HW3

Learning in a neural network

Weights are initialized to small random values.

During training the network adjusts weights, increasing weights that contribute to the correct answer, and decreasing weights that contribute to the wrong answer.’

Error-driven learning

Hebbian Learning

Neurons that fire together wire together

Heaviside Function / Binary Step / if x >= 0, return 1 if x < 0 return 0

If the sum of all the weighted inputs + the bias > 0, then this neuron has activation 1

Taste receptors on the tongue activate based on the presence of chemicals

Ways to set the weights and biases to represent logical connectives

And – w1 = 1; w2 =1, b = -1.5

OR – w1 = 1; w2 =1, b = -0.5

XOR – why is there no solution?

Reflection:

Thinking mathematically about neural networks, and visualizing activation functions.

A neural network is a program modeled after the function and structure of a brain. The smallest anatomical units in the brain are called neurons, and every neuron connects to other neurons to form a network. To communicate information, neurons send electrical impulses to their axon terminals (“firing”), and when several neurons adjacent to a single neuron fire, that single neuron tends to fire as well. In fact, this idea of how neurons interact with each other is the basis for Hebbian learning, in which when neurons fire together, the functional connectivity between them increases.

Activation functions take a net input of activity going into a neuron (sum of weighted activations of all the neurons connecting to it), and return the activation of that affected neuron. The idea behind this is that if we let the activation of one neuron become completely determined by the activations of neurons it connect to, we have modeled a large part of how real neurons feedforward to a single neurons.