A sensitizing robot

Johan Strumpfer

15 November 2008

1 Introduction

Synaptic plasticity is what provides neural networks with their power and hence us with our intelligence. This has been recognized and implemented in many ways that allow the use of learning rules such as Hebb’s Rule [3, 1, 6, 4]. To make this implementation closer to biological reality, we decided to look closer at how neurons adapt their firing rates.

To do this a closer look at the synapse is needed. Many plasticity mechanisms are used by a post-synaptic neuron to increase its sensitivity to the pre-synaptic neuron’s spikes. The net effect of most of these pathways is to increase the amount which the post-synaptic membrane depolarizes due to an incoming pre-synaptic spike. The simplest way to do this is to increase the number of ion channels that transport the ions that depolarize the membrane.

The effect of this is that for a sensitized synapse, the pre-synaptic neuron fires and the depolarization builds up quicker on the post-synaptic neuron. This happens because as the pre-synaptic neuron fires, it releases neurotransmitters which activate ion channels on the post-synaptic neuron. The more ion channels that are available, the more neurotransmitters can bind and hence increase the flux of cations into the post-synaptic neuron.

This can be modeled in a few ways. Typically this is done by using voltage dependent resistors in MOSFET’s to change the firing threshold of the post-synaptic neuron in VLSI’s [1, 6, 5, 2]. The net effect on the firing rate is the same as increasing the cation flux, but it is not biologically accurate. To change the voltage threshold of ion channels would require an alteration in the structure of the ion-channel which is difficult to do.

To model the increase in flux through the membrane we needed to be able to implement a change in the flux. To do this we needed to discretize the current flowing into the post-synaptic neuron. One way to do this is to use a micro-controller. The model is simple: the pre-synaptic neuron generates an action potential which arrives at the micro-controller.
The micro-controller then releases pulses of charge at a certain rate which depends on the timing between action potentials for a set amount of time. The released pulses of charge then collect on the membrane of the post-synaptic neuron.

2 Implementation

While the model outlined in the introduction could be built and tested under ideal conditions, we wanted to implement in realistic conditions. For this we decided to build a touch avoiding robot. The robot would be able to roll around and when it bumps into an object, it would turn to try and avoid it.

2.1 Robot Construction

Figure 1: A depiction of the robot body where the frame is a computer hard-drive mounting bracket chosen for its light weight and ease of customization. The large, rubberized rear wheels are attached to 12V motors strong enough to power the robot forward. The touch sensor sits on the front of the robot in the center above and more forward than the small front wheel which is for balance.
The robot was constructed as simply as possible. Figure 1 shows the basic parts of the robot. The body is a hard-drive mounting bracket for a tower computer cases. It was chosen for its light weight and ease of manipulation and customization. The appropriate holes were drilled to be able to mount the motors in the rear. To each motor a rubberized wheel is attached. The rear wheels drive the robot and they can be made to turn the robot by slowing down the wheel on the side that we wish to turn towards. On the underside of the front of the robot there is a smaller, unpowered third wheel mounted for balance. The breadboards that hold the controlling electronics are mounted on top of the hard-drive bracket. On the front of the robot is the touch sensor, which is attached via a stiff spring to the front side of the breadboards.

2.2 Neuron Connectivity

![Neuron Connectivity Diagram](Image)

Figure 2: The connectivity of the neurons is shown above.

To control the robot, only two neurons are used. The first neuron is connected to the touch sensor, and fires only when the touch sensor is activated. The second neuron is connected to one of the motors, and allows us to slow down one wheel when it fires, thus turning the robot. The synapse between the two neurons is excitatory and it’s strength is controlled by the micro-controller, which is described below.

2.3 Electronics

Each neuron is made up from the circuit shown in Figure 3. The way that the circuit works is as follows: Charge collects on the first capacitor $C_1$ with each arriving pulse. When this capacitor is above the voltage threshold $V_{th}$, the comparator puts out a pulse to the synapse (the microcontroller) and the inverter. The inverter inverts the pulse from the comparator which drains the capacitor and allows the capacitor to start collecting charges again.
2.4 Microcontroller

The micro-controller was used to allow the control of synaptic strength. This was done by changing the charge flux based on the time between pre-synaptic spikes. This gives the neural network a spike-timing dependent plasticity without modifying the firing thresholds.

The charge from the firing pre-synaptic neuron collects on a capacitor which drains slowly. This is a method by which the time between spikes can be measured by measuring the voltage on the capacitor. The micro-controller is programmed to measure this voltage with its analogue-to-digital converter (ADC) and put out a burst of +5V pulses with a frequency dependent on the voltage.

The micro-controller used is a PIC 18F452. The algorithm that was used is given below and the code in Appendix A.

3 Results

4 Conclusion

References


