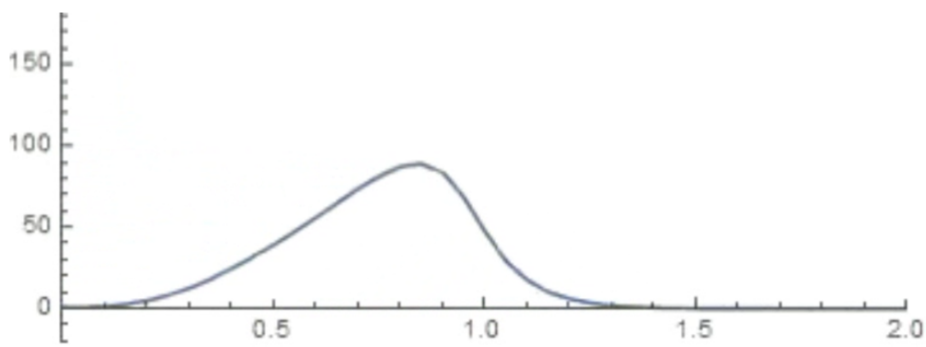


Figure 15: Second implementation of neural impulse

Although the local truncation error (LTE) of the "cable" equation is $O(h^2 + \tau)$, the LTE of the system is $O(h + \tau)$, because the derivatives in the boundary conditions are approximated using forward differences. In order to increase the order of approximation, we would like to modify the forward finite difference in such way that the LTE of the boundary conditions is $O(h^2)$. We shall expand $\frac{\partial V}{\partial x}|_{x=0} = 0$ using the Taylor series around $x = 0$ in order to compute the LTE and so we obtain:

$$\psi_{h,\tau} = \frac{1}{h}\left(u_0^j + \frac{\partial u}{\partial x}\bigg|_0^j h + \frac{\partial^2 u}{\partial x^2}\bigg|_0^j \frac{h^2}{2} + O(h^3) - u_0^j\right) = \frac{h}{2}\frac{\partial^2 u}{\partial x^2} + O(h^2).$$

In order to get our boundary condition to have local truncation error of $O(h^2)$ we can simply subtract $\frac{h}{2}\frac{\partial^2 u}{\partial x^2}$ from the approximation of the left boundary condition using forward difference:

$$\frac{u_1^j - u_0^j}{h} - \frac{h}{2}\frac{\partial^2 u}{\partial x^2} = 0 + O(h^2).$$

Assuming that the "cable" equation is fulfilled for $t = 0$, we can express the second order derivative of space with the derivative of time and using the approximation for the time derivative of $V(x,t)$ we obtain:

$$y_0^{j+1} = \frac{2\tau}{Ch^2} y_1^j + \left(1 - \frac{2\tau}{Ch^2}\right) y_0^j - \frac{\tau}{C} j_{ion}(x_0, t_j), \quad j = \overline{0, m-1},$$

which has the desired local truncation error of $O(h^2 + \tau)$. Analogically, for the right boundary condition we obtain:

$$y_n^{j+1} = \frac{2\tau}{Ch^2} y_{n-1}^j + \left(1 - \frac{2\tau}{Ch^2}\right) y_n^j - \frac{\tau}{C} j_{ion}(x_n, t_j), \quad j = \overline{0, m-1}.$$

Finally, we would like to model two consecutive impulses. We use the same initial condition and right boundary condition. The difference is with the implementation of the left boundary condition. When the first impulse passes through the axon, we model the second using an approximation of the Dirac delta function to generate a stimulus. Figure 16 and Figure 17 show the two impulses.

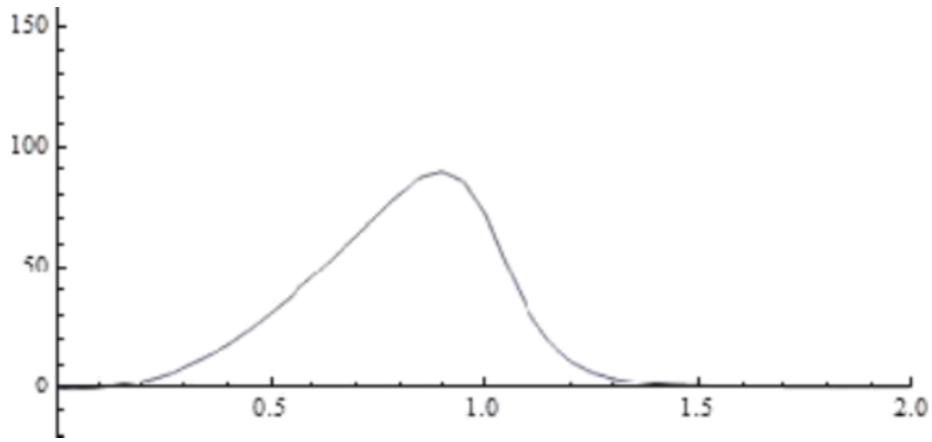


Figure 16: First impulse of the two consecutive impulses implementation

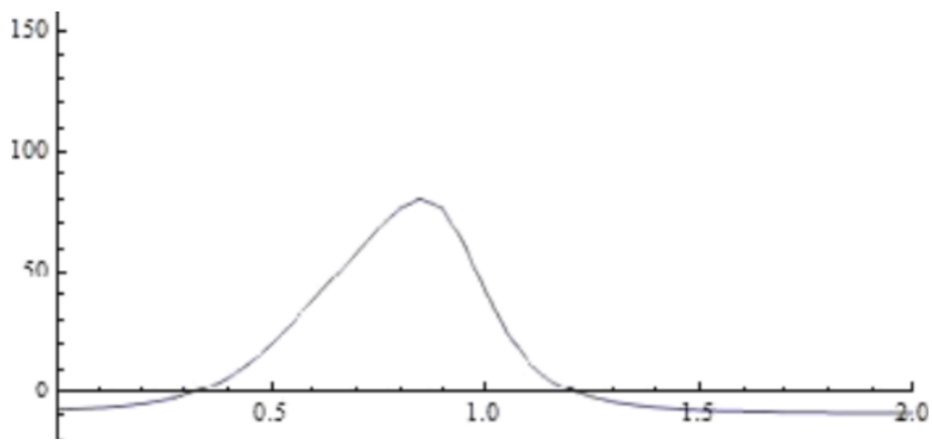


Figure 17: Second impulse of the two consecutive impulses implementation

4 Machine Learning and neural networks

4.1 Machine learning

Machine learning is the idea that there are generic algorithms that can tell you something interesting about a set of data without you having to write any custom code specific to the problem. Instead of custom code, you feed data to the generic algorithm and it builds its own logic based on the data. For example, one kind of algorithm is a classification algorithm. It can put data into different groups. The same classification algorithm used to recognize handwritten numbers could also be used to classify emails into spam and not-spam without changing a line of code. It is the same algorithm but it is fed different training data so it comes up with different classification logic(See Figure 18).

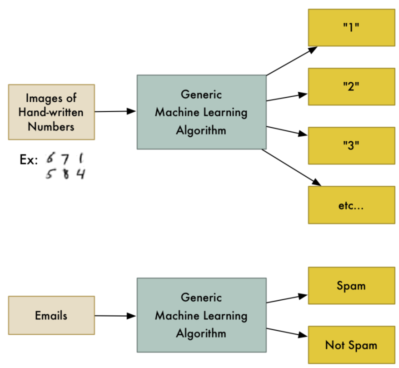


Figure 18: Example of generic machine learning algorithm

4.2 Neural networks

Artificial neural networks (ANNs) or connectionist systems are a computational model used in machine learning, computer science and other research disciplines. ANN is based on a large collection of connected simple units called artificial neurons(nodes), loosely analogous to axons in a biological brain. The idea behind neural networks is to have nodes that have links to each other similar to synapses between neurons.

4.2.1 A Single Neuron (Node)

The basic unit of computation in a neural network is the neuron, often called a node or unit. It receives input from some other nodes, or from an external source and computes an output. Each input has an associated weight (w), which is assigned on the basis of its relative importance to other inputs. The node applies a function f (defined below) to the weighted sum of its inputs as shown in Figure 19: The network in Figure 19 takes numerical

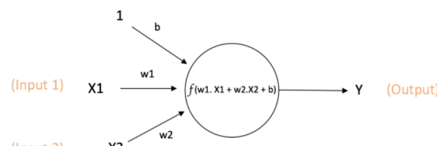


Figure 19: Neuron network's node

inputs X1 and X2 and has weights w1 and w2 associated with those inputs. Additionally, there is another input 1 with weight b associated with it. The output Y from the node is computed as follows:

Output of neuron $= Y = f(w1 * x1 + w2 * x2 + b)$

The function f is non—linear and is called the Activation Function. The purpose of the activation function is to introduce non—linearity into the output of a node. This is important because most real world data is non linear and our purpose is to define an algorithm so that the nodes to learn these non linear representations. Every activation function (or non-linearity) takes a single number and performs a certain fixed mathematical operation on it.

4.2.2 Feedforward Neural Network

The feedforward neural network was the first and simplest type of artificial neural network(See Figure 20). It contains multiple neurons (nodes) arranged in layers. Nodes from adjacent layers have connections or edges between them. All these connections have weights associated with them.

- Input nodes: provide information from the outside world to the network and are together referred to as the "Input Layer"; no computation is performed in any of the Input nodes —- they just pass on the information to the hidden nodes.
- "Hidden" nodes: have no direct connection with the outside world (hence the name "hidden"); perform computations and

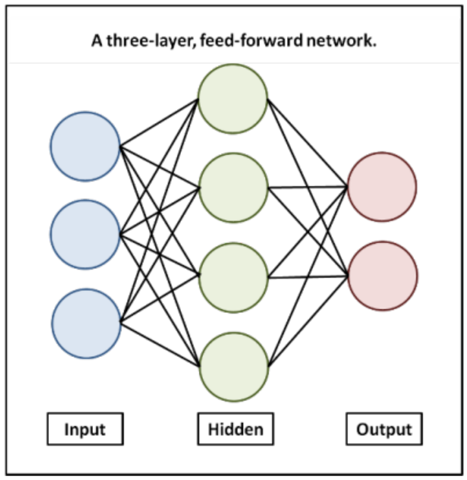


Figure 20: Fred-forward neural network

transfer information from the input nodes to the output nodes; a collection of hidden nodes forms a “Hidden Layer”.

- Output nodes: collectively referred to as the “Output Layer” and are responsible for computations and transferring information from the network to the outside world.

4.2.3 Real estate agent problem

Let us look at the following example — we are searching for algorithm which gives us the real house price based on given details(bedrooms, square footage and neighbourhood). See Table 1 on Figure 21.

Bedrooms	Sq. feet	Neighborhood	Sale price
3	2000	Normaltown	$250,000
2	800	Hipsterton	$300,000
2	850	Normaltown	$150,000
1	550	Normaltown	$78,000
4	2000	Skid Row	$150,000

Figure 21: Table 1 including example data for real estate agent problem

The searched algorithm should determine the real price of a house with given parameters. See Table 2 on Figure 22.

Bedrooms	Sq. feet	Neighborhood	Sale price
3	2000	Hipsterton	???

Figure 22: Table 2 including example data for real estate agent problem

We would like to model the price as follows:

Price = w1 $*$ sq.feet + w2*bedrooms + w3*neighbourhood

Our algorithms should determine right values for w1, w2 and w3. But how are the right weight values calculated? We will cover this topic in the next section.

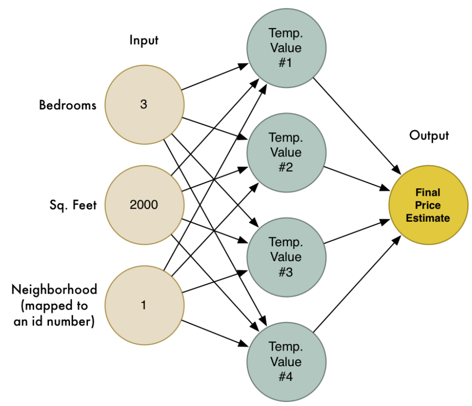


Figure 23: Neural network representing real estate agent problem.

4.2.4 Neural Networks Learning: The Back-Propagation Algorithm

The process by which a neural network learns is called the Backpropagation algorithm. We submit input data and compare outputs with the desired ones. In networks, the error is called - difference. New weights are extinguished by using an equation based on previous weights, input data, error rate and learning rate. It is impossible to try every number of combinations for values of w1, w2 and w3 so we need equation which represents how wrong our price estimating function is for the weights we currently have set. This is the cost function which is defined as follows:

$$Cost = \frac{\sum_{n=1}^{500}(MyGuess(i) - RealAnswer(i))^2}{500 * 2}$$

. This equation represents how wrong our price estimating function is for the weights we currently have set. ó is what represents your current weights. J(ó) means the 'cost for your current weights':

$$J(ó) = \frac{1}{2m}\sum_{n=1}^{m}(h(i) - y(i))^2$$

The graph of the cost function looks like a bowl. The vertical axis represents the cost. The coordinates of the point represent weights for square footage and number of bedrooms. In the graph below the points in blue colour represent values of weights for which the error of the function is the smallest. The points in red colour represent values for which the error of the function is the biggest. When the weights which represents the lowest point are found, the solution for the problem is also found.

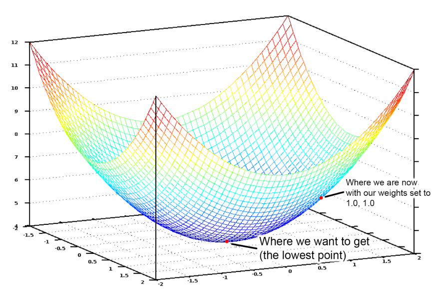


Figure 24: The cost function graph.

Machine learning algorithms is only effective when the input data is proper. If the data is improper there will be no meaningful result. For example if we use types of plans in each house to determine its price then neural network would not succeed in finding a solution. This is because real relationship between types of plants and price does not exist(see Figure 25).

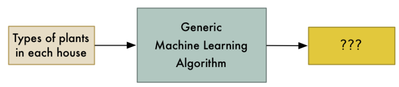


Figure 25

Conslusion

This article describes the model of Hodgkin-Huxley, which was obtained for a specific species — a squid. During this work we derived the model, based on the charge conservation principle and Ohm's law. It contains 4 PDEs — 1 of second order in terms of a space variable x and 3 of first order in terms of time t. The final system cannot be solved analytically and it was approximated using numerical methods. We experimented with different initial and boundary conditions (Dirichlet and Neumann) from which different behaviour of the impulse was obtained. After the visualization of the propagation of the impulse, we modelled two consecutive ones — when the first one passes the axon we model the second one using the Dirac delta function. Plots of the final results are provided.
We have considered two possibilities for future work — machine learning and neural networks.

References

[1] A. Scott *Neuroscience. A mathematical primer*

[2] KS Cole *Membranes, Ions and Impulses* University of California, 1968.

[3] https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471