**Question 1** (3-3-3-4-3-4-3)**.** Consider training a standard feed-forward neural network. For the purposes of this question we are interested in a single iteration of SGD on a single training example : $(\boldsymbol{x}, y)$. We denote $f(\boldsymbol{x}, \boldsymbol{\theta})$ as the output of the neural network with model parameters $\boldsymbol{\theta}$. Now let's say $g$ is the output activation function and $a(\boldsymbol{x}, \boldsymbol{\theta})$ is the pre-activation network output such that $f(\boldsymbol{x}, \boldsymbol{\theta}) = g(a(\boldsymbol{x}, \boldsymbol{\theta}))$.

1.1 Assuming the network's goal is to do binary classification (with the detailed structure above), what would be an appropriate activation function for the output layer, i.e. what would be an appropriate function $g$ ? *We will keep this choice of g for the rest of this exercise.*
**Solution** : Since it is binary classification sigmoid=$\sigma(z) = \frac{1}{1+e^{-z}}$ can be used as an activation function.

1.2 What does the output represent under this activation function ?
**Solution :** Output of sigmoid can be interpreted as probability since its value lies between 0 and 1.

1.3 Let $L_{CE}(f(\boldsymbol{x}, \boldsymbol{\theta}), y)$ be cross-entropy loss, express it as a function of $f(\boldsymbol{x}, \boldsymbol{\theta})$ and $y$.
**Solution :**

$$L_{CE}(f(\boldsymbol{x}, \boldsymbol{\theta}), y) = \frac{1}{N} \sum_{i=1}^{N} - \left(y \cdot log(f(\boldsymbol{x}, \boldsymbol{\theta})) + (1 - y) \cdot (log(1 - f(\boldsymbol{x}, \boldsymbol{\theta})))\right)$$

Since we are interested in one training example, N=1.

$$L_{CE}(f(\boldsymbol{x}, \boldsymbol{\theta}), y) = - \left(y \cdot log(f(\boldsymbol{x}, \boldsymbol{\theta})) + (1 - y) \cdot (log(1 - f(\boldsymbol{x}, \boldsymbol{\theta})))\right)$$

1.4 Compute the partial derivative $\frac{\partial L_{CE}(f(\boldsymbol{x}, \boldsymbol{\theta}), y)}{\partial a(\boldsymbol{x}, \boldsymbol{\theta})}$.
**Solution :**

$$\frac{\partial L_{CE}(f(\boldsymbol{x}, \boldsymbol{\theta}), y)}{\partial a(\boldsymbol{x}, \boldsymbol{\theta})} = \frac{\partial}{\partial a(\boldsymbol{x}, \boldsymbol{\theta})} \left[- \left(y \cdot log(f(\boldsymbol{x}, \boldsymbol{\theta})) + (1 - y) \cdot (log(1 - f(\boldsymbol{x}, \boldsymbol{\theta})))\right)\right]$$

$$\frac{\partial L_{CE}(f(\boldsymbol{x}, \boldsymbol{\theta}), y)}{\partial a(\boldsymbol{x}, \boldsymbol{\theta})} = -[y \cdot \frac{\partial log(g(a(\boldsymbol{x}, \boldsymbol{\theta})))}{\partial a(\boldsymbol{x}, \boldsymbol{\theta})} + (1 - y) \cdot \frac{\partial log(1 - g(a(\boldsymbol{x}, \boldsymbol{\theta})))}{\partial a(\boldsymbol{x}, \boldsymbol{\theta})}] \tag{1}$$

Since $g(a(\boldsymbol{x}, \boldsymbol{\theta}))$ is a sigmoid function its derivative with respect to $a(\boldsymbol{x}, \boldsymbol{\theta})$ :

$$\frac{\partial g(a(\boldsymbol{x}, \boldsymbol{\theta}))}{\partial a(\boldsymbol{x}, \boldsymbol{\theta})} = \frac{\partial \frac{e^{a(\boldsymbol{x}, \boldsymbol{\theta})}}{1+e^{a(\boldsymbol{x}, \boldsymbol{\theta})}}}{\partial a(\boldsymbol{x}, \boldsymbol{\theta})}$$

Using quotient rule for derivative :

$$\frac{\partial g(a(\boldsymbol{x}, \boldsymbol{\theta}))}{\partial a(\boldsymbol{x}, \boldsymbol{\theta})} = \frac{(1 + e^{a(\boldsymbol{x}, \boldsymbol{\theta})}) \cdot \frac{\partial e^{a(\boldsymbol{x}, \boldsymbol{\theta})}}{\partial a(\boldsymbol{x}, \boldsymbol{\theta})} - e^{a(\boldsymbol{x}, \boldsymbol{\theta})} \cdot \frac{\partial (1+e^{a(\boldsymbol{x}, \boldsymbol{\theta})})}{\partial a(\boldsymbol{x}, \boldsymbol{\theta})}}{(1 + e^{a(\boldsymbol{x}, \boldsymbol{\theta})})^2}$$

$$= \frac{(1 + e^{a(\boldsymbol{x}, \boldsymbol{\theta})}) \cdot e^{a(\boldsymbol{x}, \boldsymbol{\theta})} - e^{a(\boldsymbol{x}, \boldsymbol{\theta})} \cdot e^{a(\boldsymbol{x}, \boldsymbol{\theta})})}{(1 + e^{a(\boldsymbol{x}, \boldsymbol{\theta})})^2}$$

$$= \frac{e^{a(\boldsymbol{x}, \boldsymbol{\theta})}}{(1 + e^{a(\boldsymbol{x}, \boldsymbol{\theta})})} - \frac{e^{a(\boldsymbol{x}, \boldsymbol{\theta})}}{(1 + e^{a(\boldsymbol{x}, \boldsymbol{\theta})})} \cdot \frac{e^{a(\boldsymbol{x}, \boldsymbol{\theta})}}{(1 + e^{a(\boldsymbol{x}, \boldsymbol{\theta})})} = g(a(\boldsymbol{x}, \boldsymbol{\theta}))[1 - g(a(\boldsymbol{x}, \boldsymbol{\theta}))]$$

Using above derivation in equation (1) :

$$\frac{\partial L_{CE}(f(\boldsymbol{x},\boldsymbol{\theta}),y)}{\partial a(\boldsymbol{x},\boldsymbol{\theta})} = -[\frac{y}{g(a(\boldsymbol{x},\boldsymbol{\theta}))}\cdot g(a(\boldsymbol{x},\boldsymbol{\theta}))[1-g(a(\boldsymbol{x},\boldsymbol{\theta}))] - \frac{(1-y)}{1-g(a(\boldsymbol{x},\boldsymbol{\theta}))}\cdot g(a(\boldsymbol{x},\boldsymbol{\theta}))[1-g(a(\boldsymbol{x},\boldsymbol{\theta}))]]$$

$$= -[y\cdot(1-g(a(\boldsymbol{x},\boldsymbol{\theta}))) - (1-y)\cdot g(a(\boldsymbol{x},\boldsymbol{\theta}))]$$

$$= -[y - y\cdot g(a(\boldsymbol{x},\boldsymbol{\theta})) - g(a(\boldsymbol{x},\boldsymbol{\theta})) + y\cdot g(a(\boldsymbol{x},\boldsymbol{\theta}))] = g(a(\boldsymbol{x},\boldsymbol{\theta})) - y$$

$$\frac{\partial L_{CE}(f(\boldsymbol{x},\boldsymbol{\theta}),y)}{\partial a(\boldsymbol{x},\boldsymbol{\theta})} = g(a(\boldsymbol{x},\boldsymbol{\theta})) - y = f(\boldsymbol{x},\boldsymbol{\theta}) - y$$

1.5 Let $L_{MSE}(f(\boldsymbol{x},\boldsymbol{\theta}),y)$ be the mean-squared error, express it as a function of $f(\boldsymbol{x},\boldsymbol{\theta})$ and $y$ (multiplicative factors can be ignored).
**Solution** $L_{MSE}(f(\boldsymbol{x},\boldsymbol{\theta}),y) = \sum_{i=1}^{n}(y_i - f(x,\theta))_i^2$. Here n=1(single training example).

$$L_{MSE}(f(\boldsymbol{x},\boldsymbol{\theta}),y) = (y - f(x,\theta))^2$$

1.6 Compute the partial derivative $\frac{\partial L_{MSE}(f(\boldsymbol{x},\boldsymbol{\theta}),y)}{\partial a(\boldsymbol{x},\boldsymbol{\theta})}$.
**Solution :**

$$\frac{\partial L_{MSE}(f(\boldsymbol{x},\boldsymbol{\theta}),y)}{\partial a(\boldsymbol{x},\boldsymbol{\theta})} = \frac{\partial \sum_{i=1}^{n}(y - f(\boldsymbol{x},\boldsymbol{\theta}))^2}{\partial a(\boldsymbol{x},\boldsymbol{\theta})} = \frac{\partial \sum_{i=1}^{n}(y - g(a(\boldsymbol{x},\boldsymbol{\theta})))^2}{\partial a(\boldsymbol{x},\boldsymbol{\theta})}$$

$$= 2\,(y - g(a(\boldsymbol{x},\boldsymbol{\theta})))\,\frac{\partial\,[y - g(a(\boldsymbol{x},\boldsymbol{\theta}))]}{\partial a(\boldsymbol{x},\boldsymbol{\theta})}$$

As $g(a(x,\theta))$ is considered as sigmoid activation function.$g(a(\boldsymbol{x},\boldsymbol{\theta})) = \frac{e^{a(\boldsymbol{x},\boldsymbol{\theta})}}{1+e^{a(x,\theta)}}$

$$\frac{\partial L_{MSE}(f(\boldsymbol{x},\boldsymbol{\theta}),y)}{\partial a(\boldsymbol{x},\boldsymbol{\theta})} = 2\,(y - g(a(\boldsymbol{x},\boldsymbol{\theta})))\,\frac{\partial\left[y - \frac{e^{a(\boldsymbol{x},\boldsymbol{\theta})}}{1+e^{a(\boldsymbol{x},\boldsymbol{\theta})}}\right]}{\partial a(\boldsymbol{x},\boldsymbol{\theta})} \tag{2}$$

Using quotient rule,

$$\frac{\partial \frac{e^{a(\boldsymbol{x},\boldsymbol{\theta})}}{1+e^{a(\boldsymbol{x},\boldsymbol{\theta})}}}{\partial a(x,\theta)} = \frac{e^{a(\boldsymbol{x},\boldsymbol{\theta})}\cdot(1+e^{a(\boldsymbol{x},\boldsymbol{\theta})}) - e^{a(\boldsymbol{x},\boldsymbol{\theta})}\cdot e^{a(\boldsymbol{x},\boldsymbol{\theta})}}{(1+e^{a(\boldsymbol{x},\boldsymbol{\theta})})^2}$$

$$= \frac{e^{a(\boldsymbol{x},\boldsymbol{\theta})}}{1+e^{a(\boldsymbol{x},\boldsymbol{\theta})}}\cdot(1 - \frac{e^{a(\boldsymbol{x},\boldsymbol{\theta})}}{1+e^{a(\boldsymbol{x},\boldsymbol{\theta})}}) = g(a(\boldsymbol{x},\boldsymbol{\theta}))\cdot(1 - g(a(\boldsymbol{x},\boldsymbol{\theta})))$$

Substituting value of $\frac{\partial \frac{e^{a(\boldsymbol{x},\boldsymbol{\theta})}}{1+e^{a(\boldsymbol{x},\boldsymbol{\theta})}}}{\partial a(x,\theta)}$ in equation (2)

$$\frac{\partial L_{MSE}(f(\boldsymbol{x},\boldsymbol{\theta}),y)}{\partial a(\boldsymbol{x},\boldsymbol{\theta})} = -2\,(y - g(a(\boldsymbol{x},\boldsymbol{\theta})))\cdot g(a(\boldsymbol{x},\boldsymbol{\theta}))\cdot(1 - g(a(\boldsymbol{x},\boldsymbol{\theta})))$$

$$\frac{\partial L_{MSE}(f(\boldsymbol{x},\boldsymbol{\theta}),y)}{\partial a(\boldsymbol{x},\boldsymbol{\theta})} = -2\,(y - f(\boldsymbol{x},\boldsymbol{\theta}))\cdot f(\boldsymbol{x},\boldsymbol{\theta})\cdot(1 - f(\boldsymbol{x},\boldsymbol{\theta}))$$

1.7 Based on your answers to the above questions, what would be the more appropriate loss function for binary classification and why ?

**Solution :** Cross entropy is more appropriate loss function than mean square error for binary class classification because for binary classification we want our output between 2 values so that we can apply threshold and determine which class it belongs, here since we use sigmoid our output would be between 0 and 1 so that we are certain about which class to choose. Cross entropy as a loss function for binary classification fulfils above condition whereas mean sqaure error is not bounded. If we use mean square error where output from sigmoid activation function saturates easily and produces values between 0 to 1 its value will be very large even if gradients were very small and hence little learning would happen.

**Question 2** (4-4-5-6)**.** Recall the definition of the softmax function : $S(\boldsymbol{x})_i = e^{\boldsymbol{x}_i}/\sum_j e^{\boldsymbol{x}_j}$.

2.1 Show that softmax is translation-invariant, that is : $S(\boldsymbol{x} + c) = S(\boldsymbol{x})$, where $c$ is a scalar constant.

**Solution :** To prove that softmax is translational invariant we show that $S(\boldsymbol{x} + c) = S(x)$ where $c$ is a scalar constant.
softmax= $S(\boldsymbol{x}) = \frac{e^{\boldsymbol{x}_i}}{\sum_j e^{\boldsymbol{x}_j}}$

$$S(\boldsymbol{x}+c) = \frac{e^{\boldsymbol{x}_i+c}}{\sum_j e^{\boldsymbol{x}_j+c}} = \frac{e^{\boldsymbol{x}_i}e^c}{\sum_j e^c e^{\boldsymbol{x}_j}} = \frac{e^{\boldsymbol{x}_i}e^c}{e^c\sum_j e^{\boldsymbol{x}_j}}$$
$$= \frac{e^{\boldsymbol{x}_i}}{\sum_j e^{\boldsymbol{x}_j}} = S(\boldsymbol{x})$$

Hence $S(\boldsymbol{x} + c) = S(\boldsymbol{x})$, so softmax is translational-invariant.

2.2 Let $\boldsymbol{x}$ be a 2-dimensional vector. One can represent a 2-class categorical probability using softmax $S(\boldsymbol{x})$. Show that $S(\boldsymbol{x})$ can be reparameterized using sigmoid function, i.e. $S(\boldsymbol{x}) = [\sigma(z), 1 - \sigma(z)]^\top$ where $z$ is a scalar function of $\boldsymbol{x}$.

**Solution :** As it is given that one can represent 2-class categorical probability using softmax. Lets suppose Probability of class 1 is $\frac{e^{x_1}}{e^{x_1}+e^{x_2}}$ and probability of class 2 is $\frac{e^{x_2}}{e^{x_1}+e^{x_2}}$.

$$S(\boldsymbol{x}) = \begin{bmatrix} \frac{e^{x_1}}{e^{x_1}+e^{x_2}} \\ \frac{e^{x_2}}{e^{x_1}+e^{x_2}} \end{bmatrix}$$

Since it is softmax function and output of $S(\boldsymbol{x})$ represents probability,
we can write $\frac{e^{x_1}}{e^{x_1}+e^{x_2}} + \frac{e^{x_2}}{e^{x_1}+e^{x_2}} = 1$ and $\frac{e^{x_2}}{e^{x_1}+e^{x_2}} = 1 - \frac{e^{x_1}}{e^{x_1}+e^{x_2}}$

$$S(\boldsymbol{x}) = \begin{bmatrix} \frac{e^{x_1}}{e^{x_1}+e^{x_2}} \\ 1 - \frac{e^{x_1}}{e^{x_1}+e^{x_2}} \end{bmatrix}$$
$$S(\boldsymbol{x}) = \begin{bmatrix} \frac{1}{1+e^{x_2-x_1}} \\ 1 - \frac{1}{1+e^{x_2-x_1}} \end{bmatrix}$$

Let $z = x_1 - x_2$,

$$S(\boldsymbol{x}) = \begin{bmatrix} \frac{1}{1+e^{-z}} \\ 1 - \frac{1}{1+e^{-z}} \end{bmatrix} = \begin{bmatrix} \sigma(z) \\ 1 - \sigma(z) \end{bmatrix}$$

Hence $S(\boldsymbol{x})$ can be reparameterized using sigmoid function, $S(\boldsymbol{x}) = [\sigma(z), 1 - \sigma(z)]^\top$ where z is scalar function of $\boldsymbol{x}$.

2.3 Let $\boldsymbol{x}$ be a $K$-dimensional vector ($K \geq 2$). Show that $S(\boldsymbol{x})$ can be represented using $K-1$ parameters, i.e. $S(\boldsymbol{x}) = S([0, y_1, y_2, ..., y_{K-1}]^\top)$, where $y_i$ is a scalar function of $\boldsymbol{x}$ for $i \in \{1, ..., K-1\}$.

**Solution :** From above question we see that for 2 class, to calculate softmax $S(x)$ we divide numerator by denominator and we get $1+e^{x_2-x_1}$ in denominator. For three class, we will divide with same numerator and get $1+e^{x_3-x_1}+e^{x_2-x_1}$ in denominator where $y1 = x_3-x_1, y2 = x_2-x_1$ which represents 3 class softmax in terms of 2 variables i.e $y_1$ and $y_2$. For K class, softmax can be written as S(x) = $S([x_1, x_2, x_3, ..., x_K]^\top)$

$$S(x) = S([x_1 - x_1, x_2 - x_1, x_3 - x_1, ..., x_K - x_1]^\top)$$

$$S(x) = S([0, x_2 - x_1, x_3 - x_1, ..., x_K - x_1]^\top)$$

As given $y_i$ is a scalar function of $\boldsymbol{x}$ for $i \in \{1, ..., K-1\}$. Let $y_1 = x_2-x_1, y_2 = x_3-x_1, ..., y_{K-1} = x_K - x_1$. Replacing this values in $S(x) = S([0, x_2 - x_1, x_3 - x_1, ..., x_K - x_1]^\top)$, where $y_i$ is a scalar function of $\boldsymbol{x}$ for $i \in \{1, ..., K-1\}$. By induction, we can write K-class softmax using K-1 variables.

$$S(x) = S([0, y_1, y_2, ..., y_{K-1}]^\top)$$

2.4 Show that the Jacobian of the softmax function $J_{\text{softmax}}(\boldsymbol{x})$ can be expressed as : $\mathbf{Diag}(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^\top$, where $\boldsymbol{p} = S(\boldsymbol{x})$. **Solution :** $S(\boldsymbol{x})_i = \frac{e^{x_i}}{\sum_{k=1}^n e^{x_k}}$, where $\boldsymbol{x} \in \mathbb{R}^n \mapsto S(\boldsymbol{x}) \in \mathbb{R}^n$

$$\frac{\partial S(\boldsymbol{x})_i}{\partial \boldsymbol{x}_j} = \frac{\partial \frac{e^{x_i}}{\sum_{k=1}^n e^{x_k}}}{\partial \boldsymbol{x}_j}$$

Let here $g(x) = e^{x_i}$, $h(x) = \sum_{k=1}^n e^{x_j}$ For Case $i = j$ : Derivative of $g(x)$ with respect to $x_j$ is $e^{x_j}$ when $i = j$ otherwise the derivative is zero.
Derivative of $h(x)$ with respect to $x_j$

$$\frac{\partial h(x)}{\partial x_j} = \frac{\partial \sum_{k=1}^N e^{x_k}}{\partial x_j} = e^{x_j} \text{for k=j}$$

Using quotient Rule : $f'(x) = \frac{g'(x)h(x) - h'(x)g(x)}{[h(x)]^2}$

$$\frac{\partial \frac{e^{x_i}}{\sum_{k=1}^N e^{x_k}}}{\partial x_j} = \frac{e^{x_i} \sum_{k=1}^n e^{x_k} - e^{x_j} e^{x_i}}{(\sum_{k=1}^N e^{x_k})^2}$$

$$= \frac{e^{x_i}}{\sum_{k=1}^n e^{x_k}} - \frac{e^{x_j}}{\sum_{k=1}^n e^{x_k}} \frac{e^{x_i}}{\sum_{k=1}^n e^{x_k}}$$

$$= S(x)_i - S(x)_j S(x)_i$$

, where $\frac{e^{x_j}}{\sum_{k=1}^n e^{x_k}} = S(x)_j$, and $\frac{e^{x_i}}{\sum_{k=1}^n e^{x_k}} = S(x)_i$

For Case $i \neq j$, $\frac{\partial e^{x_i}}{\partial x_j} = 0$ :

$$\frac{\partial \frac{e^{x_i}}{\sum_{k=1}^N e^{x_k}}}{\partial x} = \frac{(0 * \sum_{k=1}^N e^{x_k}) - (e^{x_j} e^{x_i})}{(\sum_{k=1}^N e^{x_k})^2}$$

$$= \frac{-(e^{x_j}e^{x_i})}{(\sum_{k=1}^{N} e^{x_k})^2}$$
$$= \frac{-e^{x_j}}{\sum_{k=1}^{N} e^{x_k}} \frac{e^{x_i}}{\sum_{k=1}^{N} e^{x_k}}$$
$$= -S(x)_j S(x)_i$$

$$\frac{\partial \frac{e^{x_i}}{\sum_{k=1}^{N} e^{x_k}}}{\partial x} = \begin{cases} S(x)_i - S(x)_j S(x)_i & if i = j \\ -S(x)_j S(x)_i & if i \neq j \end{cases}$$

Since $-S(x)_j S(x)_i$ term is common in both condition and taking indicator function $1_{i=j}$ and combining above conditions give :

$$\frac{\partial \frac{e^{x_i}}{\sum_{k=1}^{N} e^{x_k}}}{\partial x} = S(\boldsymbol{x})_i \mathbf{1}_{i=j} - S(\boldsymbol{x})_i S(\boldsymbol{x})_j$$

Assuming that $\frac{\partial S(\boldsymbol{x})}{\partial \boldsymbol{x}}$ is a $n \times n$ matrix, and for all $i, j \in \{1, \ldots, n\}$, the $(i, j)$ entry of the matrix is $\left(\frac{\partial S(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)_{i,j} = \frac{\partial S(\boldsymbol{x})_i}{\partial \boldsymbol{x}_j}$.

$$J_{\text{softmax}}(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial S(x)_1}{\partial x_1} & \frac{\partial S(x)_1}{\partial x_2} & \ldots & \frac{\partial S(x)_1}{\partial x_n} \\ \frac{\partial S(x)_2}{\partial x_1} & \frac{\partial S(x)_2}{\partial x_2} & \ldots & \frac{\partial S(x)_2}{\partial x_n} \\ \ldots & \ldots & \ldots & \ldots \\ \frac{\partial S(x)_n}{\partial x_1} & \frac{\partial S(x)_n}{\partial x_2} & \ldots & \frac{\partial S(x)_n}{\partial x_n} \end{bmatrix}$$

$$J_{\text{softmax}}(\boldsymbol{x}) = \begin{bmatrix} S(x)_1 - S(x)_1 S(x)_1 & -S(x)_1 S(x)_2 & \ldots & -S(x)_1 S(x)_n \\ -S(x)_2 S(x)_1 & -S(x)_2 S(x)_2 & \ldots & -S(x)_2 S(x)_n \\ \ldots & \ldots & \ldots & \ldots \\ S(x)_n - S(x)_n S(x)_1 & -S(x)_n S(x)_2 & \ldots & S(x)_n - S(x)_n S(x)_n \end{bmatrix}$$

$$J_{\text{softmax}}(\boldsymbol{x}) = diag(S(\boldsymbol{x})_i 1_{i=j}) - (S(\boldsymbol{x})_i S(\boldsymbol{x})_j)$$

Hence $J_{\text{softmax}}(\boldsymbol{x}) = \mathbf{Diag}(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^\top$, where $\boldsymbol{p} = S(\boldsymbol{x})$

**Question 3** (6)**.** Consider a 2-layer neural network $y : \mathbb{R}^D \to \mathbb{R}^K$ of the form :

$$y(x, \Theta, \sigma)_k = \sum_{j=1}^{M} \omega_{kj}^{(2)} \sigma \left( \sum_{i=1}^{D} \omega_{ji}^{(1)} x_i + \omega_{j0}^{(1)} \right) + \omega_{k0}^{(2)}$$

for $1 \leq k \leq K$, with parameters $\Theta = (\omega^{(1)}, \omega^{(2)})$ and logistic sigmoid activation function $\sigma$. Show that there exists an equivalent network of the same form, with parameters $\Theta' = (\tilde{\omega}^{(1)}, \tilde{\omega}^{(2)})$ and tanh activation function, such that $y(x, \Theta', \tanh) = y(x, \Theta, \sigma)$ for all $x \in \mathbb{R}^D$, and express $\Theta'$ as a function of $\Theta$.

**Solution :** Given that

$$y(x, \Theta, \sigma)_k = \sum_{j=1}^{M} \omega_{kj}^{(2)} \sigma \left( \sum_{i=1}^{D} \omega_{ji}^{(1)} x_i + \omega_{j0}^{(1)} \right) + \omega_{k0}^{(2)} \tag{3}$$

To represent sigmoid function as tanh : we know that $tanh(\frac{1}{2}x) = \frac{e^{\frac{x}{2}} - e^{\frac{-x}{2}}}{e^{\frac{x}{2}} + e^{\frac{-x}{2}}}$

$$tanh(\frac{x}{2}) + 1 = \frac{e^{\frac{x}{2}} - e^{\frac{-x}{2}}}{e^{\frac{x}{2}} + e^{\frac{-x}{2}}} + 1$$
$$= \frac{e^{\frac{x}{2}} - e^{\frac{-x}{2}} + e^{\frac{x}{2}} + e^{\frac{-x}{2}}}{e^{\frac{x}{2}} + e^{\frac{-x}{2}}}$$
$$= \frac{2e^{\frac{x}{2}}}{e^{\frac{x}{2}} + e^{\frac{-x}{2}}}$$

Multiplying numerator and denominator by $e^{\frac{-x}{2}}$

$$tanh(\frac{x}{2}) + 1 = \frac{2e^{\frac{x}{2}}e^{\frac{-x}{2}}}{(e^{\frac{x}{2}} + e^{\frac{-x}{2}})e^{\frac{-x}{2}}}$$
$$= \frac{2e^{0}}{e^{0} + e^{-x}}$$
$$= \frac{2}{1 + e^{-x}}$$
$$\frac{1}{2}tanh(\frac{x}{2}) + \frac{1}{2} = \frac{1}{1 + e^{-x}} = \sigma(x)$$

Substituting value of $\sigma$ in tanh in equation (2) :

$$y(x, \Theta, \sigma)_k = \sum_{j=1}^{M} \omega_{kj}^{(2)} \left( \frac{1}{2} \tanh \left( \frac{1}{2} (\sum_{i=1}^{D} \omega_{ji}^{(1)} x_i + \omega_{j0}^{(1)}) \right) + \frac{1}{2} \right) + \omega_{k0}^{(2)}$$
$$= \frac{1}{2} \sum_{j=1}^{M} \omega_{kj}^{(2)} \tanh \left( \frac{1}{2} (\sum_{i=1}^{D} \omega_{ji}^{(1)} x_i + \omega_{j0}^{(1)}) \right) + \frac{1}{2} \sum_{j=1}^{M} \omega_{kj}^{(2)} + \omega_{k0}^{(2)}$$

Equating above equation with $y(x, \Theta', \tanh)_k$ such that $y(x, \Theta', \tanh) = y(x, \Theta, \sigma)$ and expressing $\Theta'$ as a function of $\Theta$, we can write it as :

$$\tilde{\omega}_{kj}^{(2)} = \frac{1}{2} \omega_{kj}^{(2)}$$
$$\tilde{\omega}_{kj}^{(1)} = \frac{1}{2} \omega_{kj}^{(1)}$$
$$\tilde{\omega}_{j0}^{(1)} = \frac{1}{2} \omega_{j0}^{(1)}$$
$$\tilde{\omega}_{k0}^{(2)} = \frac{1}{2} \sum_{j=1}^{M} \omega_{kj}^{(2)} + \omega_{k0}^{(2)}$$

**Question 4** (5-5-6)**.** Consider a convolutional neural network. Assume the input is a colorful image of size $128 \times 128$ in the RGB representation. The first layer convolves 32 $8 \times 8$ kernels with the input, using a stride of 2 and a zero-padding of 3 (three zeros on each side). The second layer downsamples the output of the first layer with a $2 \times 2$ non-overlapping max pooling. The third layer convolves 64 $3 \times 3$ kernels with a stride of 1 and a zero-padding of size 1 on each border.

4.1 What is the dimensionality of the output of the last layer, i.e. the number of scalars it contains ?
**Solution :** Dimensionality of the output of the last layer : $64 \times 32 \times 32$ and number of scalars are 65536.

4.2 Not including the biases, how many parameters are needed for the last layer ?
**Solution :** Parameters needed for the last year - $64 \times 3 \times 3 \times 32 = 18432$

4.3 Compute the *full*, *valid*, and *same* convolution (with kernel flipping) for the following 1D matrices : $\big[1, 2, 3, 4\big] * \big[2, 0, 1\big]$
**Solution :**Full convolution :

$$[0,0,1,2,3,4,0,0] * [2,0,1] = [(2*0+0*0+1*1),(0*2+1*0+2*1),(1*2+2*0+3*1),$$

$$(2*2+3*0+4*1),(3*2+4*0+0*1),(4*2+0*0+0*1)]$$

$$= \mathbf{[1,\ 2,\ 5,\ 8,\ 6,\ 8]}$$

Valid convolution :$[1,2,3,4] * [2,0,1] = [(1*2+2*0+3*1),(2*2+3*0+4*1)] = \mathbf{[5,\ 8]}$
Full convolution : $[0,1,2,3,4,0] * [2,0,1] = [(0*2+1*0+2*1),(1*2+2*0+3*1),(2*2+3*0+4*1),(3*2+4*0+0*1)] = \mathbf{[2,\ 5,\ 8,\ 6]}$

**Question 5** (2-5-6-2-5-6)**.** Let us use the notation $*$ and $\tilde{*}$ to denote the valid and full convolution operator **without kernel flipping**, respectively. The operations are defined as

$$\text{valid} : (\boldsymbol{x} * \boldsymbol{w})_n = \sum_{j=1}^{k} x_{n+j-1} w_j \tag{4}$$

$$\text{full} : (\boldsymbol{x} \,\tilde{*}\, \boldsymbol{w})_n = \sum_{j=1}^{k} x_{n+j-k} w_j \ , \tag{5}$$

where $k$ is the size of the kernel $\boldsymbol{w}$. As a convention, the value of a vector indexed "out-of-bound" is zero, e.g. if $\boldsymbol{x} \in \mathbb{R}^d$, then $x_i = 0$ for $i < 1$ and $i > d$. We define the flip operator which reverse the ordering of the components of a vector, i.e. $\text{flip}(\boldsymbol{w})_j = w_{k-j+1}$.

Consider a convolutional network with 1-D input $\boldsymbol{x} \in \mathbb{R}^d$. Its first and second convolutional layers have kernel $\boldsymbol{w}^1 \in \mathbb{R}^{k_1}$ and $\boldsymbol{w}^2 \in \mathbb{R}^{k_2}$, respectively. Assume $k_1 < d$ and $k_2 < d$. The network is specified as follows :

$$\boldsymbol{a}^1 \leftarrow \boldsymbol{x} * \boldsymbol{w}^1 \text{ (valid convolution)} \tag{6}$$
$$\boldsymbol{h}^1 \leftarrow \text{ReLU}(\boldsymbol{a}^1) \tag{7}$$
$$\boldsymbol{a}^2 \leftarrow \boldsymbol{h}^1 * \boldsymbol{w}^2 \text{ (valid convolution)} \tag{8}$$
$$\boldsymbol{h}^2 \leftarrow \text{ReLU}(\boldsymbol{a}^2) \tag{9}$$
$$\ldots \tag{10}$$
$$L \leftarrow \ldots \tag{11}$$

where $L$ is the loss.

5.1 What is the dimensionality of $\boldsymbol{a}^2$ ? Denote it by $|\boldsymbol{a}^2|$.
**Solution :** Dimensionality of $a^1 = (d - k_1 + 1)$.

$$\text{Dimentionality of } \boldsymbol{a}^2 = |\boldsymbol{a}^2| = (d - k_1 + 1) - k_2 + 1 = d - k_1 - k_2 + 2$$

5.2 Derive $\frac{\partial a_i^2}{\partial h_n^1}$. Answer for all $i \in \{1, ..., |\boldsymbol{a}^2|\}$, given a particular $n$.
**Solution :** $a_i^2 = (h^1 * w^2)_i = \sum_{j=1}^{k_2} h_{i+j-1}^1 w_j^2$ where $i \in \{1, ..., |\boldsymbol{a}^2|\}$

$$\frac{\partial a_i}{\partial h_{i+j-1}^1} = \omega_j^2,\ i \in \{1, ..., |\boldsymbol{a}^2|\},\ j \in \{1, ..., k_2\}$$

Replacing $n = i + j - 1$ where $j = n - i + 1$.

$$\frac{\partial a_i}{\partial h_n^1} = \frac{\partial}{\partial h_n^1} \sum_{j=1}^{k_2} h_{i+j-1}^1 \omega_j^2 = \omega_j^2 = \omega_{n-i+1}^2$$

As $j \in \{1, ..., k_2\}$, which implies $1 \leq n - i + 1 \leq k_2$. Hence $\frac{\partial a_i}{\partial h_n^1}$ is zero everywhere except $n - k_2 + 1 \leq i \leq n$.

$$\frac{\partial a_i}{\partial h_n^1} = \omega_{n-i+1}^2$$

5.3 Show that $\nabla_{\boldsymbol{h}^1} L = \nabla_{\boldsymbol{a}^2} L \ \tilde{*}\ \text{flip}(\boldsymbol{w}^2)$ (full convolution). Start with

$$(\nabla_{\boldsymbol{h}^1} L)_n = \sum_{i=1}^{|\boldsymbol{a}^2|} (\nabla_{\boldsymbol{a}^2} L)_i \frac{\partial a_i^2}{\partial h_n^1}$$

**Solution :**

$$(\nabla_{\boldsymbol{h}^1} L)_n = \sum_{i=1}^{|\boldsymbol{a}^2|} (\nabla_{\boldsymbol{a}^2} L)_i \frac{\partial a_i^2}{\partial h_n^1}$$

Substituting $\frac{\partial a_i}{\partial h_n^1}$ in above equation.

$$(\nabla_{\boldsymbol{h}^1} L)_n = \sum_{i=1}^{|\boldsymbol{a}^2|} (\nabla_{\boldsymbol{a}^2} L)_i \omega_{n-i+1}^2$$

If $n - k_2 + 1$ does not belong to $\{1, ..., k_2\}$ then $\frac{\partial a_i^2}{\partial h_n^1} = 0$. Hence for $\frac{\partial a_i^2}{\partial h_n^1}$ to be non zero we need $i \in\ [n - k_2 + 1, n]$, and changing bounds :

$$(\nabla_{\boldsymbol{h}^1} L)_n = \sum_{i=1}^{|\boldsymbol{a}^2|} (\nabla_{\boldsymbol{a}^2} L)_i \frac{\partial a_i^2}{\partial h_n^1} = \sum_{i=n-k_2+1}^{n} (\nabla_{\boldsymbol{a}^2} L)_i \frac{\partial a_i^2}{\partial h_n^1}$$

Reordering the sum in above equation gives $j \in \{1, ..., k_2\}$. As given $\text{flip}(\boldsymbol{w})_j = \omega_{k-j+1}$, replacing $\omega_j$ :

$$(\nabla_{\boldsymbol{h}^1} L)_n = \sum_{j=1}^{k_2} (\nabla_{\boldsymbol{a}^2} L)_{n-j+1} \text{flip}(\boldsymbol{w})_{k_2-j+1}$$

Performing change of variable, $l = k_2 - j + 1$ and $l \in \{1, ..., k_2\}$

$$(\nabla_{\boldsymbol{h}^1} L)_n = \sum_{l=1}^{k_2} (\nabla_{\boldsymbol{a}^2} L)_i \frac{\partial a_i^2}{\partial h_n^1} = \sum_{l=1}^{k_2} (\nabla_{\boldsymbol{a}^2} L)_{n+l-k_2} \text{flip}(\boldsymbol{w})_l$$

Above equation looks similar to full convolution given in question i.e full :$(\boldsymbol{x} \tilde{*} \boldsymbol{w})_n = \sum_{j=1}^{k} x_{n+j-k} w_j$

$$(\nabla_{\boldsymbol{h}^1} L)_n = \nabla_{\boldsymbol{a}^2} L \tilde{*} \text{flip}(\boldsymbol{w}^2) \text{ (Full convolution)}$$

For the following, assume the convolutions in equations (6) and (8) are **full instead of valid**.

5.4 What is the dimensionality of $\boldsymbol{a}^2$ ? Denote it by $|\boldsymbol{a}^2|$.

**Solution :**

Dimensionality of $\boldsymbol{a}^1$ : $d + k_1 - 1$

Dimensionality of $\boldsymbol{a}^2$ : $d + k_1 + k_2 - 1$

5.5 Derive $\frac{\partial a_i^2}{\partial h_n^1}$. Answer for all $i \in \{1, ..., |\boldsymbol{a}^2|\}$, given a particular $n$.

**Solution** $a_i^2 = (h^1 * w^2)_i = \sum_{j=1}^{k_2} h_{i+j-k_2}^1 w_j^2$ where $i \in \{1, ..., |\boldsymbol{a}^2|\}$

$$\frac{\partial a_i}{\partial h_{i+j-k_2}^1} = \omega_j^2,\ i \in \{1, ..., |\boldsymbol{a}^2|\},\ j \in \{1, ..., k_2\}$$

Replacing $n = i + j - k_2$ where $j = n - i + k_2$.

$$\frac{\partial a_i}{\partial h_n^1} = \frac{\partial}{\partial h_n^1} \sum_{j=1}^{k_2} h_{i+j-k_2}^1 \omega_j^2 = \omega_j^2 = \omega_{n-i+k_2}^2$$

As $j \in \{1, ..., k_2\}$, which implies $1 \leq n - i + k_2 \leq k_2$. Hence $\frac{\partial a_i}{\partial h_n^1}$ is zero everywhere except $n \leq i \leq n + k_2 - 1$.

$$\frac{\partial a_i}{\partial h_n^1} = \omega_{n-i+k_2}^2$$

5.6 Show that $\nabla_{\boldsymbol{h}^1} L = \nabla_{\boldsymbol{a}^2} L * \text{flip}(\boldsymbol{w}^2)$ (valid convolution). Start with

$$(\nabla_{\boldsymbol{h}^1} L)_n = \sum_{i=1}^{|\boldsymbol{a}^2|} (\nabla_{\boldsymbol{a}^2} L)_i \frac{\partial a_i^2}{\partial h_n^1}$$

**Solution :**

$$(\nabla_{\boldsymbol{h}^1} L)_n = \sum_{i=1}^{|\boldsymbol{a}^2|} (\nabla_{\boldsymbol{a}^2} L)_i \frac{\partial a_i^2}{\partial h_n^1}$$

Substituting $\frac{\partial a_i}{\partial h_n^1}$ in above equation.

$$(\nabla_{\boldsymbol{h}^1} L)_n = \sum_{i=1}^{|\boldsymbol{a}^2|} (\nabla_{\boldsymbol{a}^2} L)_i \omega_{n-i+k_2}^2$$

If $n - i + k_2$ does not belong to $\{1, ..., k_2\}$ then $\frac{\partial a_i^2}{\partial h_n^1} = 0$. Hence for $\frac{\partial a_i^2}{\partial h_n^1}$ to be non zero we need $i \in [n, n + k_2 - 1]$, and changing bounds :

$$(\nabla_{\boldsymbol{h}^1} L)_n = \sum_{i=1}^{|\boldsymbol{a}^2|} (\nabla_{\boldsymbol{a}^2} L)_i \frac{\partial a_i^2}{\partial h_n^1} = \sum_{i=n}^{n+k_2-1} (\nabla_{\boldsymbol{a}^2} L)_i \frac{\partial a_i^2}{\partial h_n^1}$$

Reordering the sum in above equation gives,$j = n - i + k_2$ and $j \in \{1, ..., k_2\}$. As given $\text{flip}(\boldsymbol{w})_j = \omega_{k-j+1}$, replacing $\omega_j$ :

$$(\nabla_{\boldsymbol{h}^1} L)_n = \sum_{j=1}^{k_2} (\nabla_{\boldsymbol{a}^2} L)_{n-j+k_2} \text{flip}(\boldsymbol{w})_{k_2-j+1}$$

Performing change of variable, $l = k_2 - j + 1$ and $l \in \{1, ..., k_2\}$

$$(\nabla_{\boldsymbol{h}^1} L)_n = \sum_{l=1}^{k_2} (\nabla_{\boldsymbol{a}^2} L)_{n-j+k_2} \frac{\partial a_i^2}{\partial h_n^1} = \sum_{l=1}^{k_2} (\nabla_{\boldsymbol{a}^2} L)_{n+l-1} \text{flip}(\boldsymbol{w})_l$$

Above equation looks similar to valid convolution given in question i.e full :$(\boldsymbol{x} \tilde{*} \boldsymbol{w})_n = \sum_{j=1}^{k} x_{n+j-1} w_j$

$$(\nabla_{\boldsymbol{h}^1} L)_n = \nabla_{\boldsymbol{a}^2} L \tilde{*} \text{flip}(\boldsymbol{w}^2) \text{ (Valid convolution)}$$