Problem 6.1

Suppose you have a recurrent neural network with input $x[n]$, target $y[n]$, output $\hat{y}[n]$, and error metric

$$E = -\frac{1}{N} \sum_{n=0}^{N-1} (y[n] \ln \hat{y}[n] + (1 - y[n]) \ln(1 - \hat{y}[n]))$$

where

$$\hat{y}[n] = \sigma(e[n]),$$
$$e[n] = x[n] + \sum_{m=1}^{M-1} w[m] \hat{y}[n-m],$$

and where $\sigma(\cdot)$ is the logistic sigmoid. Write $dE/dw[3]$ in terms of the signals $y[n]$ and $\hat{y}[m]$. You can invent auxiliary signals such as $\dot{\sigma}[n]$, $\epsilon[n]$, or $\delta[n]$ if you wish, but you need to define them clearly. You may assume that $\hat{y}[n] = 0$ for $n < 0$.

Solution: First step: forward-prop. Assume that $\hat{y}[n]$ has been calculated. Second step: partial derivatives:

$$\epsilon[n] = \frac{\partial E}{\partial e[n]} = \frac{1}{N} (\hat{y}[n] - y[n])$$

Third step: total derivatives:

$$\delta[n] = \frac{dE}{de[n]}$$
$$= \epsilon[n] + \sum_{m=1}^{M-1} \frac{dE}{de[n+m]} \frac{\partial e[n+m]}{\partial e[n]}$$
$$= \epsilon[n] + \sum_{m=1}^{M-1} \delta[n+m] w[m] \hat{y}[n] (1 - \hat{y}[n])$$

Fourth step: weight gradient:

$$\frac{dE}{dw[3]} = \sum_{n=0}^{N-1} \frac{dE}{de[n]} \frac{\partial e[n]}{\partial w[3]}$$
$$= \sum_{n=0}^{N-1} \delta[n] \hat{y}[n-3]$$
Problem 6.2

Suppose that

\[ h_0 = x^3 \]
\[ h_1 = \cos(x) + \sin(h_0) \]
\[ \dot{y} = \frac{1}{2} (h_1^2 + h_0^2) \]

What is \( \frac{d\dot{y}}{dx} \)? Express your answer as a function of \( x \) only, without the variables \( h_0 \) or \( h_1 \) in your answer.

Solution:

\[
\begin{align*}
\frac{dh_0}{dx} &= 3x^2 \\
\frac{dh_1}{dx} &= \frac{\partial h_1}{\partial x} + \frac{dh_0}{dx} \frac{\partial h_1}{\partial h_0} \\
&= -\sin(x) + 3x^2 \cos(x^3) \\
\frac{d\dot{y}}{dx} &= \frac{dh_0}{dx} \frac{\partial \dot{y}}{\partial h_0} + \frac{dh_1}{dx} \frac{\partial \dot{y}}{\partial h_1} \\
&= h_0 (3x^2) + h_1 (-\sin(x) + 3x^2 \cos(x^3)) \\
&= 3x^5 + (\cos(x) + \sin(x^3)) (-\sin(x) + 3x^2 \cos(x^3))
\end{align*}
\]

Problem 6.3

Consider a one-gate recurrent neural net, defined as follows:

\[
\begin{align*}
c[n] &= c[n-1] + w_c x[n] + u_c h[n-1] + b_c \\
h[n] &= o[n] c[n] \\
o[n] &= \sigma(w_o x[n] + u_o h[n-1] + b_o)
\end{align*}
\]

where \( \sigma(\cdot) \) is the logistic sigmoid, \( x[n] \) is the network input, \( c[n] \) is the cell, \( o[n] \) is the output gate, and \( h[n] \) is the output. Suppose that you’ve already completed synchronous back-prop, which has given you the following quantity:

\[
\epsilon[n] = \frac{\partial E}{\partial h[n]}
\]

Find asynchronous back-prop formulas, i.e., find formulas for the following quantities, in terms of one another, and/or in terms of the other quantities defined above:

\[
\begin{align*}
\delta_h[n] &= \frac{dE}{dh[n]} \\
\delta_o[n] &= \frac{dE}{do[n]} \\
\delta_c[n] &= \frac{dE}{dc[n]} \\
\end{align*}
\]
**Solution:**

\[ \delta_h[n] = \epsilon[n] + \delta_c[n + 1]u_c + \delta_o[n + 1]\hat{o}(w_o x[n + 1] + u_o h[n] + b_1)u_o \]
\[ = \epsilon[n] + \delta_c[n + 1]u_c + \delta_o[n + 1]o[n + 1](1 - o[n + 1])u_o \]
\[ \delta_o[n] = \delta_h[n]c[n] \]
\[ \delta_c[n] = \delta_c[n + 1] + \delta_h[n + 1]u_c \]

**Problem 6.4**

Using the CReLU nonlinearity for both \( \sigma_h \) and \( \sigma_g \) in an LSTM, choose weights and biases,
\[ \{b_c, u_c, w_c, b_f, u_f, w_f, b_i, u_i, w_i, b_o, u_o, w_o\} \],
that will cause an LSTM to count the number of nonzero inputs, and output the tally only when the input is zero:
\[ h[n] = \begin{cases} \sum_{m=0}^n 1[x[m] \geq 1] & x[n] = 0 \\ 0 & x[n] \geq 1 \end{cases} \]
where \( 1[\cdot] \) is the unit indicator function, and you may assume that \( x[n] \) is always an integer.

**Solution:** First, we only want to generate output when \( x[n] \geq 1 \), so
\[ o[n] = \text{CReLU}(x[n]), \quad \Rightarrow \quad b_o = 0, w_o = 1, u_o = 0 \]
Second, we want \( c[n] \) to increase by exactly 1, every time \( x[n] \geq 1 \), i.e.,
\[ c[n] = c[n - 1] + \text{CReLU}(x[n]) \]
but we know that \( c[n] \) is defined to be
\[ c[n] = i[n]\text{CReLU}(w_c x[n] + u_c h[n - 1] + b_c) + f[n]c[n - 1] \]
so we want \( i[n] = 1 \) always, \( f[n] = 1 \) always, and
\[ w_c = 1, u_c = 0, b_c = 0 \]
Putting it all together, we have
\[ \{b_c = 0, u_c = 0, w_c = 1, b_f = 1, u_f = 0, w_f = 0, b_i = 1, u_i = 0, w_i = 0, b_o = 0, u_o = 0, w_o = 1\} \]