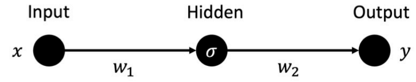
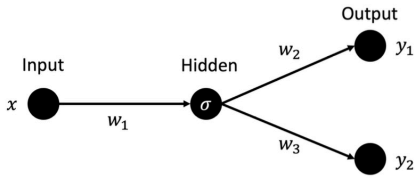
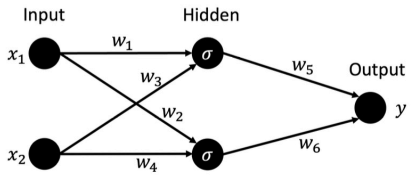
Question 1 Consider the following three neural networks:







where σ is the sigmoid activation function.

(b) Assume that you are using the following loss functions:

$$l = \sum_{i}^{n} \frac{1}{2}(y_i - t_i)^2 \tag{1}$$

where t is the target and n is the number of outputs for a given input. For each neural network, compute $\frac{\partial l}{\partial w_i}$ for each w_i (i.e. $\frac{\partial l}{\partial w_1}$, $\frac{\partial l}{\partial w_2}$, etc ...). (hint: use the chain rule)

(c) Given the table below of inputs and weights, compute the output of each network. Show your work.

	Input	Weights
Network 1	$\mathbf{x} = 0.7$	$\mathbf{w} = [0.4, 0.5]$
Network 2	$\mathbf{x} = 0.3$	$\mathbf{w} = [0.1, 0.2, 0]$
Network 3	$\mathbf{x} = [0.4, 0.8]$	$\mathbf{w} = [1, 0.2, 0.2, 0, 0.5, 0.1]$

a) ①

$z_1 = x w_1$

$a_1 = \sigma(z_1)$

$y = a_1 w_2$

②

$z_1 = x w_1$

$a_1 = \sigma(z_1)$

$y_1 = a_1 w_2$

$y_2 = a_1 w_3$

③

$z_1 = x_1 w_1 + x_2 w_3$

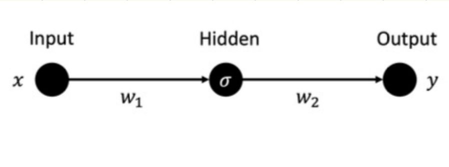
$z_2 = x_1 w_2 + x_2 w_4$

$a_1 = \sigma(z_1)$

$a_2 = \sigma(z_2)$

$y = a_1 w_5 + a_2 w_6$

b)



$z = x w_1$

$a = \sigma(z)$

$y = a_1 w_2$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial a}\frac{\partial a}{\partial z}\frac{\partial z}{\partial w_1}$$

$$\frac{\partial L}{\partial y} = y - t$$

$$\frac{\partial y}{\partial a} = w_2$$

$$\frac{\partial a}{\partial z} = \sigma(z)(1-\sigma(z))$$

$$\frac{\partial z}{\partial w_1} = x$$

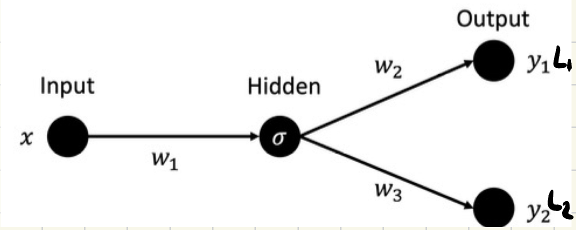
$$\frac{\partial L}{\partial w_1} = (y-t)w_2(\sigma(z)(1-\sigma(z))x$$

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial w_2}$$

$$\frac{\partial L}{\partial y} = y - t$$

$$\frac{\partial y}{\partial w_2} = a$$

$$\frac{\partial L}{\partial w_2} = (y-t)a$$



$z = x w_1$ $\quad$ $y_1 = a w_2$

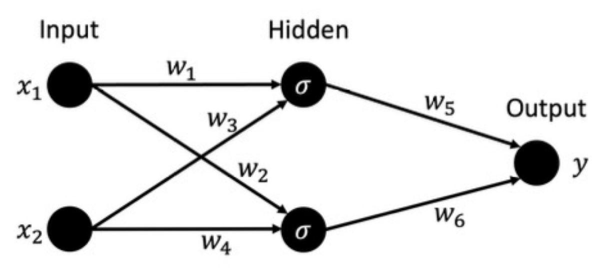
$a = \sigma(z)$ $\quad$ $y_2 = a w_3$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L_1}{\partial y_1}\frac{\partial y_1}{\partial a}\frac{\partial a}{\partial z}\frac{\partial z}{\partial w_1} + \frac{\partial L_2}{\partial y_2}\frac{\partial y_2}{\partial a}\frac{\partial a}{\partial z}\frac{\partial z}{\partial w_1}$$

$\frac{\partial L_1}{\partial y_1} = y_1 - t_1$	$\frac{\partial L_2}{\partial y_2} = y_2 - t_2$
$\frac{\partial y_1}{\partial a} = w_2$	$\frac{\partial y_2}{\partial a} = w_3$
$\frac{\partial a}{\partial z} = \sigma(z)(1-\sigma(z))$	$\frac{\partial a}{\partial z} = \sigma(z)(1-\sigma(z))$
$\frac{\partial z}{\partial w_1} = x$	$\frac{\partial z}{\partial w_1} = x$

$$\frac{\partial L}{\partial w_1} = (y_1 - t_1) w_2 (\sigma(z)(1-\sigma(z))x + (y_2 - t_2) w_3 (\sigma(z)(1-\sigma(z))x$$

$\frac{\partial L_1}{\partial w_2} = \frac{\partial L_1}{\partial y_1}\frac{\partial y_1}{\partial w_2}$	$\frac{\partial L_2}{\partial w_3} = \frac{\partial L_2}{\partial y_2}\frac{\partial y_2}{\partial w_3}$
$\frac{\partial L_1}{\partial y_1} = y_1 - t_1$	$\frac{\partial L_2}{\partial y_2} = (y_2 - t_2)$
$\frac{\partial y_1}{\partial w_2} = a$	$\frac{\partial y_2}{\partial w_3} = a$
$\frac{\partial L_1}{\partial w_2} = (y_1 - t_1)a$	$\frac{\partial L_2}{\partial w_3} = (y_2 - t_2)a$



$$z_1 = x_1 w_1 + x_2 w_3$$
$$z_2 = x_1 w_2 + x_2 w_4$$

$$a_1 = \sigma(z_1)$$
$$a_2 = \sigma(z_2)$$

$$y = a_1 w_5 + a_2 w_6$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial y} \frac{\partial y}{\partial a_1} \frac{\partial a_1}{\partial z_1} \frac{\partial z_1}{\partial w_1}$$

$$\frac{\partial L}{\partial y} = y - t \qquad \frac{\partial y}{\partial a_1} = w_5$$

$$\frac{\partial a_1}{\partial z_1} = \sigma(z_1)(1-\sigma(z_1)) \qquad \frac{\partial z_1}{\partial w_1} = x_1$$

$$\frac{\partial L}{\partial w_1} = (y-t) w_5 \, \sigma(z_1)(1-\sigma(z_1)) \, x_1$$

$$\frac{\partial L}{\partial w_4} = \frac{\partial L}{\partial y} \frac{\partial y}{\partial a_2} \frac{\partial a_2}{\partial z_2} \frac{\partial z_2}{\partial w_4}$$

$$\frac{\partial L}{\partial y} = y - t \qquad \frac{\partial y}{\partial a_2} = w_6$$

$$\frac{\partial a_2}{\partial z_2} = \sigma(z_2)(1-\sigma(z_2)) \qquad \frac{\partial z_2}{\partial w_4} = x_2$$

$$\frac{\partial L}{\partial w_1} = (y-t) w_6 \, \sigma(z_2)(1-\sigma(z_2)) \, x_2$$

$$\frac{\partial L}{\partial w_3} = \frac{\partial L}{\partial y} \frac{\partial y}{\partial a_1} \frac{\partial a_1}{\partial z_1} \frac{\partial z_1}{\partial w_3}$$

$$\frac{\partial z_1}{\partial w_3} = x_2$$

$$\frac{\partial L}{\partial w_3} = (y-t) w_5 \, \sigma(z_1)(1-\sigma(z_1)) \, x_2$$

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial y} \frac{\partial y}{\partial a_2} \frac{\partial a_2}{\partial z_2} \frac{\partial z_2}{\partial w_2}$$

$$\frac{\partial z_2}{\partial w_2} = x_1$$

$$\frac{\partial L}{\partial w_2} = (y-t) w_6 \, \sigma(z_2)(1-\sigma(z_2)) \, x_1$$

$$\frac{\partial L}{\partial w_5} = \frac{\partial L}{\partial y} \frac{\partial y}{\partial w_5}$$

$$\frac{\partial y}{\partial w_5} = a_1$$

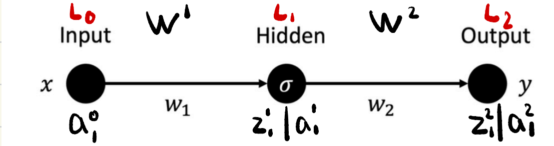
$$\frac{\partial L}{\partial w_5} = (y-t) a_1$$

$$\frac{\partial L}{\partial w_6} = \frac{\partial L}{\partial y} \frac{\partial y}{\partial w_6}$$

$$\frac{\partial y}{\partial w_6} = a_2$$

$$\frac{\partial L}{\partial w_1} = (y-t) a_2$$

c)

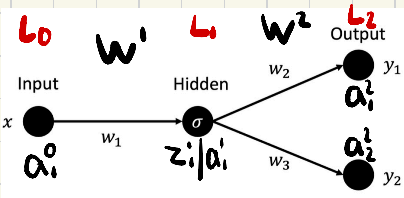


$$W^1 = [w_1] \qquad W^2 = [w_2]$$

$$z^1 = W^1 X = [0.4][0.7] = 0.28$$

$$a^1 = \sigma(z^1) = \sigma(0.28) = 0.5695$$

$$y = W^2 a^1 = [0.5][0.5695] = [0.2847]$$

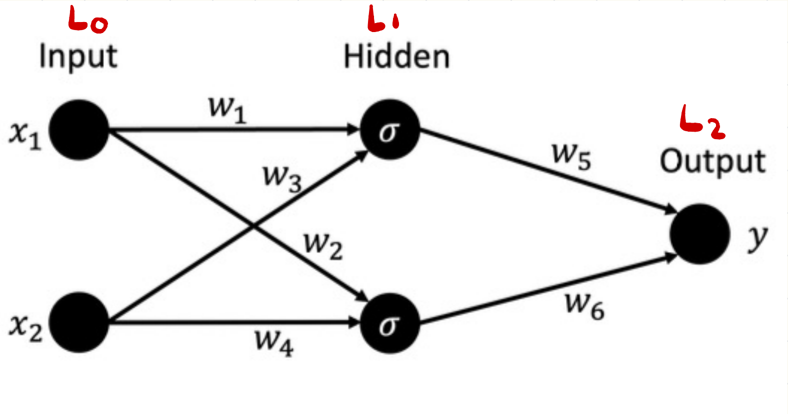


$$W^1 = [0.1] \qquad W^2 = \begin{bmatrix} W_{11} \\ W_{21} \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0 \end{bmatrix}$$

$$z^1 = W^1 X = [0.1][0.3] = 0.03$$

$$a^1 = \sigma(z^1) = \sigma(0.03) = 0.5075$$

$$y = W^2 a^1 = \begin{bmatrix} 0.2 \\ 0 \end{bmatrix} [0.5075] = \begin{bmatrix} 0.1015 \\ 0 \end{bmatrix}$$



$$W^1 = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} = \begin{bmatrix} w_1 & w_3 \\ w_2 & w_4 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0.2 \\ 0.2 & 0 \end{bmatrix}$$

$$W^2 = \begin{bmatrix} w_{11} & w_{12} \end{bmatrix} = \begin{bmatrix} w_5 & w_6 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.5 & 0.1 \end{bmatrix}$$

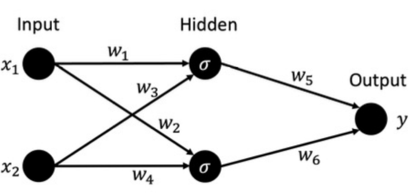
$$z^1 = W^1 X = \begin{bmatrix} 1 & 0.2 \\ 0.2 & 0 \end{bmatrix} \begin{bmatrix} 0.4 \\ 0.8 \end{bmatrix} = \begin{bmatrix} 0.56 \\ 0.08 \end{bmatrix}$$

$$a^1 = \sigma(z^1) = \begin{bmatrix} \sigma(0.56) \\ \sigma(0.08) \end{bmatrix} = \begin{bmatrix} 0.6365 \\ 0.52 \end{bmatrix}$$

$$y = W^2 a^1 = \begin{bmatrix} 0.5 & 0.1 \end{bmatrix} \begin{bmatrix} 0.6365 \\ 0.52 \end{bmatrix} = \begin{bmatrix} 0.37025 \end{bmatrix}$$

Question 2 Consider the 3rd network and the loss given the previous question, with the additional assumption that the output has a sigmoid activation function. The following are steps resulting from deriving the loss for that specific case (Reference). The aim in this question is to perform Backpropagation. Assume that the weights are initialized as ($w_1 = 0.2$, $w_2 = 0.1$, $w_3 = 1.0$, $w_4 = -1.5$, $w_5 = 0.7$, $w_6 = -0.3$), and that you are using sigmoid as the activation function at each neuron, with a learning rate of $\eta = 0.1$. Use the data point $\mathbf{x} = [0.2, 0.9]$ with the target $y = 0.7$.

(a) Perform a forward pass through the network, showing the final output as well as the output at each hidden unit.



$$X = \begin{bmatrix} 0.2 \\ 0.9 \end{bmatrix} \qquad W^2 = [w_{11} \;\; w_{12}]$$

$$[w_5 \;\; w_6]$$

$$[0.7 \;\; -0.3]$$

$$W^1 = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} = \begin{bmatrix} w_1 & w_3 \\ w_2 & w_4 \end{bmatrix} = \begin{bmatrix} 0.2 & 1 \\ 0.1 & -1.5 \end{bmatrix}$$

Forward

$$z^1 = W^1 X = \begin{bmatrix} 0.2 & 1 \\ 0.1 & -1.5 \end{bmatrix} \begin{bmatrix} 0.2 \\ 0.9 \end{bmatrix} = \begin{bmatrix} 0.94 \\ -1.33 \end{bmatrix}$$ (Preactivation)

$$a^1 = \sigma(z^1) = \begin{bmatrix} \sigma(0.94) \\ \sigma(-1.33) \end{bmatrix} = \begin{bmatrix} 0.71909966 \\ 0.20915937 \end{bmatrix}$$

$$z^2 = W^2 a^1 = [0.7 \;\; -0.3] \begin{bmatrix} 0.71909966 \\ 0.20915937 \end{bmatrix} = [0.440621951]$$

$$y = \sigma(z^2) = \sigma(0.440621951) = 0.6084072192$$

(b) For each output unit k, the error term δ_k is calculated as $\delta_k \leftarrow g'(x_k) \times Err_k$, where $g(x)$ is the activation function, x_k is the input to that unit, and Err_k is the error between the output O_k and the target T_k, given as $Err_k = O_k - T_k$. Derive an expression for δ_k in terms of O_k and T_k, and use this expression to compute δ_k for the output of your network.

$$\delta_k = g'(x_k)\, Err_k$$

$$= \sigma'(x_k)\, O_k - T_k$$

$$\delta_k = \sigma(x_k)(1 - \sigma(x_k))\, O_k - T_k \Rightarrow O_k(1 - O_k)(O_k - t_k)$$

$O_1 = 0.6084$

$T_1 = 0.7$

$$\delta_1 = 0.6084(1 - 0.6084)(0.6084 - 0.7)$$

$$\delta_1 = -0.021266$$

(c) For each hidden unit h, the error term δ_h is calculated as $\delta_h \leftarrow g'(x_h) \times Err_h$, where $g(x)$ is the activation function, x_h is the input to that unit, and Err_h is the error of the output O_h, given as $Err_h = \sum_{k \in \text{outputs}} w_{hk}\delta_k$. Derive an expression for δ_h in terms of O_h, w_{hk}, and δ_k, and use this expression to compute δ_h for each of the hidden units.

$$\delta_h = g'(X_h)(Err_h)$$

$$\sigma'(X_h)(w_{hk}\,\delta_k)$$

$$\sigma(X_h)\,(1-\sigma(X_h))\,(w_{hk}\,\delta_k)$$

$$h = \begin{bmatrix} 0.71909966 \\ 0.20915937 \end{bmatrix}$$

$$\delta_h = O_h\,(1-O_h)\,(w_{hk}\,\delta_k)$$

$O_{h1} = 0.7191$

$w_{h1k} = w_5 = 0.7$

$\delta_k = -0.021266$

$O_{h2} = 0.2092$

$w_{h2k} = w_6 = -0.3$

$\delta_k = -0.021266$

$$\delta_{h1} = 0.7191(1-0.7191)(0.7(-0.021266))$$

$$\delta_{h1} = -0.00300694$$

$$\delta_{h2} = 0.2092(1-0.2092)(-0.3(-0.021266))$$

$$\delta_{h2} = 0.00105544$$

(d) Each weight w_{ij} connecting nodes i and j is updated as $w_{ij} \leftarrow w_{ij} + \Delta w_{ij}$, where $\Delta w_{ij} = -\eta \delta_j O_i$. Update all the weights in your network.

$$w_{11}^{1} = w_1 - \eta \delta_{h_1} O_1$$

$$w_1 = 0.2 - 0.1(-0.00300694(0.2)) = 0.200060138$$

$$w_{12}^{1} = w_2 - \eta \delta_{h_2} O_1$$

$$w_2 = 0.1 - 0.1(0.00105544(0.2)) = 0.099978891$$

$$w_{21}^{1} = w_3 - \eta \delta_1 O_2$$

$$w_3 = 1 - 0.1(-0.00300694(0.9)) = 1.000270625$$

$$w_{22}^{1} = w_4 - \eta \delta_2 O_2$$

$$w_4 = -1.5 - 0.1(0.00105544(0.9)) = -1.50009499$$

$$w_{11}^{2} = w_5 - \eta \delta_1 O_1$$

$$w_5 = 0.7 - 0.1(-0.021266(0.71909966)) = 0.7015$$

$$w_{21}^{2} = w_6 - \eta \delta_1 O_2$$

$$w_6 = -0.3 - 0.1(-0.021266(0.20915937)) = -0.2996$$

(e) Perform a forward pass and compute the error at the output. Compare the error with that from the initial forward pass and comment on the results.

$$W^1 = \begin{bmatrix} w_1 & w_3 \\ w_2 & w_4 \end{bmatrix} = \begin{bmatrix} 0.200060138 & 1.000270625 \\ 0.099978891 & -1.500099499 \end{bmatrix}$$

$$z^1 = W^1 x = \begin{bmatrix} 0.9402555901 \\ -1.3300897128 \end{bmatrix}$$

$$a^1 = \sigma(z^1) = \begin{bmatrix} 0.71915128 \\ 0.20914452 \end{bmatrix}$$

$$W^2 = [0.7015 \quad -0.2996]$$

$$z^2 = W^2 a^1 = [0.44207853]$$

$$y = \sigma(z^2) = 0.60875419$$

Error after 1^{st}: $L_1 = \frac{1}{2}(y-t)^2 \Rightarrow \frac{1}{2}(0.6084072192 - 0.7)^2$
$= 0.004194618$

Error after 2^{nd}: $L_2 = \frac{1}{2}(y-t)^2 \Rightarrow \frac{1}{2}(0.60875419 - 0.7)^2$
$= 0.004162898$

L_2 is smaller than L_1 which tells us that our 1 itteration of training worked well

(f) Explain, in your own words, what would change in this process above if you are to use a ReLU activation function instead of sigmoid.

The process would be the exact same except that when we derived the sigmoid we will instead derive the ReLU function which would give us different numbers for the "after activation" values.

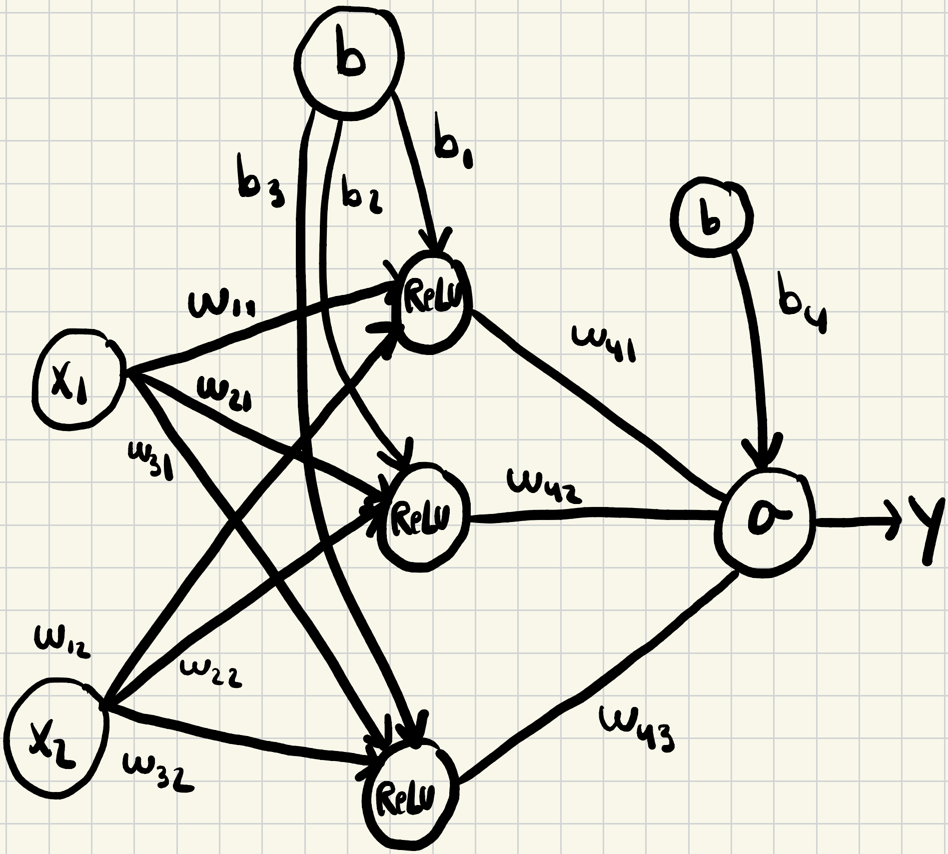
Derivative of ReLU:

$$\mathrm{ReLU}'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x > 0 \end{cases}$$

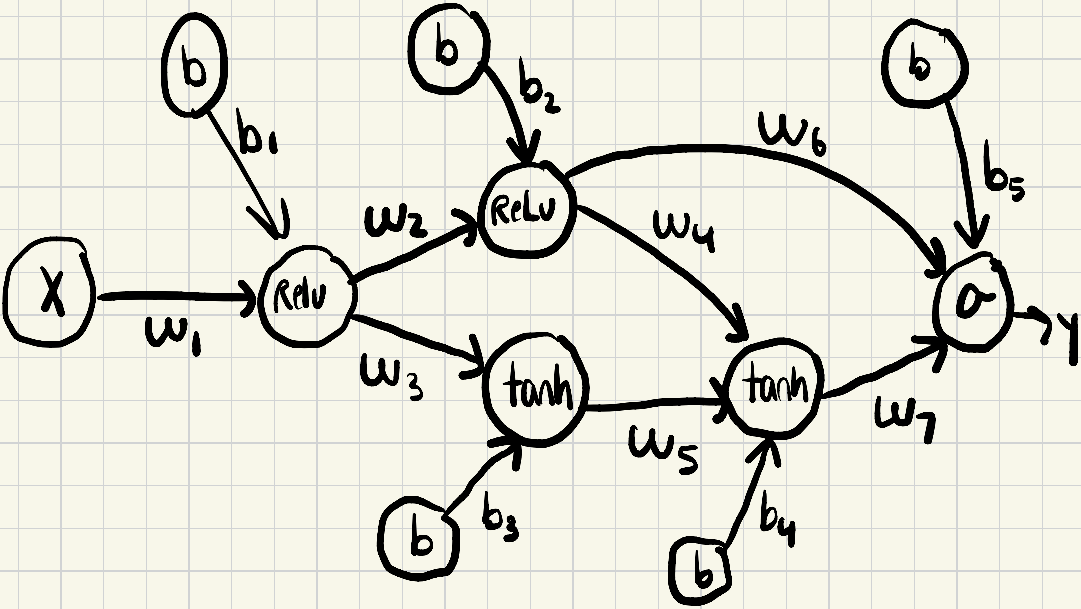
Question 3 For each of the following equations, draw the architecture of the neural network. You need to label each of the nodes, edges, and activation functions.

(a)

$$y = \sigma\Big(w_{41} \cdot \mathrm{ReLU}(w_{11}x_1 + w_{12}x_2 + b_1) + w_{42} \cdot \mathrm{ReLU}(w_{21}x_1 + w_{22}x_2 + b_2) + w_{43} \cdot \mathrm{ReLU}(w_{31}x_1 + w_{32}x_2 + b_3) + b_4\Big)$$



$$y = \sigma\Big(w_6 \cdot \mathrm{ReLU}\big(w_2 \cdot \mathrm{ReLU}(w_1 x + b_1) + b_2\big) + w_7 \cdot \tanh\Big(w_4 \cdot \mathrm{ReLU}\big(w_2 \cdot \mathrm{ReLU}(w_1 x + b_1) + b_2\big) + w_5 \cdot \tanh\big(w_3 \cdot \mathrm{ReLU}(w_1 x + b_1) + b_3\big) + b_4\Big) + b_5\Big)$$



Question 4 For each of the following 2D maps (images), apply convolution using the given filter with the specified parameters (assume no padding):

(a) Stride = 1

1	0	0	1	0
0	1	1	1	0
0	0	0	0	0
1	0	0	1	1
1	1	0	1	1

Image

1	3	1
2	4	0
2	3	5

Filter

$Y_{11} = 1\times1+0\times3+0\times1+0\times2+1\times4+1\times0+0\times2+0\times3+0\times5 = 5$

$Y_{12} = 0\times1+0\times3+1\times1+1\times2+1\times4+1\times0+0\times2+0\times3+0\times5 = 7$

$Y_{13} = 0\times1+1\times3+0\times1+1\times2+1\times4+0\times0+0\times2+0\times3+0\times5 = 9$

$Y_{21} = 0\times1+1\times3+1\times1+0\times2+0\times4+0\times0+1\times2+0\times3+0\times5 = 6$

$Y_{22} = 1\times1+1\times3+1\times1+0\times2+0\times4+0\times0+0\times2+0\times3+1\times5 = 10$

$Y_{23} = 1\times1+1\times3+0\times1+0\times2+0\times4+0\times0+0\times2+1\times3+1\times5 = 12$

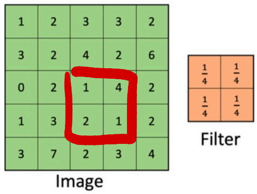
$Y_{23} = 0\times1+0\times3+0\times1+1\times2+0\times4+0\times0+1\times2+1\times3+0\times5 = 7$

$Y_{23} = 0\times1+0\times3+0\times1+0\times2+0\times4+1\times0+1\times2+0\times3+1\times5 = 7$

$Y_{23} = 0\times1+0\times3+0\times1+0\times2+1\times4+1\times0+0\times2+1\times3+1\times5 = 12$

5	7	9
6	10	12
7	7	12

(b) Stride $= 2$



$$Y_{11} = 1 \times \tfrac{1}{4} + 2 \times \tfrac{1}{4} + 3 \times \tfrac{1}{4} + 2 \times \tfrac{1}{4} = 2$$

$$Y_{12} = 3 \times \tfrac{1}{4} + 3 \times \tfrac{1}{4} + 4 \times \tfrac{1}{4} + 2 \times \tfrac{1}{4} = 3$$

$$Y_{21} = 0 \times \tfrac{1}{4} + 2 \times \tfrac{1}{4} + 1 \times \tfrac{1}{4} + 3 \times \tfrac{1}{4} = 1.5$$

$$Y_{22} = 1 \times \tfrac{1}{4} + 4 \times \tfrac{1}{4} + 2 \times \tfrac{1}{4} + 1 \times \tfrac{1}{4} = 2$$

2	3
1.5	2

(c) MaxPooling with kernel size = 2

4	2	4	4	3	4
4	3	5	2	6	1
2	4	3	4	2	4
1	6	4	1	2	3
3	7	2	5	6	1
5	4	1	2	4	5

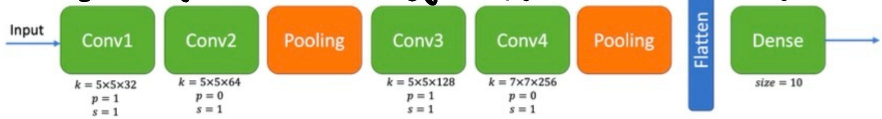
$$\begin{bmatrix} 4 & 5 & 6 \\ 6 & 4 & 4 \\ 7 & 5 & 6 \end{bmatrix}$$

CONUUITION AFTER POOLING ?! Stride?!

Question 5 For each of the architectures given below, write down the output dimensions (in $H \times W \times N$ format) of each layer. Refer to PyTorch documentations on how to handle cases with fractions. (note: in this question, pooling refers to Max Pooling layer with a 2x2 kernel and a stride of 2). The input image size is $224 \times 224 \times 3$.

(a)

222 222 32 | 218 218 64 | 109 109 64 | 107 107 128 | 101 101 256 | 50 50 256 | 640 000 1 | 10 1



$$O_c = \frac{I - k + 2P}{S} + 1 \qquad O_p = \frac{I - k}{S} + 1$$

$224 \times 224 \times 3 \rightarrow \boxed{\text{Conv 1}} \rightarrow 222 \times 222 \times 32 \rightarrow \boxed{\text{Conv 2}} \rightarrow 218 \times 218 \times 64$

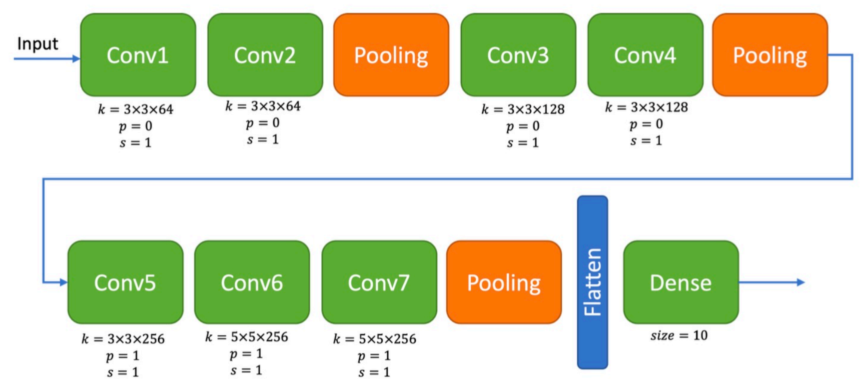
$$\frac{224 - 5 + 2(1)}{1} + 1 \qquad \frac{222 - 5 + 2(0)}{1} + 1$$

$218 \times 218 \times 64 \rightarrow \boxed{\text{Pooling}} \rightarrow 109 \times 109 \times 64 \rightarrow \boxed{\text{Conv3}} \rightarrow 107 \times 107 \times 128$

$$\frac{218 - 2}{2} + 1 \qquad \frac{109 - 5 + 2(1)}{1} + 1$$

$107 \times 107 \times 128 \rightarrow \boxed{\text{Conv4}} \rightarrow 101 \times 101 \times 256 \rightarrow \boxed{\text{Pooling}} \rightarrow 50 \times 50 \times 256 \rightarrow \boxed{\text{Flatten}} \rightarrow 640\,000 \times 1 \rightarrow 10 \times 1$

$$\frac{107 - 7 + 2(0)}{1} + 1 \qquad \left\lfloor \frac{101 - 2}{2} + 1 \right\rfloor$$



$224 \times 224 \times 3$

$$\text{Conv1} = \frac{224-3+2(0)}{1}+1 = 222 \Rightarrow 222\times222\times64$$

$$\text{Conv2} = \frac{222-3+2(0)}{1}+1 = 220 \Rightarrow 220\times220\times64$$

$$\text{pool} = \frac{220-2}{2}+1 = 110 \Rightarrow 110\times110\times64$$

$$\text{Conv3} = \frac{110-3+2(0)}{1}+1 = 108 \Rightarrow 108\times108\times128$$

$$\text{Conv4} = \frac{108-3+2(0)}{1}+1 = 106 \Rightarrow 106\times106\times128$$

$$\text{pool} = \left\lfloor \frac{106-2}{2}+1 \right\rfloor = 53 \Rightarrow 53\times53\times256$$

$$\text{Conv5} = \frac{53-3+2(1)}{1}+1 = 53 \Rightarrow 53\times53\times256$$

$$\text{Conv6} = \frac{53 - 5 + 2(1)}{1} + 1 = 51 \Rightarrow 51 \times 51 \times 256$$

$$\text{Conv7} = \frac{51 - 5 + 2(1)}{1} + 1 = 49 \Rightarrow 49 \times 49 \times 256$$

$$\text{pool} = \left\lfloor \frac{49 - 2}{2} + 1 \right\rfloor = 24 \Rightarrow 24 \times 24 \times 256$$

Flatten : $24 \times 24 \times 256 = 147\,456 \Rightarrow 147\,456 \times 1$

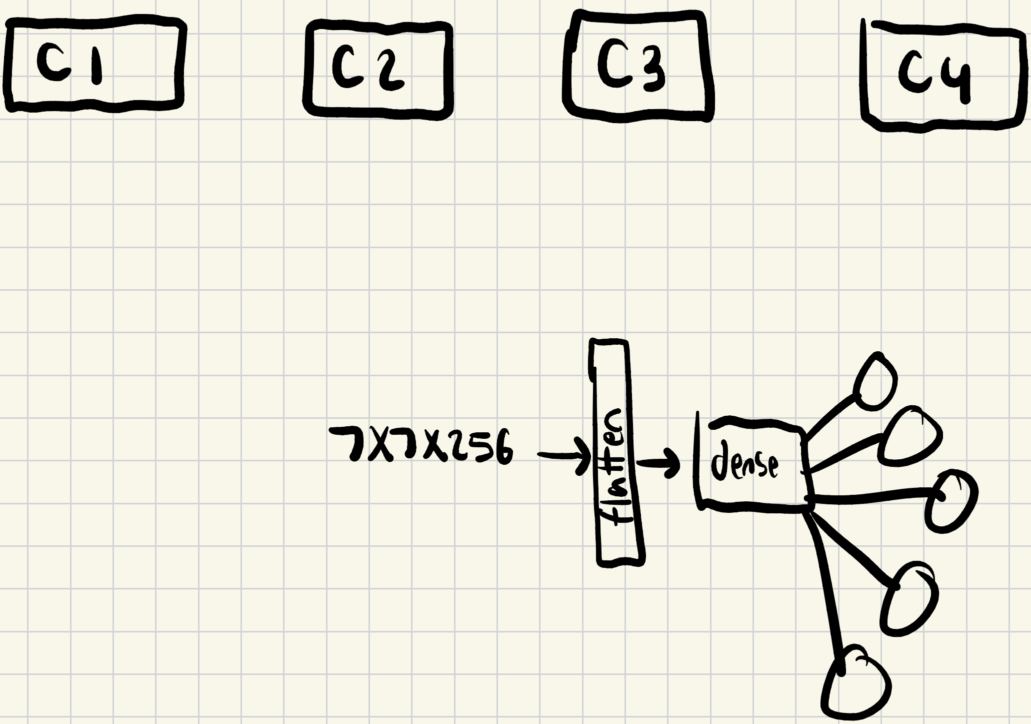
Dense: 10×1

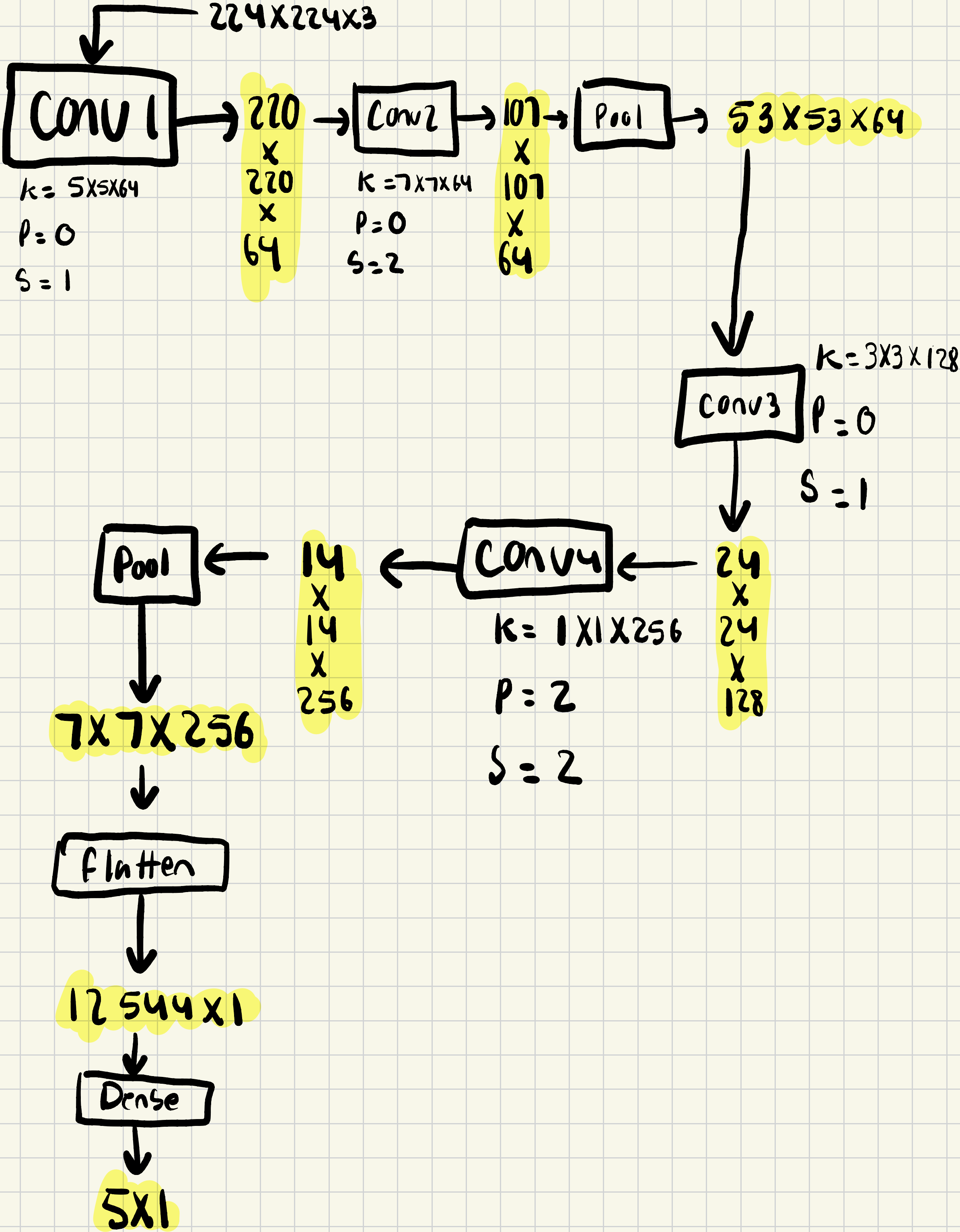
Question 6 Design a Convolutional Neural Network (CNN) for image classification with the following requirements:

- Input image size is $224 \times 224 \times 3$.
- The CNN encoder contains exactly four(4) convolutional layers.
- You are allowed a maximum of two MaxPool layers, each with a 2×2 kernel and a stride of 2.
- The output dimensions before flattening is $7 \times 7 \times 256$.
- For the convolutional layers, filters (kernels) have to be of odd dimensions with a maximum size of 7×7. The maximum padding allowed is 2 and the maximum stride allowed is 2.
- The classifier contains one Fully-Connected (FC) layer with 5 output prediction classes.

You are required to draw/sketch the whole CNN pipeline describing the details of each layer in the network (kernel size, padding, stride, input/output dimensions).

input: 224X224X3




224X224X3
Conv1
K= 5X5X64
P=0
S=1
220 X 220 X 64
Conv2
K=7X7X64
P=0
S=2
107 X 107 X 64
Pool
53X53X64
K=3X3X128
Conv3
P=0
S=1
24 X 24 X 128
Conv4
K= 1X1X256
P=2
S=2
14 X 14 X 256
Pool
7X7X256
Flatten
12544X1
Dense
5X1

Implementation Q3

I played with the batch size and learning rate until I found the a combination that trained well enough.

As we can see on the plots, the training loss rapidly decrease and converges to a value while the accuracy rapidly increases to converge to a value near 1.

This means that the training is happening as expected.

The model may be overfitting a little due to the perfect score on the test set.

I tried to reduce the overfitting by introducing a weight decay.

The model has 1 convolutional layer followed by a pooling then another convolutional and another pooling.

After this, we flatten the result (64*32*32) to introduce to a fully connected layer.

The design can probably be optimized to reduce overfitting.

```
00
00
och 1, Loss: 0.2870, Accuracy: 0.9129
och 2, Loss: 0.0215, Accuracy: 0.9931
och 3, Loss: 0.0103, Accuracy: 0.9962
och 4, Loss: 0.0109, Accuracy: 0.9962
och 5, Loss: 0.0051, Accuracy: 0.9986
och 6, Loss: 0.0056, Accuracy: 0.9983
och 7, Loss: 0.0035, Accuracy: 0.9990
och 8, Loss: 0.0021, Accuracy: 0.9998
och 9, Loss: 0.0008, Accuracy: 1.0000
och 10, Loss: 0.0038, Accuracy: 0.9993
            precision    recall  f1-score   support

 AbdomenCT       1.00      1.00      1.00       298
 BreastMRI       1.00      1.00      1.00       284
       CXR       1.00      0.99      1.00       305
   ChestCT       1.00      1.00      1.00       286
      Hand       1.00      0.99      0.99       314
    HeadCT       0.99      1.00      1.00       313

  accuracy                           1.00      1800
 macro avg       1.00      1.00      1.00      1800
ighted avg       1.00      1.00      1.00      1800
```