

NeuroLab

Assignments and Notes

Luis F. Schettino

Psychology Department

and Neuroscience Program

Lafayette College

Easton, PA 18042

Index

Page Number	Topic
3	Introduction: a work in progress
4	Working with NetLogo
7	Potentials
9	Temporal Integration
10	Antagonism
12	Axonal Guidance
14	Cortical Development
16	Tactile Lateral Inhibition
18	Edge Detection
21	Propagation of the Action Potential
22	The Synapse
23	Back Propagation Network
25	Musical Circuit
27	Kitaoka Sakura
28	References

Introduction

A work in progress

NeuroLab was born out of the need, oftentimes felt during class, of a visual means to convey the dynamic nature of neural processes in a way that captures the complexity arising from a large number of relatively simple elements that interact following a few basic rules.

My early experiences with multi-agent Logo languages, starting with StarLogo, had convinced me that they were the ideal tools for simulating such processes. The robustness and continuous development of NetLogo (Wilensky, 1999), which includes numerous 'agent-set' functions made it an easy choice for my first actually usable student assignments.

The present set of simulations has been used successfully for a few years. What that means is that my students find them easy to understand, easy to use and compelling enough to engage with them. NetLogo has a very small learning curve and its interface is very intuitive. In general, most of the learning carried out in a given simulation is not about the software but about the concepts that the simulation was designed to present.

It is important to stress that while the simulations in *NeuroLab* do a good job of showing the basic concepts of neural processes, they are not meant to provide a physiologically-sound, quantitative experience such as it may be obtained from packages like *Neuron* (neuron.yale.edu) or *Neurons in Action* (neuronsinaction.com). Perhaps the best way to characterize the simulations in *NeuroLab* is that they are educational toys that students in introductory classes can both enjoy and learn from.

As I continue to work and streamline the simulations and also add new ones to the group, I would appreciate any comments and/or corrections you may have. I would also like to know if you have an idea about a process that you would like to see simulated. Please get in touch with me at schetttil@lafayette.edu to explore the possibilities.

An article describing *NeuroLab* (Schettino, under review) and providing more information about its use in my Introduction to Neuroscience course is currently under review by the Journal of Undergraduate Neuroscience Education (JUNE). The *NeuroLab* simulations are available for download at <http://sites.lafayette.edu/schetttil/neurolab/>.

This document contains descriptions of the simulations included in *NeuroLab* as of February 2014. It also contains the full text of the assignments (in boxes) as I provide them to my students or an explanation if they are employed only as demonstrations in class.

Thank you for trying *NeuroLab*!

Luis F. Schettino

Introduction to Neuroscience

Working with NetLogo

NetLogo is a software tool to explore multi-agent processes. That is, processes that involve many similar agents that follow just a few relatively simple rules. For example, you may want to see what happens when a group of ants work together, or how is it that birds organize into a flock.

The main power of NetLogo is to actually show you in a graphical way the behavior of all agents as it unfolds. It also allows you to manipulate some of the variables that influence the agents' behavior, and you can also observe the result of your changes right away in your simulation.

In our course, we will use NetLogo to explore the behavior of neural processes, from the molecular to the network levels. These processes, as you may gather, are difficult to observe *in vivo* (live) and so we need to reproduce them in order to understand them better. NetLogo allows us to do precisely that.

This short tutorial will guide you as you interact with the NetLogo interface in one of the Models that comprise the standard library.

- 1- Start NetLogo by double-clicking on the icon on the desktop. This will open a window that contains, on the right, a black window. This is the NetLogo "world", where the agents carry out their behavior. Since this is a blank project, there won't be anything else on the window.
- 2- Go to File→Models Library. This action will bring up the Models Library window. The Library is a group of sample projects that you can open and explore. On the left hand side of the window you will see a folder view of the library. Click on "Sample Models → Biology→Flocking". This will open up a window like that presented in Figure 1.

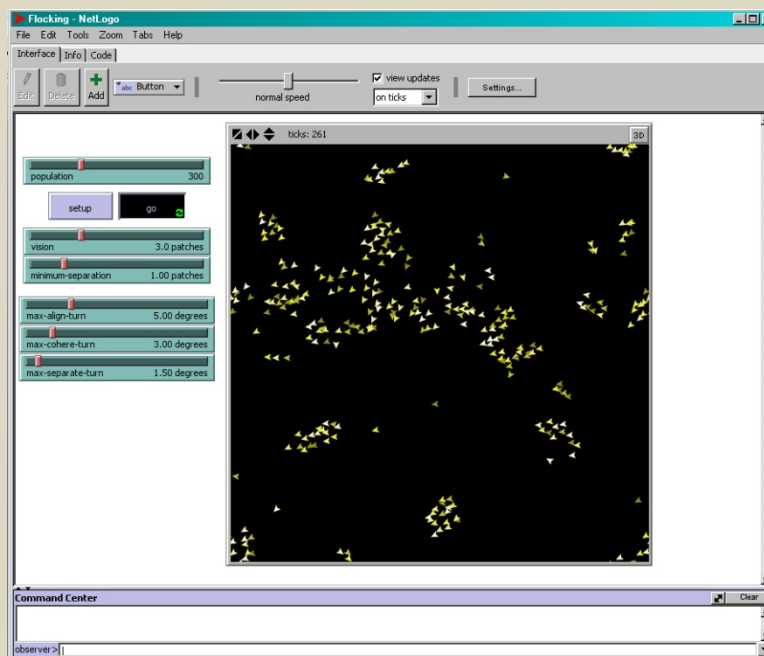


Figure 1. Flocking simulation in the NetLogo Models Library.

Typically, NetLogo models have two main buttons: a **Setup** button and a **Go** button. The Setup button prepares the NetLogo world by placing its components within it. The **Go** button causes the agents in the world to begin their activities.

Before you start using a model, notice that each model will have three “tabs” right below the main menu. The tabs are named: *Interface*, *Info* and *Code*. The Interface tab shows you the NetLogo world and its controls. The Information tab has a description of the Model that tells you what the model simulates, how it does it, how to use it and some interesting things to notice and to try with it. Finally, the procedures tab contains the actual code that controls the model.

One final point you want to know about before you start is the slider at the top margin of the NetLogo world. This slider controls the speed at which the Model runs. If at any time you want to see things in slow motion in order to understand them better, simply slide the control to the left.

In order to start the Flocking Model, which simulates the way that birds organize themselves while migrating, use the “population” slider to select how many “birds” you want in your simulation. You can also modify the other controls right now (if you have read the Information tab now you know what each one does) or you can modify them as you watch the model run.

As soon as you press the **Setup** button you will notice that your “birds”, represented by arrowhead shapes, are scattered all over the world. Once you press the **Go** button the birds will start to come together and fly in the same direction. As you manipulate the controls you will see the groups of agents grow or get smaller depending on the variable you change.

Notice that in this model the world “wraps” both horizontally and vertically. That means that when a bird exits the world through the left margin, it will come back in through the right margin.

Well, that’s about it! Each model will have its own set of controls and behaviors and, as you can probably tell already, every single run is different from the ones before.

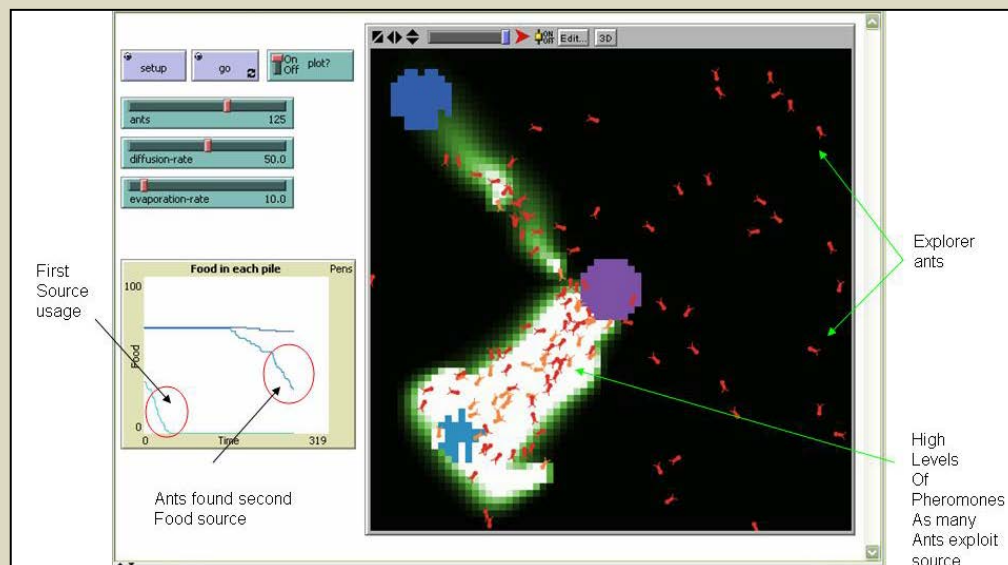


Figure 2. An example of what an assignment report should look like.

Working through the assignments.

Whenever a Model is assigned, you will need to provide a short description of your exploration. Each assignment will give you specific directions, but in general, here is how to do it:

1-Run the assigned model and carry out the different manipulations. For example, “run the Flocking model with 200 agents, 3.5 patches of vision, etc.” When you have run the model and understood it, stop it by releasing the **Go** button and make a copy of your computer screen by pressing on the “Print Screen” button of your keyboard (all the way to the top, close to the F keys). This will save a picture of your screen to the clipboard. Windows 7 also provides a ‘snipping tool’ accessory that clips small regions of the screen.

2-Open PowerPoint and Paste the screen picture on a slide. Using PowerPoint’s tools (arrows, lines, text), describe what you did and why you obtained those results. Use font size 12 on your text in PowerPoint. Figure 2 is a sample of how an assignment should look like. You can also crop plots or portions of the world to stress particular information. Also notice that you can right-click the world or any plot and copy just that image to be pasted in your PowerPoint. Plots are particularly important, so make sure you copy and describe them.

3- Print the page and turn it in whenever it is due. Don’t forget to write your name on it. You can save all of your assignment slides in the same PowerPoint presentation. If you want to save a specific PowerPoint slide as an image file go to “File → Save as...” and select JPEG as the file format in the dialog box (under “Save as type”). This will save the current slide as an image.

About NetLogo.

NetLogo was developed by Uri Wilensky at the Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL. The NetLogo software, models and documentation are distributed free of charge for use by the public to explore and construct models.

Potentials

The potentials simulation allows the user to observe the establishment of the resting potential and the production of the action potential in a simulated membrane. Switches can turn the sodium-potassium switch on and off as well as add the effects of channel blockers such as TEA and TTX. Plots of the cell's potential as well as the ionic influx/efflux are also available.

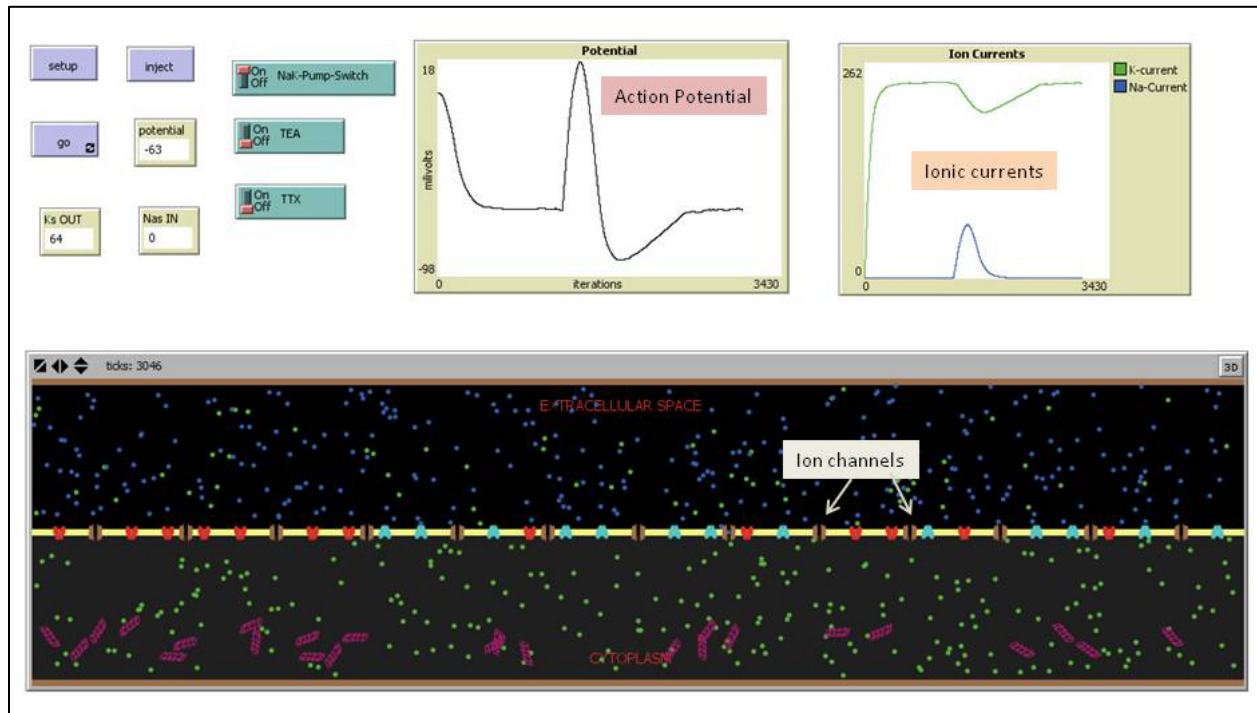


Figure 3. Example of a run of the Potentials simulation.

Introduction to Neuroscience

1st Assignment: Potentials

Now that you have played around a bit with NetLogo, it is time to put your knowledge to use. The first assignment using NetLogo will be running a simulation of the resting and action potentials of a neuron. Find the file *Potentials.nlogo* in Moodle, download and open it.

Read the *Info* tab carefully and run the simulation in order to answer the following questions. Don't forget to turn in a screen image of the exercises with your comments as described in the "Working with NetLogo" document.

Exercises:

1- Press the **Setup** button. You should see a simulated membrane running lengthwise in the NetLogo window plus a large number of blue and green dots. Now press the go button and follow the plots. On the top you have the charge inside the neuron with respect to its outside, that is, the potential. On the bottom you have a graph of the amounts of K ions (green) and the Na ions (blue) inside the membrane. As you can see, the charge inside the membrane goes down as some K ions flow out of the neuron. After a few seconds the charge and the flow stabilize around -70mV. This is equivalent to the resting potential of the membrane.

2- Wait for the charge to stabilize. When it does, press the **inject** button. This is equivalent to injecting an electrical current into the cell through an electrode and bringing it to threshold. The result will be an Action Potential and ion flow through the membrane. Look at the respective inflow of Na ions and outflow of K ions. Explain why they happen. You may want to slow down the simulation to see the timing of the channels as they open and close.

3- Wait for the resting potential to stabilize again. What happens if you switch OFF the **Na-Kpump** switch? Why? Press **inject** once again and observe the response of the membrane. Explain what happens.

4- Turn the pump back on. Wait for the resting potential to stabilize. Now turn on the **TTX** (Na channel blocker) switch on. Then press the **inject** button again. What happens? Why?

Temporal Integration

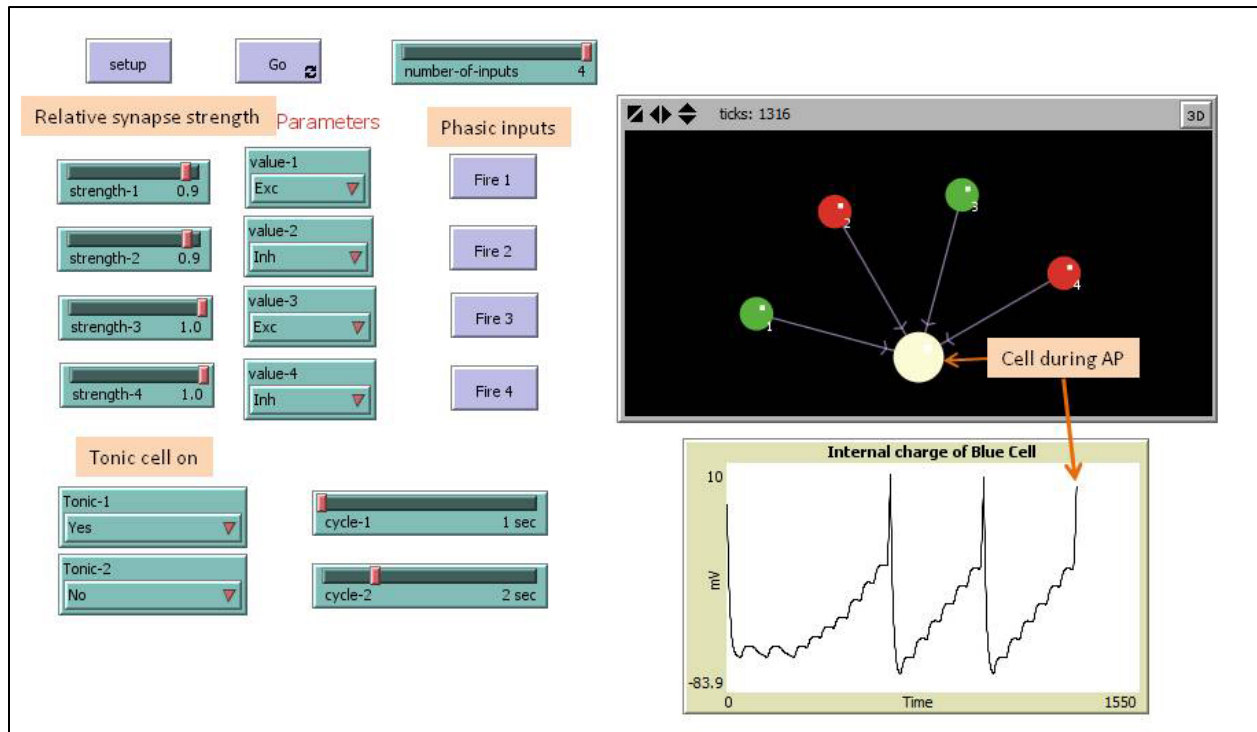


Figure 4. The temporal integration simulation interface

Introduction to Neuroscience 2nd Assignment: Temporal Integration

This assignment will allow you to explore the way in which neurons “integrate” or add up their inputs and decide whether they too should produce an action potential.

Open the *Integration.nlogo* model. Go to the *Info* tab and read carefully the information provided there. Once you have done so, go back to the Interface tab and run the simulation. Since you cannot control precisely which neuron is inhibitory or excitatory, simply click on the **Setup** button until you come up with a configuration you like.

Exercises:

- 1- Set one of your excitatory inputs (neurons 1 or 2) to Tonic. At what strength and cycle do you have to set that input in order to make the blue neuron to fire in 10 to 12 EPSPs (not more, not less)? Here you may want to slow down the simulation to give yourself enough time to interact with it.
- 2- Set one of your inhibitory inputs to Tonic. Set its cycle to 3 sec and its strength to 0.8. Press the Go button. Then press repeatedly the **Fire** button of an excitatory neuron. How many times do you have to press it as fast as you can in order to get the blue neuron to fire? Why?
- 3- If you have two tonic excitatory cells, what kind of input is more efficient at getting the blue neuron to fire: both inputs with the same cycle at 4 sec or one neuron with a cycle of 4 sec and the other in a cycle of 3? (Hint: hover the mouse over the plot to see the values on each axis.)

Explain why this is the case.

Antagonism

The antagonism simulation allows the user to observe neurotransmitter binding to receptors in the postsynaptic membrane followed by the opening of ionic channels and postsynaptic potential (PSP) production. The user can select from four different types of molecules to modulate the number of PSPs produced. The four types of molecules are agonists, competitive antagonists, non-competitive antagonists and enzyme-antagonists. These last bind temporarily to enzymatic molecules in the synaptic gap and prevent neurotransmitter inactivation for a short period.

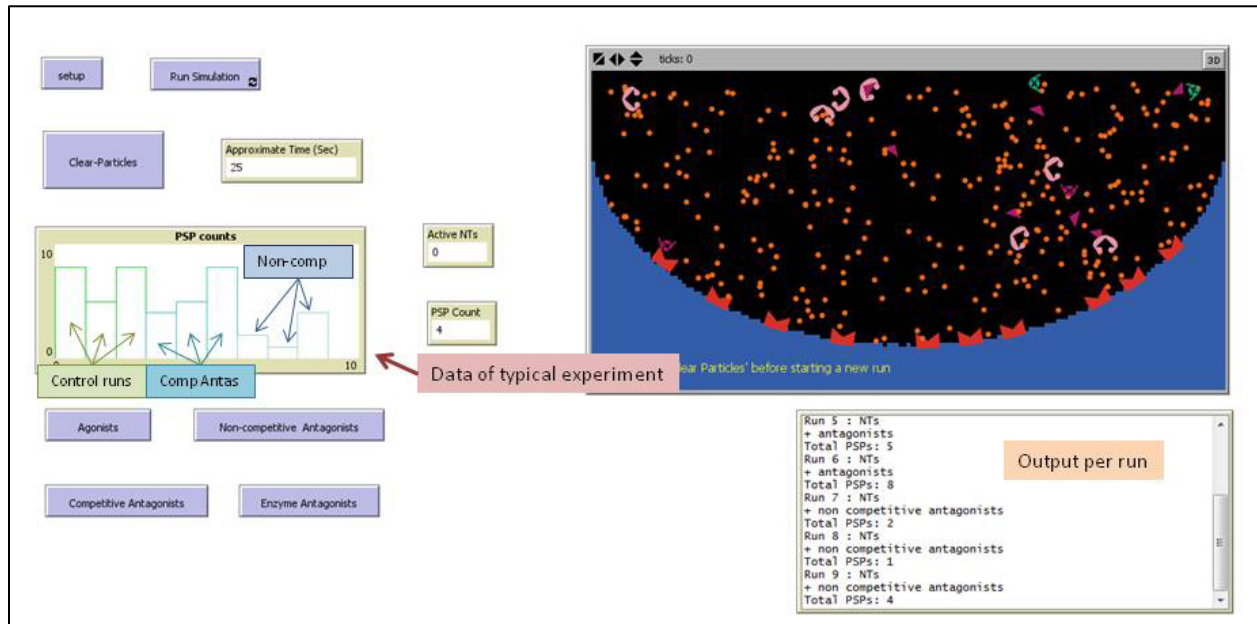


Figure 5. The antagonism simulation interface

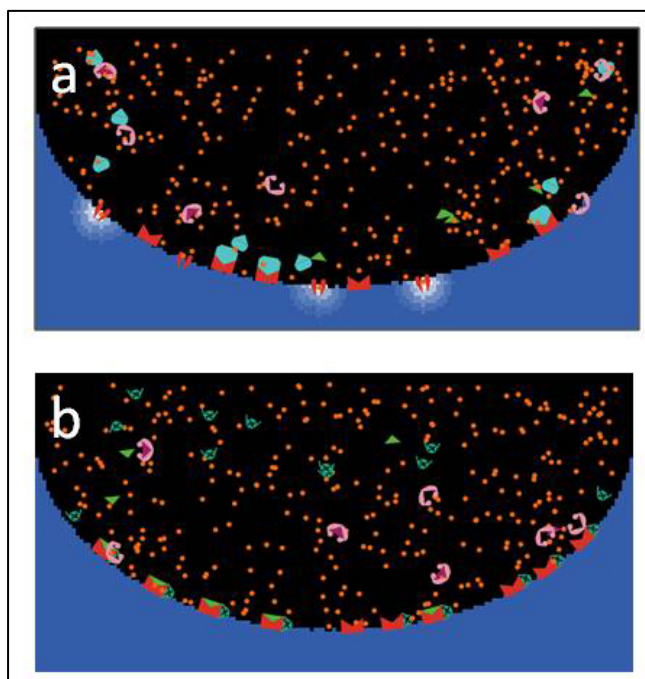


Figure 6. Two typical runs of the simulation: a) competitive antagonists (cyan), b) non-competitive antagonists (dark green).

Introduction to Neuroscience

3rd Assignment: Antagonism

This assignment will help you explore the dynamics of neurotransmitter and drug molecules as they compete for receptors on the postsynaptic membrane. You will be running a short experiment to test the effects of different ligands (small molecules that bind to large biomolecules, such as cell receptors) on the number of postsynaptic potentials (PSPs).

Open the *Antagonism.nlogo* model. Go to the *Info* tab and read carefully the information provided there. Once you have done so, go back to the *Interface* tab and run the simulation. You may want to slow down the simulation a little to be better able to see all the interactions taking place.

Exercises:

1- Press the **setup** button to produce the postsynaptic membrane and molecules within the synaptic cleft. Every time you press the **Run Simulation** button, 15 active neurotransmitter (NT) molecules will be produced. The NTs will bind to the receptors in the postsynaptic membrane, causing PSPs. Once the active NTs are inactivated, the run will end.

Run the simulation three times pressing the **Clear Particles** button after each one. As you will notice, the number of PSPs will vary somewhat each time. This is due to the probabilistic nature of the process. On a separate piece of paper obtain the average of the three runs. This will be your *control* runs.

2- Now add competitive antagonists by pressing the appropriate button and then run the simulation. Run three trials of the same condition and calculate the average PSP number.

3-Now do the same as above but using non-competitive antagonists. Report your averages as a bar graph (which you can produce in Excel and copy to your report) and discuss whether your results were those you expected.

4- Finally, choose a type of drug that you'd want to experiment with: agonists or antagonists to the receptors or enzyme antagonists. Run the simulation again. What changes do you notice? Why do they happen?

Exercises:

1- Press the **setup** button to produce the postsynaptic membrane and molecules within the synaptic cleft. Every time you press the **Run Simulation** button, 15 active neurotransmitter (NT) molecules will be produced. The NTs will bind to the receptors in the postsynaptic membrane, causing PSPs. Once the active NTs are inactivated, the run will end.

Run the simulation three times pressing the **Clear Particles** button after each one. As you will notice, the number of PSPs will vary somewhat each time. This is due to the probabilistic nature of the process. On a separate piece of paper obtain the average of the three runs. This will be your *control* runs.

2- Now add competitive antagonists by pressing the appropriate button and then run the simulation. Run three trials of the same condition and calculate the average PSP number.

3-Now do the same as above but using non-competitive antagonists. Report your averages as a bar graph (which you can produce in Excel and copy to your report) and discuss whether your results were those you expected.

4- Finally, choose a type of drug that you'd want to experiment with: agonists or antagonists to the receptors or enzyme antagonists. Run the simulation again. What changes do you notice? Why do they happen?

Axonal Guidance

The axonal guidance simulation assignment asks students to find out the functions of each of four different factors in the path-finding mechanisms for a set of axons. Each factor can be turned on or off using simple switches and the effects of each manipulation is readily observable.

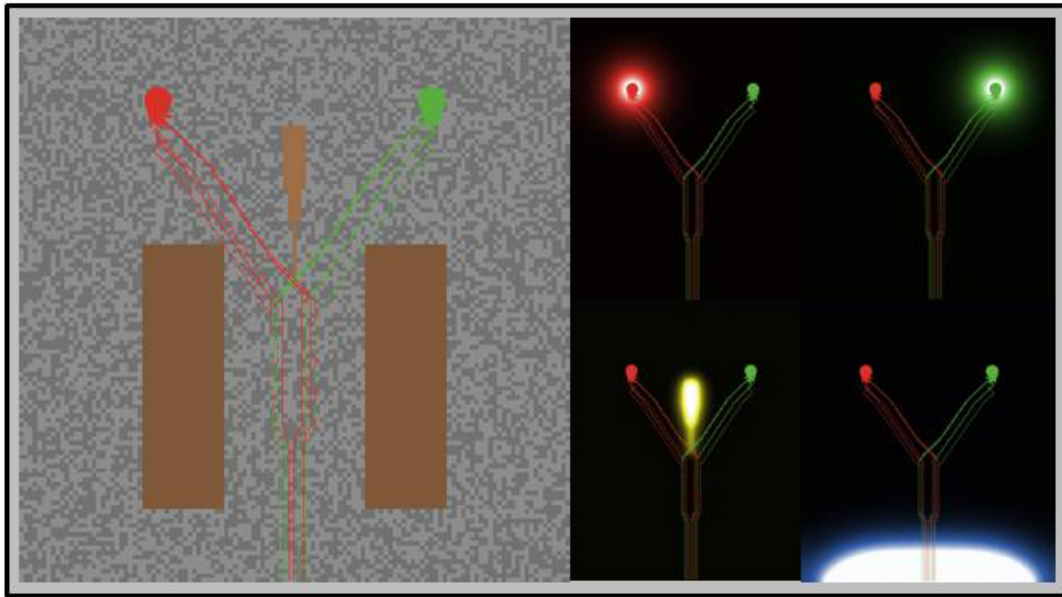


Figure 7. The axonal guidance simulation (left). The gradients of different attractant and repellant factors that guide the axons to their targets (right).

Introduction to Neuroscience

NetLogo Assignment: Axonal Guidance

This simulation looks at the way that axons find their targets during development. Axons produce a 'growth cone' on their tips that look for particular substances thanks to receptors on its membrane.

Now it is your turn to do a little experimentation. Open the *Axonal Guidance.nlogo* simulation and run it as usual, by pressing first **Setup** and then **Go**. As you can see, from a set of 20 axons (10 red, 10 green) with random locations within a 'fiber tract', their growth cones begin moving towards the top and decussate (separate) by color and continue moving towards the target that matches their own color.

How does the system work? By now you know about axons being guided through chemical cues in the form of attractants and repellants. This simulation is based on the same principles. Your job will be to find out what each of the substances at play in this simulation is doing to guide the axons to their developmental targets.

Exercise:

The switches on the left hand side of the interface panel allow you to turn on and off the substances released by the targets, the midline and the side structures. Once you run a trial (when the axons stop moving), you can then press the "show-factor" buttons to see the diffusion of each one of the substances. Note that even though it may not be visible, the substance may nonetheless reach farther than it seems. The iteration counter ('ticks') can give you a sense of how long the system has been running.

By turning on and off the substance-release switches, you can observe changes in the behavior of the axons.

Questions:

- What does each substance do? (is it an attractant or a repellant?)
- How did you figure it out?

Cortical Development

The cortical development simulation shows the creation of a layered structure through the coordination of factor release, neuronal migration and differentiation. The user can implant labeled cells to determine their fate and learn the constraints that give rise to the typical appearance of cortical tissue.

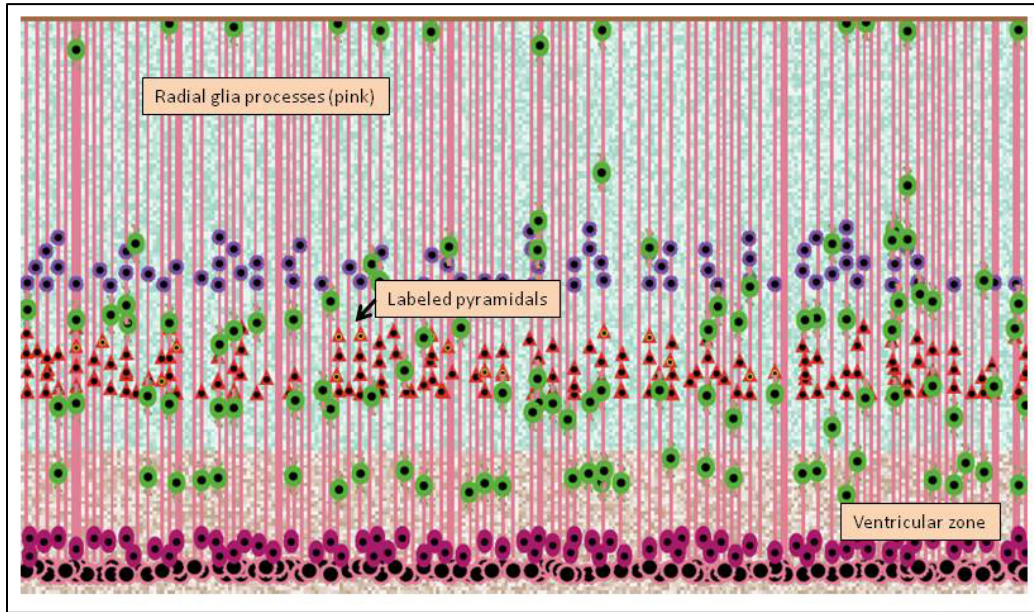


Figure 8. The cortical development simulation showing different cellular types as the process unfolds.

Introduction to Neuroscience 5th Assignment: Cortical Development

This assignment will help you explore the mechanisms of cell-to-cell interaction that result in the development of the characteristic layered structure of the mammalian cortex. In this simulation, only three of the six layers of the cortex are shown (layers III-V).

Open the *Cortical Development.nlogo* model and read the *Info* tab. Go back to the *Interface* tab and run the simulation once by pressing **Setup**. Then press **Go**. This will begin the process by having the Radial Glia (pink cells) develop projections to the top border of the tissue (the black space is meant to be empty space). As time passes, the tissue will grow and small yellow cells will move upwards.

Cellular migration begins when the purple cells (progenitors) divide and some of their daughter cells (precursors, green) begin to climb up the radial glia processes. At a given level, the green cells will begin to differentiate into either pyramidal (red or blue) or stellate cells (purple). This is due to the production of specific chemical factors at the different layers. You can see the factors by selecting the factor on the chooser and pressing the show-factor button. At the end of the process, all other cells but the pyramidal and stellates die away, giving rise to the mature cortical structure.

In order to study the cellular fate of precursors, researchers commonly add radioactive labels to cells transplanted into the ventricular zone tissue at different developmental stages. Once development is completed, the labeled cells can be detected to determine their location and thereby infer the effects of developmental timing on cellular fate.

Exercise:

In order to test the timing in which cellular fate is established, you will use the *Implanted-Type* chooser and the *implant* button to select the type of cell you would like to implant as a labeled cell, either a young precursor (a cell that has divided from the progenitor at early stage of development) or a young or old progenitor (progenitors obtained from an early or late stage of development). In actual experiments, these cells come from embryos at early or late developmental stages. Labeled cells will have a yellow ring around their nucleus.

- 1- Press the **Setup** button to clear the world and select the type of cell you'd like to implant. Then press the **Develop** button to start the development process. Press the implant button at the appropriate timing to have the labeled cells appear. Wait for the full development cycle to finish and then enter on the table where labeled cells ended up (you can use simple Y/N notation in the layer columns).
- 2- In order to determine when cellular fate is established you will need to have evidence that supports unequivocally your conclusions. When is the fate of cells determined, at the Precursor or at the Progenitor state? How do you know that? Provide a couple of run images in your report as well as the full table below.

Cell Type	Timing	Labeled Cortical Cells			
		ticks	Layer V (red)	Layer IV (purple)	Layer III (blue)
Young Precursor	Early	250			
Young Precursor	Late	500			
Young Progenitor	Early	250			
Young Progenitor	Late	500			
Old Progenitor	Early	250			
Old Progenitor	Late	500			

Hints:

You can increase the speed of the simulation during your experimental runs. But make sure that your implantation timing is close enough to the tick values described.

Tactile Lateral Inhibition

The tactile lateral inhibition simulation shows the process through which the borders of tactile stimuli are detected against the skin. The user can turn inhibition on and off to observe the effects of this manipulation on the activity of the cells and on the 'experience' of the organism. In addition, a comparison between perception of a single versus two stimuli can be conducted with and without inhibition in order to observe the results of each experimental condition.

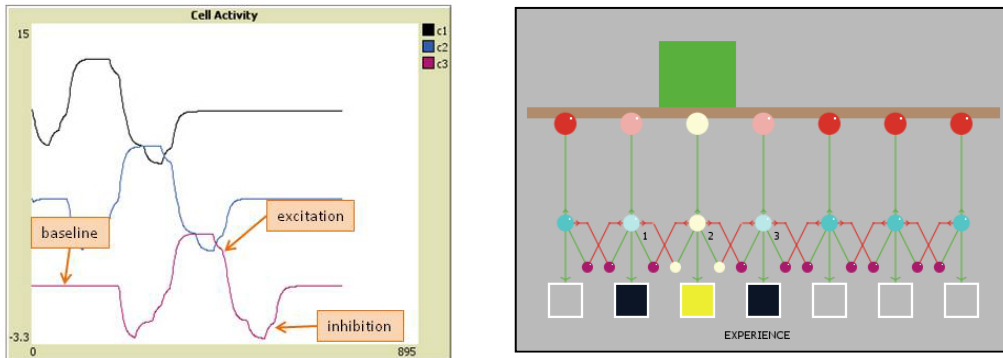


Figure 9. Cell activity of the neurons labeled 1 to 3 in the simulation showing the typical 'Mexican hat' profile due to lateral inhibition (left). The simulation window showing the relative level of activation of tactile receptors and their efferents (right).

Introduction to Neuroscience Assignment: Tactile Lateral Inhibition

As you know by now, lateral inhibition is a widespread process in the nervous system. In the tactile (touch) system it allows the organism to detect the regions of the skin that are being currently stimulated by creating a gradient of activation between neighboring cells. This is useful because it helps the organism localize precisely the location of stimuli on the skin.

Open the *Tactile Lateral Inhibition.nlogo* simulation and read the *Info* tab to get a feel for how it works. In your report, answer the questions you will find below.

Exercises:

1- Press **Setup**. What you will see is a light brown bar that represents the skin with a number of red receptors embedded in it. To the left of it, there is a green rectangle that will serve as a tactile stimulus. You also see a layer of light blue neurons and a layer of magenta inhibitory interneurons. Finally, at the bottom, is a set of rectangles that will represent the 'experience' of the organism as the skin is stimulated.

2- Press the **Go** button. This will activate the system and you will see that the receptor right under the green stimulus turns pale yellow. The same thing happens to the blue neuron it projects to and to the magenta interneurons that are the blue neuron's efferent. This represents the activation of that information channel as it travels towards the brain and the parallel inhibition of the neighboring blue neuron. The experience of the organism is represented by the 'experience rectangle' turning bright yellow and its neighbor turning dark blue.

3- If you look at the plot on the left-hand side, you will see that there are three traces. Each one represents the relative activity of one of the three neurons labeled 1, 2 and 3. While the traces are 5 units apart in order to

make them easily observable, you should only take into account the fluctuations above and beyond each cell's baseline (the value at which they settle without receiving direct activation).

4-Using the arrow buttons, move the stimulus left and right to observe its effects on the experience rectangles and on the relative activity on the plots. You will notice that as the stimulus comes close to a receptor, its corresponding experience rectangle turns dark blue.

5- Now press the 'remove-inhibition' button and compare the activation of the network without the inhibitory processes. What do you see? How would this affect the experience of the organism?

6-Press **Setup** again and flip the 'slide-object' switch to on. When you press **Go** the object will move sequentially from left to right, stimulating the network as it goes. Right-click on the plot, copy it and paste it on your PowerPoint. Then repeat the process but this time remove the inhibitory cells before you press **Go**. Once again, copy and paste the plot right below your original plot so that they are perfectly aligned. What do you see? How would the difference that you observe affect experiential perception of the stimulus?

7- Turn the **slide-object** switch off and turn the **two-objects** switch on. Press setup. This will produce two small objects on the skin. Press **Go** to set the simulation on. As you move the objects towards the right you will notice that two non-adjacent experience rectangles turn bright yellow and they are surrounded by black rectangles. This represents the sensation by the organism of two distinct objects on the skin.

8-What happens to the experience rectangles if you press the remove-inhibition button? Why? What happens to the network? For one extra point: Imagine that you are looking at the skin from above. Each object activates a circular area of the skin below it. What would the area activated by the two objects look like with and without inhibition? Draw a small diagram.

Edge Detection

The edge detection simulation exemplifies the process of lateral inhibition in the visual system. The user draws a simple image which is then processed using a basic 'Mexican hat' activation algorithm. The user can observe the effects of the drawing directly on the window observing the activation differences occurring in high contrast regions of the image.

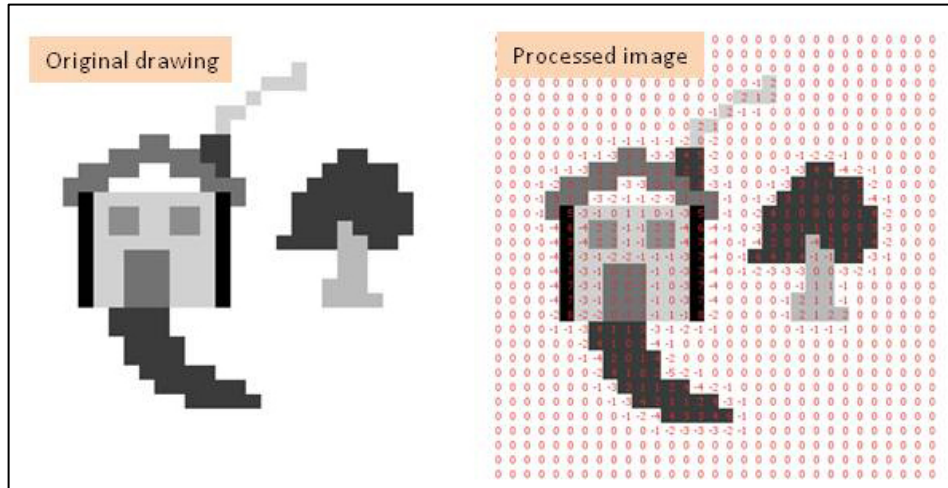


Figure 10. The edge detection simulation.

Once the image has been pre-processed, the user selects a threshold to be used to detect the edges. The notion of thresholding sensory input under different contexts can be broached using this example.

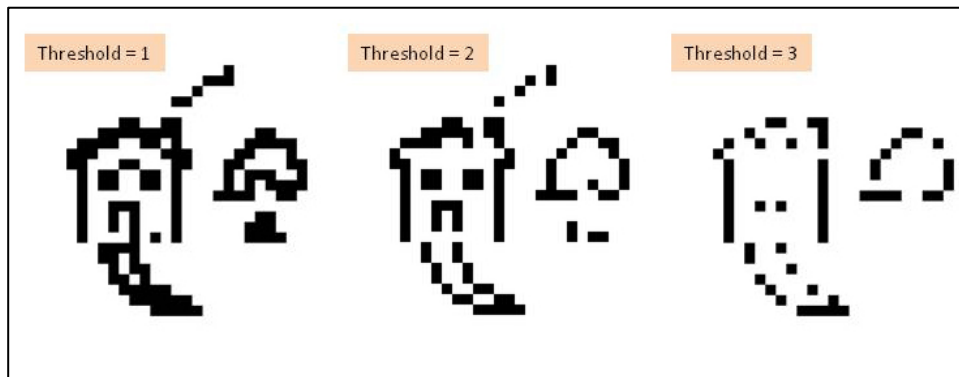


Figure 11. The image from figure 10 at different thresholds

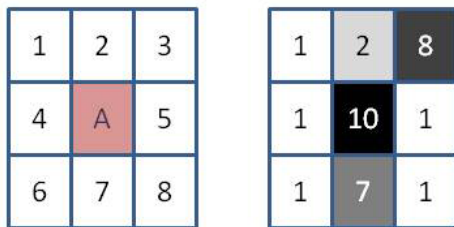
The simulation also allows the user to explore two well known contrast illusions: the Vassarely and the Mach effects.

Introduction to Neuroscience Assignment: Edge Detection

As we discussed in class, lateral inhibition is a process that helps ‘sheets’ (2D planes) of sensory receptors stress the position of edges in a sensory input. This ability is critical for the *segmentation* of the sensory image at locations where distinct objects are likely to be. In this exercise, you will use a simple algorithm to locate edges in a way that is analogous to lateral inhibition. Please make sure that all descriptions of your work compare what you found to what your own visual system does.

Exercises:

- 1- Open *V1-interactive.nlogo* and press **Setup**. This will turn the patches in the NetLogo world to white.
- 2- Using the **draw** button and the hue selector, draw a simple drawing in the world. Turn drawing on, select a hue (0 is darkest) and use the mouse to turn the ‘pixels’ in the window to the hue you selected. Draw something like a face or a house, but use different hues. This will represent the image in the real world that the visual system needs to process. More elaborate or contrasting images will provide the best results. Turning the erase switch on will allow you to erase instead of drawing.
- 3- Turn drawing off and press the **process-edges** button. This will carry out a calculation that asks each pixel to set its ‘level of activation’ to a value determined by its own color and by that of its neighboring pixels. As you can see in the left drawing below, each pixel has 8 neighboring pixels. Darker colors take higher number (Black = 10).



$$\begin{aligned}\text{Activation} &= \text{Color of A} - \text{average of neighbors} \\ &= 10 - 22/8 \\ &= 10 - 2.75 \approx 7\end{aligned}$$

If we color some pixels including pixel A, we can then calculate A’s activation using the formula shown below the drawings: A takes the original value 10 because its color is black. The average of the pixels around it is 2.75. If we round the answer to the nearest integer, we end up with an activation value of about 7.

If you press the button **show-labels**, the activation values for each pixel will be shown.

Look at the values. You will notice that darker pixels generally have higher activation values, especially when neighboring light colored pixels. This represents the idea that highly activated cells inhibit their neighbors very strongly.

- 4- Press the **edge** button. This procedure will cause any pixel with activations equal or higher

than the **threshold** (set by moving the slider) to turn black and those that are lower to turn white. This very simple calculation can detect edges in your original picture in most cases. Lower thresholds will cause more pixels to turn black. You can play around with the threshold to see this. Save an image of your drawing and another one of its edges and describe what you did in terms of what our visual system does to detect edges.

- 5- The “preset drawing” buttons will produce either a drawing of squares of decreasing sizes and darkening colors (square) or a simultaneous contrast illusion. Press **Setup** and then try them both. You will notice that both of them have interesting lateral inhibition phenomena. Make sure to remove the labels before you run them.

Vassarely: when you press this button, you will see that a large gray square appears, followed by a smaller darker square and so on. What do you see? You should notice an interesting ‘radiating’ pattern that is an illusion (Vassarely Illusion) produced by lateral inhibition in your own visual system. Press the **Vassarely** button again and look carefully at each square so that you are convinced that the diagonal rays are not there at all, they appear only when the smaller, darker squares are superimposed. Now, using the edge detection process (press **process-edges**) and moving the threshold, demonstrate why it happens and describe it.

Bands: when you press this button, you will see a series of vertical bars of increasing luminance appear from left to right. When they are all there, notice the very clear contrast observable between each pair of bars. It almost appears as though there is are two very thin lines separating them: a light line on the dark side and a dark line on the light side (Mach Bands Effect). If you press the Bands button again and look carefully as they bands appear, you will see that the edge touching the light blue background is not as clear as that between the bands themselves. Use the edge-detection process to see the interaction between pixels.

Vassarely-2: you may be wondering why it is that the Vassarely illusion resulted in a single radiating pattern when the difference between each contiguous square is exactly the same as that observed in the contiguous Mach bands. This example will show you why by overlapping a smaller number of squares two pixels apart.

Propagation of the Action Potential

This simulation has not yet been written into an assignment but could easily serve as one. I have only employed it in class as a demonstration of the effects of myelin on the speed of conduction.

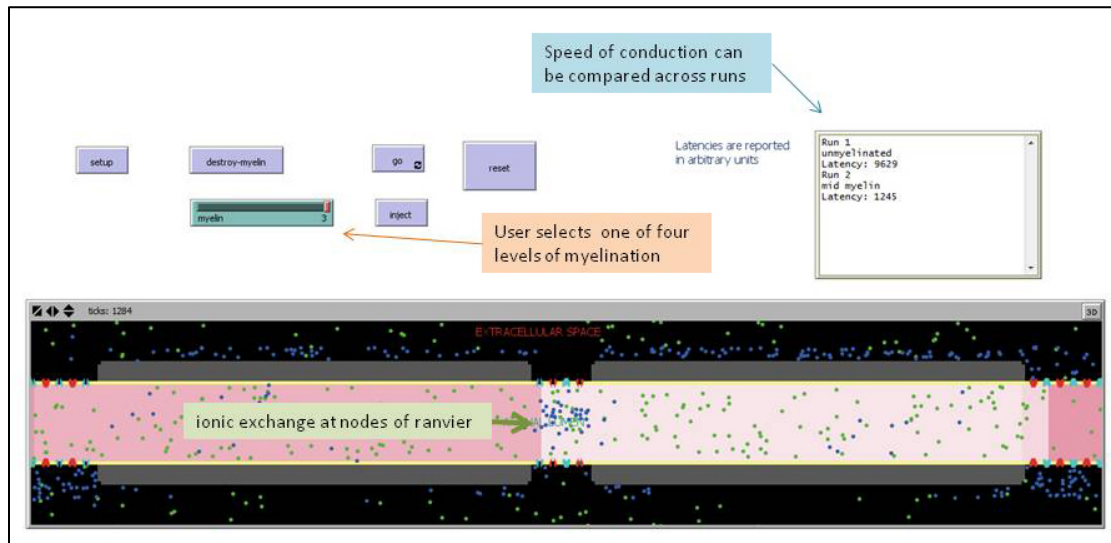


Figure 12. An example of a run in the propagation simulation.

Use:

- 1-Select a level of myelination (0-3): 0 means a non-myelinated fiber, 1-3 are increasing thicknesses of the myelin segments.
- 2- Press the **setup** button. A segment of an axon will be shown, with green and blue ions representing potassium and sodium ions, as in the potentials simulation. On the axonal membrane there will be cyan and red voltage-gated channels, representing potassium and sodium channels, also like in the potentials simulation.
- 3-Pressing the **go** button initiates the propagation of the impulse from left to right by opening the channels and creating ionic in- and efflux. The impulse charge is represented by lighter axonal lumen colors. In myelinated fibers, the exchange occurs only at the nodes of ranvier and the impulse skips from segment to segment.
- 4-In order to compare runs with different levels of myelinization, press the **reset** button after choosing the new myelin level. If you press setup, the simulation reverts to its original state, erasing the previous runs.

The Synapse

This simulation is similar to the antagonism simulation. However, in this case you can see both the pre- and postsynaptic membranes. When the **AP** (action potential) button is pressed, a 'wave' moves down the presynaptic region and opens calcium channels. This is followed by neurotransmitter release from vesicles and binding on the postsynaptic receptors.

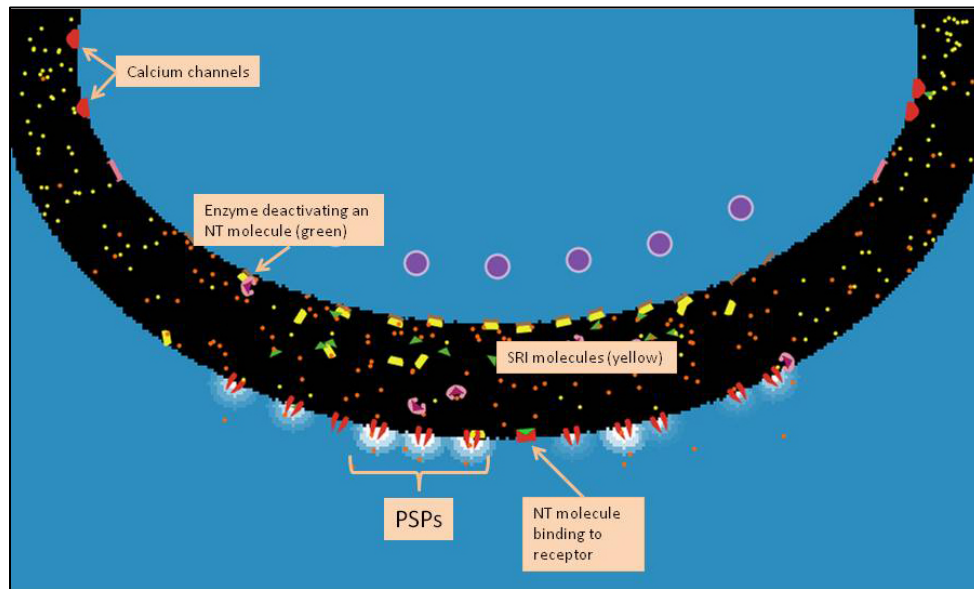


Figure 13. The 'world' window of the synapse simulation.

Use:

- 1-Pressing the setup button creates the pre- and postsynaptic structures. Press go to start the simulation and AP to produce an action potential wave down the presynaptic terminal.
- 2-As neurotransmitter (NT) molecules are released, they will travel and bind to receptors on the postsynaptic membrane. Excess NT molecules are reabsorbed through reuptake receptors on the presynaptic membrane.
- 3-The number of postsynaptic potentials (PSPs) produced is a function of the number of NT molecules released minus the number of NT molecules reabsorbed.
- 3-Adding selective reuptake inhibitors (SRIs) produces an user-defined number of yellow molecules that bind to the reuptake receptors, increasing the available molecules of NT for binding to the postsynaptic receptors.

Back-Propagation Network

Connectionist networks are useful as examples of the computational power of large groups of interconnected nodes. In class I use this simulation to show how a small network can learn to identify symbols through practice. Moreover, connectionist networks, similar to neural networks, are good at tasks such as pattern completion and exhibit interesting properties such as graceful degradation.

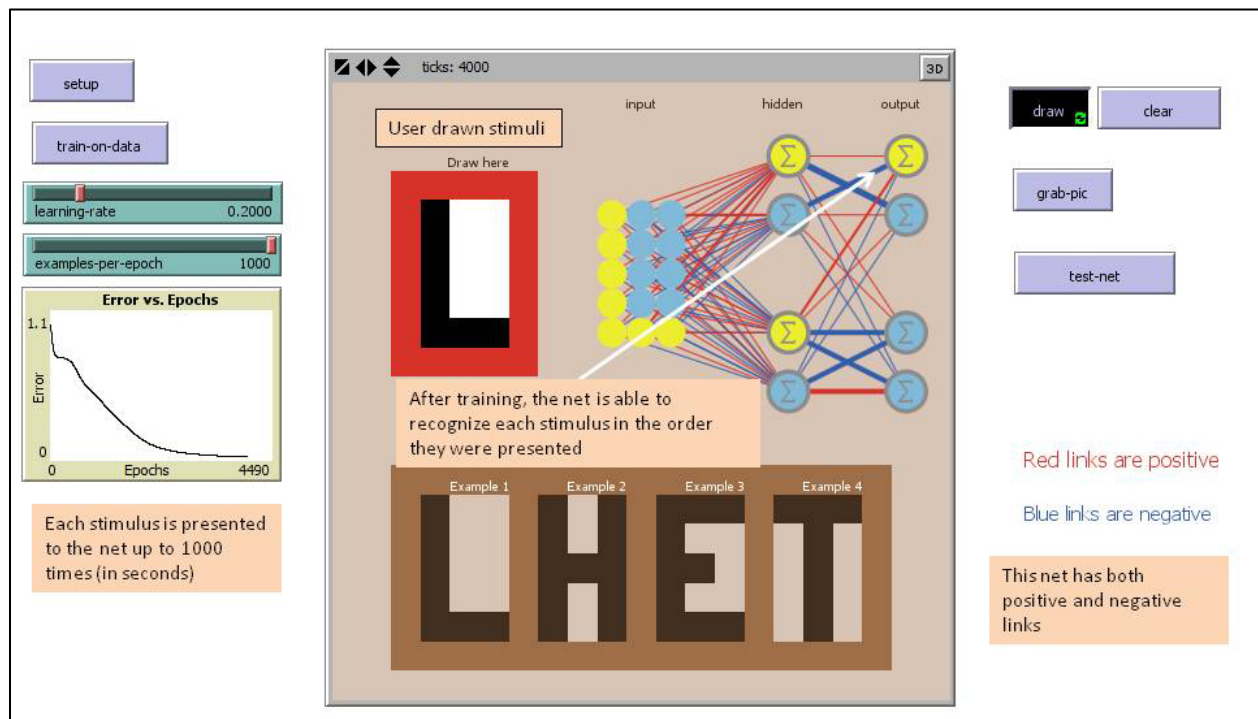


Figure 14. The interface of the back-propagation simulation

Use:

1-Pressing **setup** will create the net and the interface. Press **draw** and use the mouse to draw simple 3X5 pictures (they don't have to be actual symbols). After you have finished each picture, press **grab-pic** to have that picture appear in the box labeled 'Example 1' below. Draw up to four pictures.

2-Press train-on-data to have the four drawings presented to the net one after the other up to 1000 times each. The net uses back-propagation (hence the BP name) to propagate the error through the net and optimize the strength of the links between nodes. The thickness of the nodes represents their 'weight'.

3-After the net is done training, draw again in the white box. You don't have to draw exactly the same symbols, changing one or two pixels will usually not affect the results unless there are two symbols that are equally similar to the picture. Press test-net and the net should choose one fo the four output nodes. The nodes correspond to the examples 1 through 4 in a vertical fashion.

4-In class, I usually show the students that the heavier links, if eliminated, result in the worse disruption of the recognition of the symbols. To eliminate a link, right click the mouse over the link and hover it over the 'inspect link' options. When the correct link is selected click once more. Then

in the open field at the bottom of the open window write 'die' and press enter. That should kill the link. Use the **test-net** button to show the effects of eliminating the link.

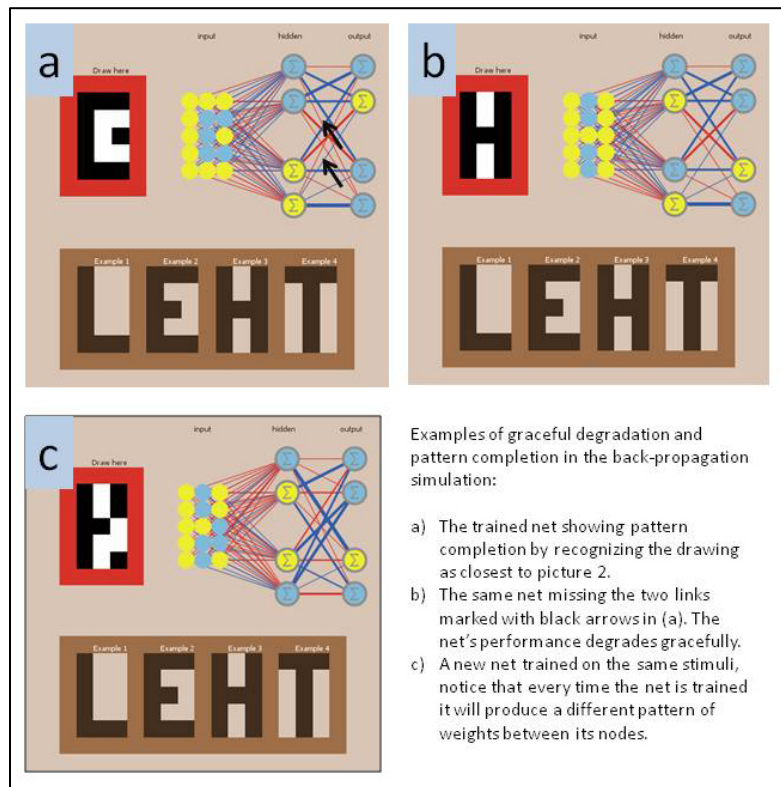


Figure 15. Examples of back propagation runs

In class I commonly show that eliminating heavy links between nodes causes the network to settle into 'confused' states. With the appropriate background discussion, the students can articulate how that would be similar to performance deficits in Alzheimer's or stroke patients.

In the figure at left, we observe examples of the properties of the back-propagation network.

These properties are observable in neural networks as well. For example, the onset of neurodegenerative diseases is typically difficult to detect due to the ability of brains to degrade their performance gracefully.

Another interesting property of the BP network is that every time it is trained, even on the same stimuli, it will produce different networks. This is a good example to use when explaining why every person's brain is different at the network level even if there are similarities at the system level.

Musical Circuit

The musical circuit simulation was created to give students a feeling for the complexity and capabilities of groups of interconnected cells. So far, it has been given as an optional assignment with the aim of getting students to try their hand at getting the simulation to play short songs such as Twinkle twinkle little star or Mary had a little lamb.

The behavior of each one of the cells is exactly the same as that used for the Temporal Integration simulation, allowing for the interesting use of inhibition and excitation for the creation of rhythms.

The simulation has two keyboards, a diatonic (shown in Figure X) and a chromatic. It can save circuits (as small comma separated text files) and load them subsequently. The user chooses the location of each cell in the circuit other than the sound cells (cells 1-12) and can use phasic or tonic firing of any cell to get the circuit going.

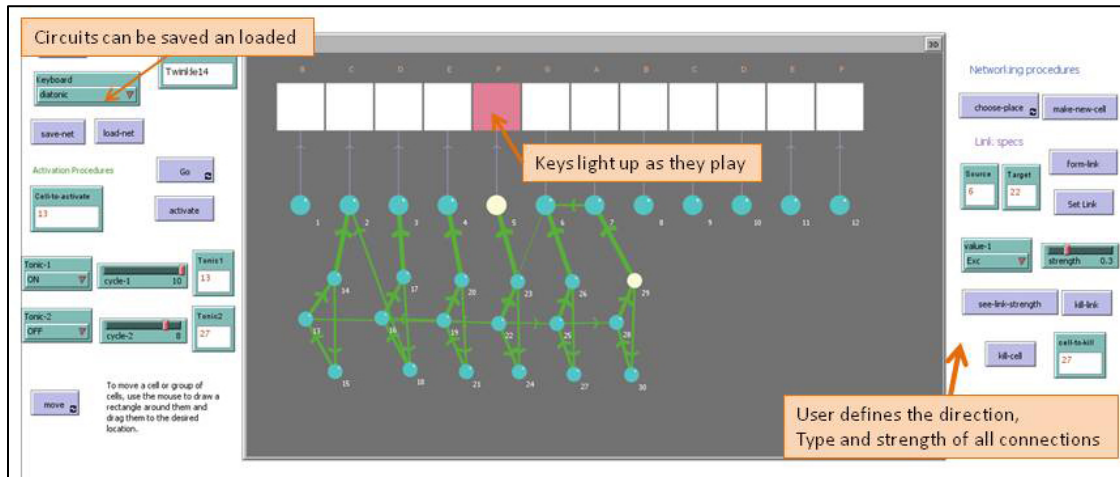


Figure 16. The Musical Circuit simulation

I have only assigned this circuit once so far. About seven students decided to give it a try. Some of them produced very complex circuits but all of them learned how difficult it is to create a circuit that does exactly what you want it to do. It was quite fun to play them in class to show everyone the work their classmates had done. In one case, a student created a circuit with over 50 new cells all with excitatory connections. Needless to say, the circuit was very unwieldy and, even though it produced a melody, it was nothing recognizable. When I asked the student how the circuit worked he answered: "I have no idea!", to the hilarity of the rest of the class. I took advantage of the chance to point out that it is precisely the level of circuit design that neuroscientists have found the most difficult to describe.

Musical Circuit Assignment

Here are some directions on how to use the Musical Circuit Simulation. Also refer to the 'Information' tab in the program itself.

Right now, the goal would be to produce one of the following two types of circuits:

- a circuit that can play an actual melody or
- smaller 'fundamental' circuits that can be used reliably to build more complex ones.

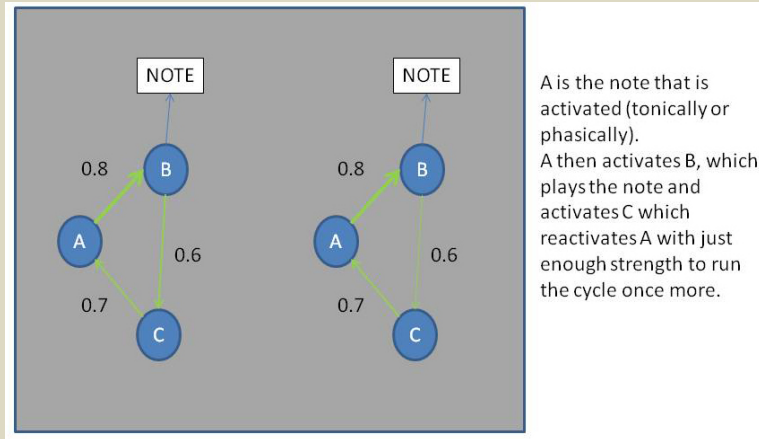


Figure 17. Example of a 'fundamental circuit'

Most of the other notions that you need to know you already have learned from the Integration assignment, so the learning curve shouldn't be too steep. One thing that may be a good idea is to not leave the **choose-place** button on during a run. Turn it off after you are done creating your cells.

There shouldn't be many actions that will cause an error. But there may be some. If you receive an error please copy it and send it to me so I can address it. As a rule of thumb, whenever you have worked on a circuit for 15 minutes or so, save it as a file to prevent frustration if the program causes trouble.

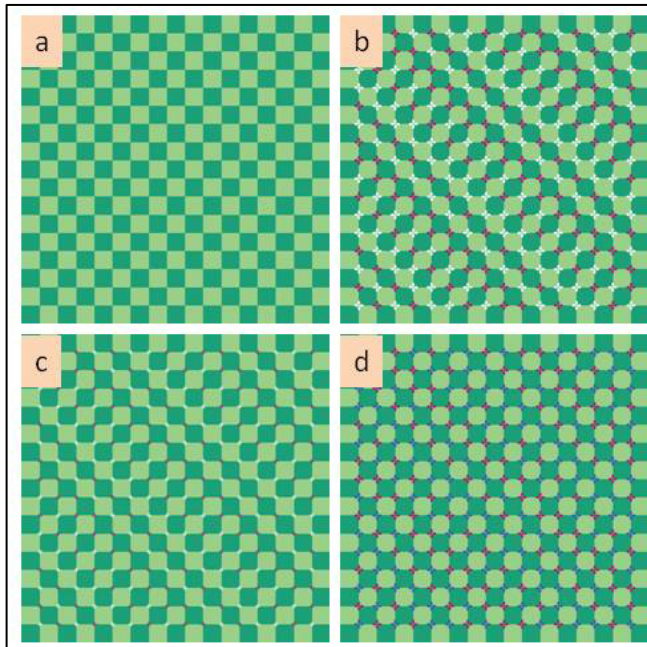
Thank you for doing this. Please give me your comments on the simulation. As mentioned in class, I'll give 5 extra points to whoever can create some interesting circuits (particularly fundamental ones) and 10 to whoever can play Twinkle Twinkle Little Star or similar ditty.

If you want to play a keyboard to check your melody, go to:

http://www.bgfl.org/bgfl/custom/resources_fftp/client_fftp/ks2/music/piano/

The first one is by far the more fun to try, but my feeling is that it won't be long before you wish someone had done the second one already. That is because as you start putting together cells and projections, the circuit will quickly become unpredictable (that is what I meant in class when I said that dynamic systems are very complex). Having a set of fundamental circuits would facilitate building a larger one that actually does something. Figure 17 is an example of a small circuit that plays two times a given note.

Kitaoka Sakura



This simulation shows the well known Sakura illusion by Akiyoshi Kitaoka. After pressing the **setup** button, a checkered background appears **(a)**. The go button creates a series of flower-shaped objects at the intersection of the squares. Importantly, the relative luminance of the objects results in an illusory distortion of the checkered pattern (b). The size of the objects can be reduced and still observe the illusion (c). The shape of the objects can also change without a change in the illusion (not shown). Finally, objects of similar luminance make the illusion disappear (d).

Figure 18. Examples of the capabilities of the Kitaoka Sakura simulation.

References

Schettino(2014) NeuroLab: a set of graphical computer simulations to support neuroscience instruction at the high school and undergraduate level. Journal for Undergraduate Neuroscience Education (JUNE)

Wilensky, U. (1999) GasLab: an extensible modeling toolkit for exploring micro and macro- views of gases. In Roberts, N., Feurzieg, W. & Hunter, B. (Eds.) Computer modeling and simulation in science education (pp. 151-178). Berlin: Springer Verlag.