**Neuron Simulator**

**Team Members:**

Sophia Wilhelmi, Ai Li, Renee Sajedian

**Project Idea:**

We will create a web-based interactive learning tool to teach students about neurons and neural networks. Students will begin by learning about the action potential and the biological basis of how neurons can send excitatory or inhibitory signals to each other. Then, we will introduce them to how neurons can be thought of as having weights and activation, and demonstrate this through a simple 3-neuron network. Next, we will go through some basic circuit types (converging, diverging, feedback, feedforward, etc). Then, we will introduce some examples of actual neural circuits. Finally, we will have tutorials where students create their own networks (in specific, guided ways) and adjust the weights and activations of neurons to achieve certain properties. Then, we will allow the students full freedom to play with small networks and see what they can create.

Through working with this interactive tool, we want students to understand neurons as a computational unit, and to gain some intuitions about neural networks and how the complex connections neurons can give rise to greater computational power. Neural networks can seem very abstract, and we want to remove some of this abstraction and give students a hands-on way to learn the underlying principles.

The students will be guided through how to work with this tool in a homework assignment format. The homework will have increasingly challenging steps, each of which will reveal more ways to utilize the neuron simulator. They will first have to construct a circuit to mock one pictured in the assignment, and list the results of 10 trials. Their last task will be to create their own circuit and list the important takeaways in the results of the trials. They will be encouraged to make complex circuits. They will be able to place neurons on the workspace, and edit their properties by clicking on the neuron, highlighting it, and having boxes appear in which they can edit the weight and threshold. They can also edit the incoming voltage. Lastly, they can either run the simulation or clear the screen and start over with an empty workspace. Running the simulation will highlight the firing neurons in bold green, and the neurons that didn’t fire in a grey-red.

**Code**

The code will be written in JavaScript and presented on a web page. The neural network will be represented as a directed graph of connected neurons, where each vertex in the graph is a Neuron object that has its weight and threshold. The connectivity of the network is stored both globally and locally. The final project will feature click-and-drag editing of elements and individual neuron update in pop-up windows. For more advanced implementations, we are considering the possibility of allowing input to become a function of time. Another advanced feature would be to include more types of neurons such as light sensors and mechanical sensors.