1.

$x_0 = 1.0$

$w_0 = 0.3$

$w_1 = -0.2$

$b_0 = 0.1$

$b_1 = -0.3$

$xw_0 = x_0 \cdot w_0 = 0.3$ $\quad dxw_0 = w_0 dx_0, dxw_0 = x_0 dw_0$

$xwb_0 = xw_0 + b_0 = 0.4$ $\quad dxwb_0 = dxw_0, dxwb_0 = db_0$

$x_1 = ReLU(xwb_0) = 0.4$ $\quad dx_1 = 1 \cdot (xwb_0 > 0?)\, dxwb_0$

$xw_1 = x_1 \cdot w_1 = -0.08$ $\quad dxw_1 = w_1 dx_1, dxw_1 = x_1 dw_1$

$y_1 = xw_1 + b_1 = -0.38$ $\quad dy_1 = dxw_1, dy_1 = db_1$

$yx = y_1 + x_0 = 0.62$ $\quad dyx = dy_1, dyx = dx_0$

$z = ReLU(yx) = 0.62$ $\quad dz = 1 \cdot (yx > 0?)\, dyx$

$$\frac{dz}{dw_0} = \frac{dz}{dyx} \cdot \frac{dyx}{dy_1} \cdot \frac{dy_1}{dxw_1} \cdot \frac{dxw_1}{dx_1} \cdot \frac{dx_1}{dxwb_0} \cdot \frac{dxwb_0}{dxw_0} \cdot \frac{dxw_0}{dw_0}$$

$$= 1 \cdot 1 \cdot 1 \cdot (-0.2) \cdot 1 \cdot 1 \cdot 1$$

$$= -0.2$$

$$\frac{dz}{dw_1} = \frac{dz}{dyx} \cdot \frac{dyx}{dy_1} \cdot \frac{dy_1}{dxw_1} \cdot \frac{dxw_1}{dw_1}$$

$$= 1 \cdot 1 \cdot 1 \cdot (0.4)$$

$$= 0.4$$

$$\frac{dz}{db_0} = \frac{dz}{dyx} \cdot \frac{dyx}{dy_1} \cdot \frac{dy_1}{dxw_1} \cdot \frac{dxw_1}{dx_1} \cdot \frac{dx_1}{dxwb_0} \cdot \frac{dxwb_0}{db_0}$$

$$= 1 \cdot 1 \cdot 1 \cdot (-0.2) \cdot 1 \cdot 1$$

$$= -0.2$$

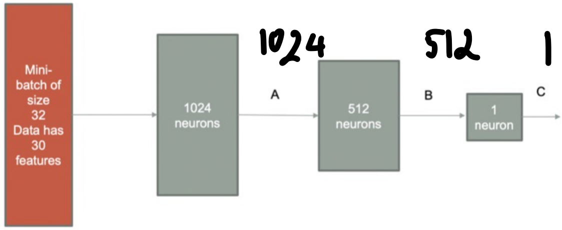
$$\frac{dz}{db_1} = \frac{dz}{dyx} \cdot \frac{dyx}{dy_1} \cdot \frac{dy_1}{db_1}$$

$$= 1 \cdot 1 \cdot 1$$

$$= 1$$

2.

T2. Given the following network architecture specifications, determine the size of the output A, B, and C.



T3. What is the total number of learnable parameters in this network? (Don't forget the bias term)

3. $30 \times 1024 + 1024 + 1024 \times 512 + 512 + 512 \times 1 + 1$

4.

Recall in class we define the softmax layer as:

$$P(y=j) = \frac{exp(h_j)}{\Sigma_k exp(h_k)} \tag{1}$$

where h_j is the output of the previous layer for class index j
The cross entropy loss is defined as:

$$L = -\Sigma_j y_j log P(y=j) \tag{2}$$

where y_j is 1 if y is class j, and 0 otherwise.

T4. Prove that the derivative of the loss with respect to h_i is $P(y=i) - y_i$. In other words, find $\frac{\partial L}{\partial h_i}$ for $i \in \{0, ..., N-1\}$ where N is the number of classes. Hint: first find $\frac{\partial P(y=j)}{\partial h_i}$ for the case where $j = i$, and the case where $j \neq i$. Then, use the results with chain rule to find the derivative of the loss.

Next, we will code a simple neural network using numpy. Use the starter code `hw4.zip` on github. There are 8 tasks you need to complete in the starter code.

Hints: In order to do this part of the assignment, you will need to find

$$\frac{\partial L}{\partial h_i} = \frac{\partial}{\partial h_i}\left(-\Sigma_j y_j \log P(y=j)\right)$$

$$= -\Sigma_j \frac{\partial}{h_i} y_j \log P(y=j)$$

$$= -\Sigma_j \frac{\partial}{h_i} y_j \log \frac{exp(h_j)}{\Sigma_k exp(h_k)}$$

$$= -\Sigma_j \frac{\partial}{h_i} y_j \left[h_j - \log\left(\Sigma_k exp(h_k)\right)\right]$$

$$= -\left(\frac{\partial}{h_i} y_i h_i + \cancel{\Sigma_{j(j\neq i)} \frac{\partial}{\partial h_i} y_j h_j} \right.$$

$$\left. - \Sigma_j \frac{\partial}{\partial h_i} y_j \log\left(\Sigma_k exp(h_k)\right)\right)$$

$$= -\left(y_i - \sum_j y_j \frac{1}{\sum_k \exp(h_k)} \cdot \frac{d}{dh_i} \sum_k \exp(h_k)\right)$$

$$= -\left(y_i - \sum_j y_j \frac{1}{\sum_k \exp(h_k)} \exp(h_i) \cdot \frac{dh_i}{dh_i} + \sum_{k(k \neq i)} 0\right)$$

$$= -\left(y_i - \sum_j y_j \frac{\exp(h_i)}{\sum_k \exp(h_k)}\right)$$

$$= -\left(y_i - \left(\cancel{y_i}^{\,1} P(y=i) + \sum_j \cancel{y_j}^{\,0}{}_{(j \neq i)} P(y=i)\right)\right)$$

$$= -\left(y_i - P(y=i)\right)$$

$$= P(y=i) - y_i$$

$$x_1 = w_1 I + b_1$$

$$x_2 = \mathrm{ReLU}(x_1)$$

$$x_3 = w_2 x_2 + b_2$$

$$x_4 = \frac{\exp(x_3)}{\sum_k \exp(x_k)}$$

$$\frac{dx_4}{dx_3} = \frac{\sum_k \exp(x_k) \cdot \exp(x_3) - \left(\exp(x_3)\right)^2}{\left(\sum_k \exp(x_k)\right)^2}$$