Computational Neuroscience Homework 8

Zihan Zhang (Steven)

December 22, 2022

1 Section 1

In this section, we consider a very simple network with 3 neurons and denote the activity (firing rate) of each neuron as x_1, x_2, x_3, connected by linear synapses with weights w_1 and w_2. Thus, the firing rate of x_3 is:

$$x_3 = w_2 x_2 = w_2 w_1 x_1. \tag{1}$$

We will construct a learning rule that modifies w_1 and w_2 so that the firing rate of x_3 matches a target firing rate y. Also, we define a quadratic loss function:

$$l(y, x_3) = \frac{1}{2}(y - x_3)^2. \tag{2}$$

When $x_1 > 0, w_1 < 0$, and $x_3 < y$, to improve the loss, we need to increase w_2 and w_1 if w_2 is positive and decrease them if w_2 is negative. To implement the gradient descent, by chain rule, we have:

$$\begin{aligned}
\frac{dl}{dx_3} &= x_3 - y \\
\frac{dl}{dw_2} &= \frac{dl}{dx_3} \cdot \frac{dx_3}{dw_2} = (x_3 - y)x_2 \\
\frac{dl}{dx_2} &= \frac{dl}{dx_3} \cdot \frac{dx_3}{dx_2} = (x_3 - y)w_2 \\
\frac{dl}{dw_1} &= \frac{dl}{dx_3} \cdot \frac{dx_3}{dx_2} \cdot \frac{dx_2}{dw_1} = (x_3 - y)w_2 x_1.
\end{aligned} \tag{3}$$

Thus, we could generate an update rule that is guaranteed to decrease the loss function for a sufficiently small step size (learning rate α):

$$\begin{aligned}
w_2 &= w_2 - \alpha \cdot \frac{\partial l}{\partial w_2} = w_2 - \alpha(x_3 - y)x_2 \\
w_1 &= w_1 - \alpha \cdot \frac{\partial l}{\partial w_1} = w_1 - \alpha(x_3 - y)w_2 x_1.
\end{aligned} \tag{4}$$

2 Section 2

In this part, we adapt more to the real biological nonlinear networks. Consider σ as the nonlinear function and the model:

$$\begin{aligned}
x_3 &= w_2 \sigma(w_1 x_1) \\
x_2 &= w_1 x_1.
\end{aligned} \tag{5}$$

Reproduce the previous calculation, we have:

$$\begin{aligned}
\frac{dl}{dx_3} &= x_3 - y \\
\frac{dl}{dw_2} &= \frac{dl}{dx_3} \cdot \frac{dx_3}{dw_2} = (x_3 - y)\sigma(x_2) \\
\frac{dl}{dx_2} &= \frac{dl}{dx_3} \cdot \frac{dx_3}{dx_2} = (x_3 - y)w_2 \sigma'(x_2) \\
\frac{dl}{dw_1} &= \frac{dl}{dx_3} \cdot \frac{dx_3}{dx_2} \cdot \frac{dx_2}{dw_1} = (x_3 - y)w_2 \sigma'(x_2) x_1.
\end{aligned} \tag{6}$$

Comparing Equation (3) and (6), there are more nonlinearity terms involved in the calculation, including $\sigma(\cdot)$ and $\sigma'(\cdot)$. Also, if we use the rectified linear unit (ReLU) function, $\sigma(x) = \max(0, x)$, into the calculation, we could have $\sigma'(x) = 1$ if $x > 0$ and $\sigma'(x) = 0$ if $x \leq 0$. We plug this information into Equation (6) to compute the gradient of the weights for $\frac{dl}{dw_2}$ and $\frac{dl}{dw_1}$.

However, if x_1 and w_1 have opposite sign, meaning x_2 would always be negative. Retrospecting to Equation (6), the learning could not be conducted regardless of the learning rate since $\sigma'(x_2)$ would always be 0, which leads to a problem.