Problem Set 1, Part 1

Perceptrons

Note: if you get stuck, you can find sample code on the course website at: http://druckmann-lab.github.io/nepr208/ps1.html

1. Implement Perceptron learning for binary patterns. An input pattern ξ^μ generates an output y^μ that we would like by learning to be equal to σ^μ. Recall that the perceptron learning rule is:

$$\Delta w_j = \begin{cases} 0 & \text{if } \sigma^\mu = y^\mu \\ 2\eta\xi_j^\mu\sigma^\mu & \text{if } \sigma^\mu \neq y^\mu \end{cases} \tag{1}$$

   (a) Implement inputs and outputs for the Boolean AND function. Show that it can be learned.

   (b) Implement inputs and outputs for the Boolean XOR function. Show that it can *not* be learned.

   (c) **Bonus:** implement perceptron learning for multi-layer perceptrons to show that the XOR function can be classified once a hidden layer is added.

2. How many patterns can a perceptron learn? Implement a 10 input neuron Perceptron. Generate datasets with different number of randomly generated binary patterns. Train the Perceptron. See how large a dataset can still successfully train.