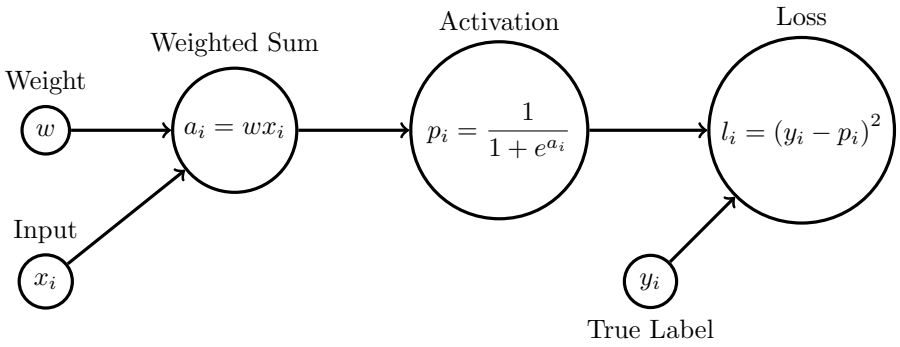
Assignment 4, Written Part

1. *Consider a linear model for classification in which we use a logistic activation, but instead of cross-entropy loss, we use squared error loss. Assume a 1-dimensional input x, a single weight w and an outcome $y_i \in \{0, 1\}$. We will ignore the intercept term.*

$$
\begin{aligned}
a_i &= wx_i \\
p_i &= \text{logistic}\left(a_i\right) = \frac{1}{1+e^{-a_i}} = \frac{1}{1+e^{-wx_i}} \\
l_i &= \left(y_i - p_i\right)^2
\end{aligned}
$$

Recall that $\text{logistic}\left(u\right) = \frac{1}{1+e^{-u}}$.



Calculate the following:

a. $\frac{dl_i}{dp_i} = 2\left(y_i - p_i\right)(-1) = \boxed{2p_i - 2y_i}$

b. $\frac{dp_i}{da_i}$, as a function of a_i, $\frac{dp_i}{da_i} = \frac{1}{(1+e^{-a_i})^2}\left(0 + e^{a_i}\right) = \frac{e^{-a_i}}{(1+e^{-a_i})^2}$

c. $\frac{dp_i}{da_i}$, rewritten as a function of p_i only: $\boxed{\frac{d}{da_i}\left(p_i\right)}$

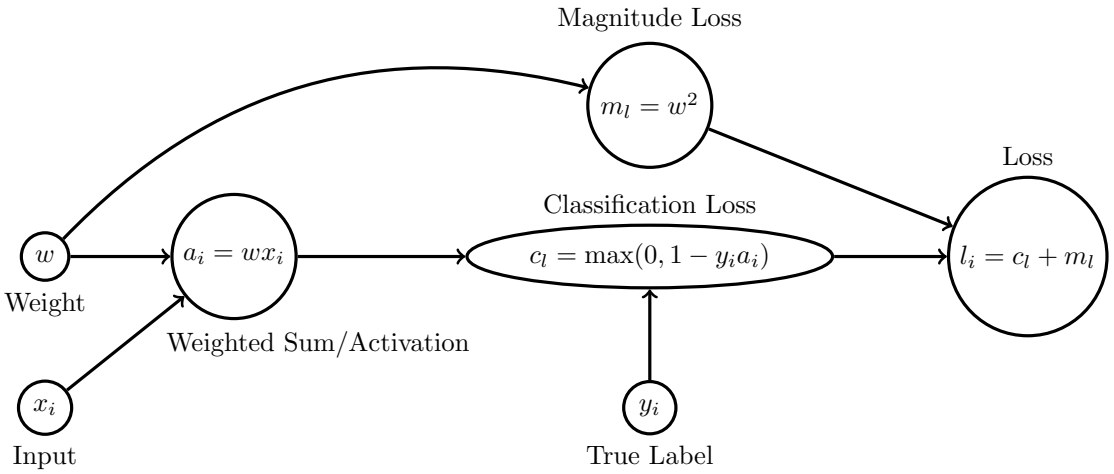
d. $\frac{da_i}{dw} = \boxed{x_i}$

e. $\frac{dl_i}{dw} = \frac{dl_i}{dp_i} \cdot \frac{dp_i}{da_i} \cdot \frac{da_i}{dw} = (2p_i - 2y_i)\left(\frac{e^{-a_i}}{(1+e^{-a_i})^2}\right)(x_i) = (2p_i - 2y_i)\left(e^{-a_i}p_i^2\right)(x_i) = \boxed{2p_i^3 e^{-a_i}x_i - 3p_i^2 e^{-a_i}y_i x_i}$

f. *Assume that $y_i = 1$. What is $\lim_{p\to 0} \frac{dl_i}{dw}$? Is this good or bad for learning? Explain why.*
$\lim_{p\to 0}\left[\frac{dl_i}{dw} = 2(0)^3 e^{-a_i}x_i - 3(0)^2 e^{-a_i}(1)x_i\right] = \boxed{0}$
This is bad for learning because in this case ($p_i = 0$) the sample has been mis-classified, but no weight adjustment occurs during backpropagation: $\frac{dl_i}{dw} = 0$. The above network cannot learn to classify this sample as 1. Likewise, the less activation there is, the less weight adjustment there is per training sample. This defeats the purpose of backpropagation.

2. *Consider a linear model for classification based on the hinge loss, with a penalty for weight magnitude. This is the basic support vector machine (don't worry if you haven't studied it). Unlike question 1, we will now assume that $y_i \in \{-1, 1\}$. Again, assume a single input variable x_i, and ignore the intercept term.*

$$a_i = wx_i$$
$$l_i = \max(0, 1 - y_i a_i) + w^2$$



Calculate the following:

a. $\dfrac{\partial l_i}{\partial a_i}$ [Note: This technically should be a subgradient. Only worry about the two cases of $y_i a_i < 1$ and $y_i a_i > 1$. Don't worry about the non-differentiable point where $y_i a_i = 1$.]

$$\frac{\partial l_i}{\partial a_i} = \begin{cases} \dfrac{\partial l_i}{\partial c_l} \cdot \dfrac{\partial c_l}{\partial a_i} = (1)(-y_i) = \boxed{-y_i} & \text{, if } y_i a_i < 1 \\ \dfrac{\partial l_i}{\partial c_l} \cdot \dfrac{\partial c_l}{\partial a_i} = (1)(0) = \boxed{0} & \text{, if } y_i a_i > 1 \end{cases}$$

b. $\dfrac{dl_i}{dw}$ [*Again, there are two cases.*]

$$\frac{dl_i}{dw} = \begin{cases} \dfrac{\partial l_i}{\partial c_l} \cdot \dfrac{\partial c_l}{\partial a_i} \cdot \dfrac{\partial a_i}{\partial w} + \dfrac{\partial l_i}{\partial m_l} \cdot \dfrac{\partial m_l}{\partial w} = (1)(-y_i)(x_i) + (1)(2w) = \boxed{-y_i x_i + 2w} & \text{, if } y_i a_i < 1 \\ \dfrac{\partial l_i}{\partial c_l} \cdot \dfrac{\partial c_l}{\partial a_i} \cdot \dfrac{\partial a_i}{\partial w} + \dfrac{\partial l_i}{\partial m_l} \cdot \dfrac{\partial m_l}{\partial w} = (1)(0)(x_i) + (1)(2w) = \boxed{2w} & \text{, if } y_i a_i > 1 \end{cases}$$

c. *Assume that $y_i = 1$. What is update rule for w for stochastic gradient descent?*

$$w \leftarrow w - \eta \nabla Q(w)$$
$$w \leftarrow w - \eta \left(\frac{d}{dw} \left(\max(0, 1 - y_i a_i) + w^2 \right) \right)$$
$$w \leftarrow \begin{cases} w + \eta y_i x_i - \eta 2w & \text{, if } y_i w x_i < 1 \\ w - \eta 2w & \text{, if } y_i w x_i > 1 \end{cases}$$

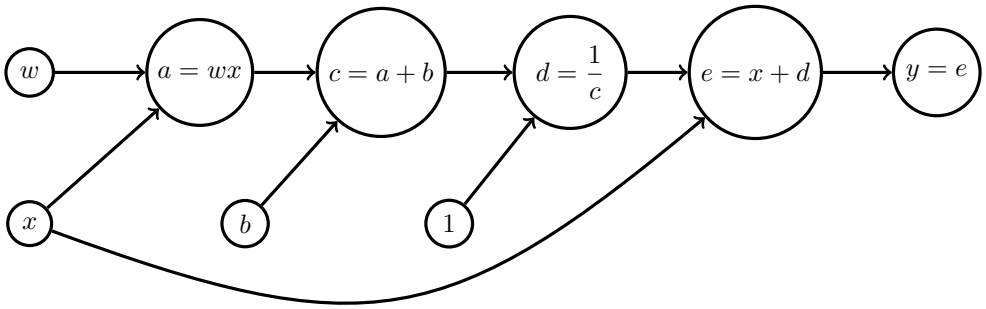
d. *Contrast this rule with the update rule for the perceptron.*
Perceptron Update Rule: $w \leftarrow w + \eta y_i x_i$
When y_i and wx_i have different signs (prediction error) or $|wx_i| < 1$ (narrow margin), then the derived update is just the perceptron update with an additional weight magnitude penalty. If the prediction is correct and the margin is greater than 1, then the weight magnitude is reduced, without changing the hyperplane normal vector direction. In the latter case, the perceptron update would have still nudged the hyperplane normal vector direction towards the training example.

3.

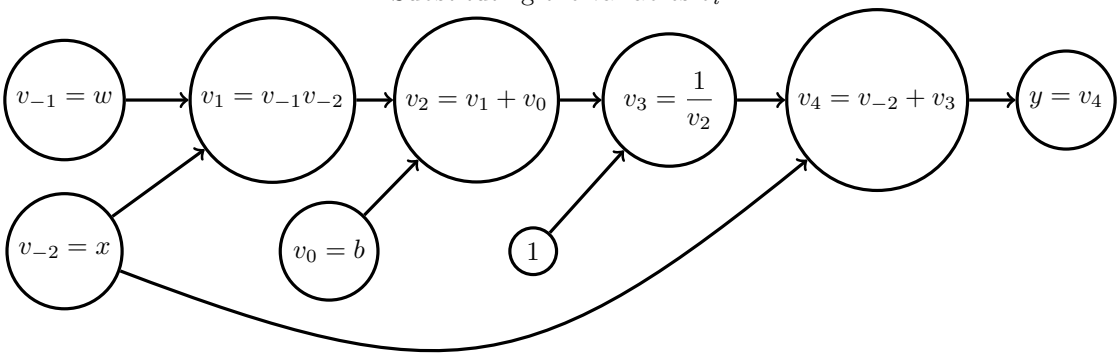
$$y = x + \frac{1}{wx + b}$$

Draw the computation graph for calculating y from x, w and b, Fill in the blanks for the reverse mode AD table at $x = 0.3, w = 0.5, b = 0.1$

Part 1 - Computation Graph



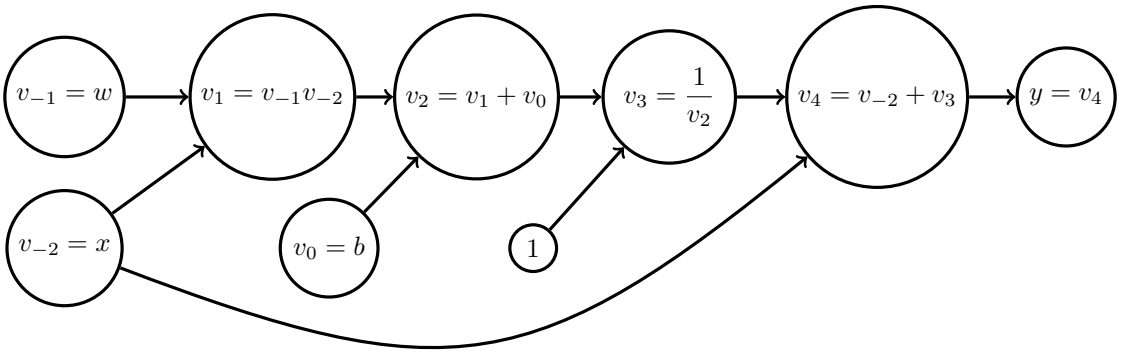
Substituting the variables v_i:



Forward Primal Trace

v_{-2}	$= x$	$= 0.3$
v_{-1}	$= w$	$= 0.5$
v_0	$= b$	$= 0.1$
v_1	$= v_{-2}v_{-1}$	$= 0.15$
v_2	$= v_1 + v_0$	$= 0.25$
v_3	$= \frac{1}{v_2}$	$= 4$
v_4	$= v_{-2} + v_3$	$= 4.3$
y	$= v_4$	$= 4.3$

Part 2 - Reverse Adjoint Trace



The adjoint of the variable v_i is $\bar{v}_i = \dfrac{\partial y}{\partial v_i}$, the sensitivity of the output with respect to the intermediate variable v_i. A link between the variables on the graph represents a dependency between intermediate variables. When differentiating composite functions of intermediate variables, the Chain Rule is used. Working backwards from $\bar{v}_4$ to $\bar{v}_{-2}$:

$$
\begin{array}{rlll}
\bar{v}_{-2} = & \dfrac{\partial y}{\partial v_4}\cdot\dfrac{\partial v_4}{\partial v_3}\cdot\dfrac{\partial v_3}{\partial v_2}\cdot\dfrac{\partial v_2}{\partial v_1}\cdot\dfrac{\partial v_1}{\partial v_{-2}} + \dfrac{\partial y}{\partial v_4}\cdot\dfrac{\partial v_4}{\partial v_{-2}} & = & (1)(1)\left(-\dfrac{1}{v_2^2}\right)(1)(v_{-2}) + (1)(1) \;=\; 1 - \dfrac{v_{-1}}{v_2^2} = 1 - \dfrac{0.5}{(0.25)^2} = -7 \\
\bar{v}_{-1} = & \dfrac{\partial y}{\partial v_4}\cdot\dfrac{\partial v_4}{\partial v_3}\cdot\dfrac{\partial v_3}{\partial v_2}\cdot\dfrac{\partial v_2}{\partial v_1}\cdot\dfrac{\partial v_1}{\partial v_{-1}} & = & (1)(1)\left(-\dfrac{1}{v_2^2}\right)(1)(v_{-2}) \;=\; -\dfrac{v_{-2}}{v_2^2} = -\dfrac{0.3}{(0.25)^2} = -4.8 \\
\bar{v}_0 = & \dfrac{\partial y}{\partial v_4}\cdot\dfrac{\partial v_4}{\partial v_3}\cdot\dfrac{\partial v_3}{\partial v_2}\cdot\dfrac{\partial v_2}{\partial v_0} & = & (1)(1)\left(-\dfrac{1}{v_2^2}\right)(1) \;=\; -\dfrac{1}{v_2^2} = -16 \\
\hline
\bar{v}_1 = & \dfrac{\partial y}{\partial v_4}\cdot\dfrac{\partial v_4}{\partial v_3}\cdot\dfrac{\partial v_3}{\partial v_2}\cdot\dfrac{\partial v_2}{\partial v_1} & = & (1)(1)\left(-\dfrac{1}{v_2^2}\right)(1) \;=\; -\dfrac{1}{v_2^2} = -16 \\
\bar{v}_2 = & \dfrac{\partial y}{\partial v_4}\cdot\dfrac{\partial v_4}{\partial v_3}\cdot\dfrac{\partial v_3}{\partial v_2} & = & (1)(1)\left[\dfrac{\partial}{\partial v_2}\dfrac{1}{v_2}\right] \;=\; -\dfrac{1}{v_2^2} = -\dfrac{1}{(0.25)^2} = -16 \\
\bar{v}_3 = & \dfrac{\partial y}{\partial v_4}\cdot\dfrac{\partial v_4}{\partial v_3} & = & (1)(1) \;=\; 1 \\
\hline
\bar{v}_4 = & \dfrac{\partial y}{\partial v_4} & = & \dfrac{\partial v_4}{\partial v_4} \;=\; 1
\end{array}
$$