Due Date : February 16th, 2019

Instructions

- *For all questions, show your work!*
- *Use a document preparation system such as LaTeX.*
- *Submit your answers electronically via the course studium page, and via Gradescope.*

Question 1. Using the following definition of the derivative and the definition of the Heaviside step function :

$$\frac{d}{dx}f(x) = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon} \qquad H(x) = \begin{cases} 1 & \text{if } x > 0 \\ \frac{1}{2} & \text{if } x = 0 \\ 0 & \text{if } x < 0 \end{cases}$$

1. Show that the derivative of the rectified linear unit $g(x) = \max\{0, x\}$, **wherever it exists**, is equal to the Heaviside step function.
2. Give two alternative definitions of $g(x)$ using $H(x)$.
3. Show that $H(x)$ can be well approximated by the sigmoid function $\sigma(x) = \frac{1}{1+e^{-kx}}$ asymptotically (i.e for large k), where k is a parameter.

*4. Although the Heaviside step function is not differentiable, we can define its **distributional derivative**. For a function F, consider the functional $F[\phi] = \int_{\mathbb{R}} F(x)\phi(x)dx$, where ϕ is a smooth function (infinitely differentiable) with compact support ($\phi(x) = 0$ whenever $|x| \geq A$, for some $A > 0$).

Show that whenever F is differentiable, $F'[\phi] = -\int_{\mathbb{R}} F(x)\phi'(x)dx$. Using this formula as a definition in the case of non-differentiable functions, show that $H'[\phi] = \phi(0)$. ($\delta[\phi] \doteq \phi(0)$ is known as the Dirac delta function.)

Answer 1.

1. (a) if $x > 0$, then $x + \epsilon > 0 (\epsilon \to 0)$, such that :

$$\begin{aligned} \frac{d}{dx}g(x) &= \lim_{\epsilon \to 0} \frac{g(x+\epsilon) - g(x)}{\epsilon} \\ &= \lim_{\epsilon \to 0} \frac{\max\{0, x+\epsilon\} - \max\{0, x\}}{\epsilon} \\ &= \lim_{\epsilon \to 0} \frac{x+\epsilon - x}{\epsilon} \\ &= \lim_{\epsilon \to 0} \frac{\epsilon}{\epsilon} \\ &= \lim_{\epsilon \to 0} 1 \\ &= 1 \\ H(x) &= 1 \end{aligned}$$

(b) if $x < 0$, then $x + \epsilon < 0 (\epsilon \to 0)$, such that :

$$\begin{aligned}
\frac{d}{dx}g(x) &= \lim_{\epsilon \to 0} \frac{g(x+\epsilon) - g(x)}{\epsilon} \\
&= \lim_{\epsilon \to 0} \frac{\max\{0, x+\epsilon\} - \max\{0, x\}}{\epsilon} \\
&= \lim_{\epsilon \to 0} \frac{0-0}{\epsilon} \\
&= \lim_{\epsilon \to 0} \frac{0}{\epsilon} \\
&= 0 \\
H(x) &= 0
\end{aligned}$$

(c) if $x = 0$, then :

$$\lim_{\epsilon \to 0^+} \frac{g(x+\epsilon) - g(x)}{\epsilon} = \lim_{\epsilon \to 0^+} \frac{\max\{0, x+\epsilon\} - \max\{0, x\}}{\epsilon} = \lim_{\epsilon \to 0^+} \frac{\epsilon - 0}{\epsilon} = 1$$

$$\lim_{\epsilon \to 0^-} \frac{g(x+\epsilon) - g(x)}{\epsilon} = \lim_{\epsilon \to 0^-} \frac{\max\{0, x+\epsilon\} - \max\{0, x\}}{\epsilon} = \lim_{\epsilon \to 0^+} \frac{0 - 0}{\epsilon} = 0$$

$$\lim_{\epsilon \to 0^+} \frac{g(x+\epsilon) - g(x)}{\epsilon} \neq \lim_{\epsilon \to 0^-} \frac{g(x+\epsilon) - g(x)}{\epsilon} \Rightarrow \lim_{\epsilon \to 0} \frac{g(x+\epsilon) - g(x)}{\epsilon} \text{ does not exist.}$$

Therefore, wherever the derivative of $g(x) = \max\{0, x\}$ exists, $g(x) = H(x)$.

2.

$$g(x) = \max\{0, x\} = xH(x)$$

or

$$g(x) = \max\{0, x\} = \int_{-\infty}^{x} H(x)dx$$

3.

$$\begin{aligned}
\lim_{k \to \infty} \frac{1}{1 + e^{-kx}} &= \lim_{k \to \infty} \frac{1}{1 + \frac{1}{e^{kx}}} \\
&= \lim_{k \to \infty} \frac{e^{kx}}{1 + e^{kx}} \\
&= \lim_{k \to \infty} \left(1 - \frac{1}{1 + e^{kx}}\right) \\
&= 1 - \lim_{k \to \infty} \frac{1}{1 + e^{kx}} \\
&= \begin{cases} 1 & \text{if } x > 0 \\ \frac{1}{2} & \text{if } x = 0 \\ 0 & \text{if } x < 0 \end{cases} \\
&= H(x)
\end{aligned}$$

*4. Given $F[\phi] = \int_{\mathbb{R}} F(x)\phi(x)dx$, where ϕ is infinitely differentiable, and $\phi(x) = 0$ whenever $|x| \geq A$ for some $A > 0$. If F is differentiable :

$$\begin{aligned}\int_{\mathbb{R}} (F(x)\phi(x))' dx &= F(x)\phi(x)|_{-\infty}^{+\infty} \\ &= F(+\infty)\phi(+\infty) - F(-\infty)\phi(-\infty) \\ &= 0 - 0 \\ &= 0\end{aligned}$$

Meanwhile,

$$\begin{aligned}\int_{\mathbb{R}} (F(x)\phi(x))' dx &= \int_{\mathbb{R}} (F'(x)\phi(x) + F(x)\phi'(x))dx \\ &= \int_{\mathbb{R}} F'(x)\phi(x)dx + \int_{\mathbb{R}} F(x)\phi'(x)dx\end{aligned}$$

Therefore,

$$\int_{\mathbb{R}} F'(x)\phi(x)dx = -\int_{\mathbb{R}} F(x)\phi'(x)dx$$

By the definition of $F[\phi]$, we have,

$$F'[\phi] = \int_{\mathbb{R}} F'(x)\phi(x)dx = -\int_{\mathbb{R}} F(x)\phi'(x)dx$$

Let $\epsilon > 0$,

$$\begin{aligned}H'[\phi] &= -\int_{\mathbb{R}} H(x)\phi'(x)dx \\ &= -\lim_{\epsilon\to 0}\Big(\int_{-\infty}^{-\epsilon} H(x)\phi'(x)dx + \int_{-\epsilon}^{\epsilon} H(x)\phi'(x)dx + \int_{\epsilon}^{+\infty} H(x)\phi'(x)dx\Big) \\ &= -\Big(\lim_{\epsilon\to 0}\int_{-\infty}^{-\epsilon} H(x)d\phi(x) + \lim_{\epsilon\to 0}\int_{-\epsilon}^{\epsilon} H(x)d\phi(x) + \lim_{\epsilon\to 0}\int_{\epsilon}^{+\infty} H(x)d\phi(x)\Big) \\ &= -\Big(\lim_{\epsilon\to 0}\int_{-\infty}^{-\epsilon} 0 d\phi(x) + \frac{1}{2}\lim_{\epsilon\to 0}(\phi(\epsilon) - \phi(-\epsilon)) + \lim_{\epsilon\to 0}\int_{\epsilon}^{+\infty} 1 d\phi(x)\Big) \\ &= -\big(0 + \frac{1}{2}(\phi(0) - \phi(0)) + (\phi(\infty) - \phi(0))\big) \\ &= \phi(0)\end{aligned}$$

Question 2. Let x be an n-dimentional vector. Recall the softmax function : $S : \boldsymbol{x} \in \mathbb{R}^n \mapsto S(\boldsymbol{x}) \in \mathbb{R}^n$ such that $S(\boldsymbol{x})_i = \frac{e^{\boldsymbol{x}_i}}{\sum_j e^{\boldsymbol{x}_j}}$; the diagonal function : $\text{diag}(\boldsymbol{x})_{ij} = \boldsymbol{x}_i$ if $i = j$ and $\text{diag}(\boldsymbol{x})_{ij} = 0$ if $i \neq j$; and the Kronecker delta function : $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$.

1. Show that the derivative of the softmax function is $\frac{dS(\boldsymbol{x})_i}{d\boldsymbol{x}_j} = S(\boldsymbol{x})_i\,(\delta_{ij} - S(\boldsymbol{x})_j)$.
2. Express the Jacobian matrix $\frac{\partial S(\boldsymbol{x})}{\partial \boldsymbol{x}}$ using matrix-vector notation. Use $\text{diag}(\cdot)$.

3. Compute the Jacobian of the sigmoid function $\sigma(\boldsymbol{x}) = 1/(1+e^{-\boldsymbol{x}})$.
4. Let $\boldsymbol{y}$ and $\boldsymbol{x}$ be $n-$dimensional vectors related by $\boldsymbol{y} = f(\boldsymbol{x})$, L be an unspecified differentiable loss function. According to the chain rule of calculus, $\nabla_{\boldsymbol{x}} L = (\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}})^\top \nabla_{\boldsymbol{y}} L$, which takes up $\mathcal{O}(n^2)$ computational time in general. Show that if $f(\boldsymbol{x}) = \sigma(\boldsymbol{x})$ or $f(\boldsymbol{x}) = S(\boldsymbol{x})$, the above matrix-vector multiplication can be simplified to a $\mathcal{O}(n)$ operation.

Answer 2.

1.

$$
\begin{aligned}
\frac{dS(\boldsymbol{x})_i}{d\boldsymbol{x}_j} &= \frac{d\frac{e^{\boldsymbol{x}_i}}{\sum_j e^{\boldsymbol{x}_j}}}{d\boldsymbol{x}_j} \\
&= \begin{cases} \frac{e^{\boldsymbol{x}_i}\ \sum_j e^{\boldsymbol{x}_j} - e^{\boldsymbol{x}_i} e^{\boldsymbol{x}_i}}{(\sum_j e^{\boldsymbol{x}_j})^2} & \text{if } i = j, \big(\text{ recall : } \frac{d\frac{f}{g}}{x} = \frac{\frac{df}{x}g - \frac{dg}{x}f}{g^2}\big) \\ e^{\boldsymbol{x}_i}\frac{-e^{\boldsymbol{x}_j}}{(\sum_j e^{\boldsymbol{x}_j})^2} & \text{if } i \neq j, \big(\text{ recall : } \frac{d\frac{c}{f}}{x} = -c\frac{\frac{df}{x}}{f^2}\big) \end{cases} \\
&= \begin{cases} S(\boldsymbol{x})_i - S(\boldsymbol{x})_i^2 & \text{if } i = j \\ -S(\boldsymbol{x})_i S(\boldsymbol{x})_j & \text{if } i \neq j \end{cases} \\
&= \begin{cases} S(\boldsymbol{x})_i(1 - S(\boldsymbol{x})_j) & \text{if } i = j \\ S(\boldsymbol{x})_i(0 - S(\boldsymbol{x})_j) & \text{if } i \neq j \end{cases} \\
&= S(\boldsymbol{x})_i(\delta_{ij} - S(\boldsymbol{x})_j)
\end{aligned}
$$

2. For this question, Jacobian matrix $\frac{\partial S(\boldsymbol{x})}{\partial \boldsymbol{x}}$ is a $n \times n$ matrix, where the ith row, jth column element :

$$
\begin{aligned}
\frac{\partial S(\boldsymbol{x})}{\partial \boldsymbol{x}}_{i,j} &= \frac{dS(\boldsymbol{x})_i}{d\boldsymbol{x}_j} \\
&= S(\boldsymbol{x})_i(\delta_{ij} - S(\boldsymbol{x})_j) \\
&= \delta_{ij} S(\boldsymbol{x})_i - S(\boldsymbol{x})_i S(\boldsymbol{x})_j \\
&= \mathrm{diag}(S(\boldsymbol{x}))_{ij} - S(\boldsymbol{x})_i S(\boldsymbol{x})_j
\end{aligned}
$$

By default, $S(\boldsymbol{x})$ is a column vector ; therefore,

$$
\frac{\partial S(\boldsymbol{x})}{\partial \boldsymbol{x}} = \mathrm{diag}(S(\boldsymbol{x})) - S(\boldsymbol{x})S(\boldsymbol{x})^\top
$$

3.

$$
\begin{aligned}
\frac{\partial \sigma(\boldsymbol{x})}{\partial \boldsymbol{x}}_{i,j} &= \frac{d\sigma(\boldsymbol{x})_i}{d\boldsymbol{x}_j} \\
&= \frac{d\sigma(\boldsymbol{x}_i)}{d\boldsymbol{x}_j} \\
&= \begin{cases} \sigma(\boldsymbol{x}_i)(1 - \sigma(\boldsymbol{x}_i)), & \text{if } i = j, \text{ recall : } \sigma'(x) = \sigma(x)(1 - \sigma(x)) \\ 0, & \text{if } i \neq j \end{cases} \\
&= \begin{cases} \sigma(\boldsymbol{x})_i(1 - \sigma(\boldsymbol{x})_i), & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}
\end{aligned}
$$

Therefore,

$$\frac{\partial \sigma(\boldsymbol{x})}{\partial \boldsymbol{x}} = \text{diag}\big(\sigma(\boldsymbol{x})(1-\sigma(\boldsymbol{x}))\big)$$

Justification of $\sigma'(x) = \sigma(x)(1-\sigma(x))$:

$$\begin{aligned}
\sigma'(x) &= \frac{d\sigma(x)}{dx} \\
&= \frac{d\,\frac{1}{1+e^{-x}}}{dx} \\
&= -(\frac{1}{1+e^{-x}})^2 \frac{d(1+e^{-x})}{dx} \\
&= \frac{1}{(1+e^{-x})^2} e^{-x} \\
&= \frac{1}{1+e^{-x}} \frac{e^{-x}}{1+e^{-x}} \\
&= \frac{1}{1+e^{-x}} \frac{1+e^{-x}-1}{1+e^{-x}} \\
&= \frac{1}{1+e^{-x}} \Big(1 - \frac{1}{1+e^{-x}}\Big) \\
&= \sigma(x)(1-\sigma(x))
\end{aligned}$$

4. Denote : $\odot$ represents element-wise matrix(or column vector) multiplication, and $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ represents the inner-product of two column vectors(or matrices).

1). For the case $\boldsymbol{y} = f(\boldsymbol{x}) = \sigma(\boldsymbol{x})$,

$$\begin{aligned}
\nabla_{\boldsymbol{x}} L = (\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}})^\top \nabla_{\boldsymbol{y}} L &= (\frac{\partial \sigma(\boldsymbol{x})}{\partial \boldsymbol{x}})^\top \nabla_{\boldsymbol{y}} L \\
&= \Big(\text{diag}\big(\sigma(\boldsymbol{x})(1-\sigma(\boldsymbol{x}))\big)\Big)^\top \nabla_{\boldsymbol{y}} L \\
&= \text{diag}\big(\sigma(\boldsymbol{x})(1-\sigma(\boldsymbol{x}))\big) \nabla_{\boldsymbol{y}} L \\
&= \sigma(\boldsymbol{x}) \odot (1-\sigma(\boldsymbol{x})) \odot \nabla_{\boldsymbol{y}} L
\end{aligned}$$

which means, the computation of $\nabla_{\boldsymbol{x}} L$ can be decomposed to one vector minus calculation, and three element-wise vector multiplications. All these calculations can be done in $\mathcal{O}(n)$ time complexity ; therefore, the whole time complexity is $\mathcal{O}(4n + C) = \mathcal{O}(n)$.

2). For the case $\boldsymbol{y} = f(\boldsymbol{x}) = S(\boldsymbol{x})$,

$$\begin{aligned}
\nabla_{\boldsymbol{x}} L = (\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}})^\top \nabla_{\boldsymbol{y}} L &= (\frac{\partial S(\boldsymbol{x})}{\partial \boldsymbol{x}})^\top \nabla_{\boldsymbol{y}} L \\
&= \big(\text{diag}(S(\boldsymbol{x})) - S(\boldsymbol{x})S(\boldsymbol{x})^\top\big)^\top \nabla_{\boldsymbol{y}} L \\
&= \big(\text{diag}(S(\boldsymbol{x})) - S(\boldsymbol{x})S(\boldsymbol{x})^\top\big) \nabla_{\boldsymbol{y}} L \\
&= \text{diag}(S(\boldsymbol{x})) \nabla_{\boldsymbol{y}} L - S(\boldsymbol{x})S(\boldsymbol{x})^\top \nabla_{\boldsymbol{y}} L \\
&= S(\boldsymbol{x}) \odot \nabla_{\boldsymbol{y}} L - S(\boldsymbol{x}) \langle (S(\boldsymbol{x})^\top, \nabla_{\boldsymbol{y}} L \rangle
\end{aligned}$$

which means, the computation of $\nabla_{\boldsymbol{x}} L$ can be decomposed to one inner-product, one matrix and scalar multiplication, one element-wise vector multiplication, and a vector minor operation.

All of this calculations can be done with the time complexity of $\mathcal{O}(n)$, Therefore, the whole calculation has the the complexity of $\mathcal{O}(4n + C) = \mathcal{O}(n)$.

Question 3. Recall the definition of the softmax function : $S(\boldsymbol{x})_i = e^{\boldsymbol{x}_i} / \sum_j e^{\boldsymbol{x}_j}$.

1. Show that softmax is translation-invariant, that is : $S(\boldsymbol{x}+c) = S(\boldsymbol{x})$, where c is a scalar constant.
2. Show that softmax is not invariant under scalar multiplication. Let $S_c(\boldsymbol{x}) = S(c\boldsymbol{x})$ where $c \geq 0$. What are the effects of taking c to be 0 and arbitrarily large ?
3. Let $\boldsymbol{x}$ be a 2-dimentional 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 $\boldsymbol{x}$.
4. Let $\boldsymbol{x}$ be a K-dimentional 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 $i \in \{1, ..., K-1\}$.

Answer 3.

1. Recall : $(\boldsymbol{x} + c)_i = \boldsymbol{x}_i + c$.

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

which means that each element in vectors $S(\boldsymbol{x}+c)$ is equal to the element in $S(\boldsymbol{x})$ with the same index. That is to say, the two vectors are same :

$$S(\boldsymbol{x} + c) = S(\boldsymbol{x})$$

2. Recall : $(c\boldsymbol{x})_i = c\boldsymbol{x}_i$.

To prove $S(\boldsymbol{x})$ is not invariant under scalar multiplication, we only need to provide an example where $S(c\boldsymbol{x}) \neq S(\boldsymbol{x})$, with $c \geq 0$.

Consider a 2- dimensional vector $\boldsymbol{x} = [0, \ln 3]^T$, and $c = 2$, $S(c\boldsymbol{x}) = S([0, 2\ln 3]^T) = [0.1, 0.9]^T$, whereas $S(\boldsymbol{x}) = S([0, \ln 3]^T) = [0.25, 0.75]^T$. Therefore :

$$S(c\boldsymbol{x}) \neq S(\boldsymbol{x})$$

That $S(\boldsymbol{x})$ is not invariant under scalar multiplication with $c \geq 0$ doesn't means $S(c\boldsymbol{x})$ has no chance to be equal to $S(\boldsymbol{x})$. If elements in a n-dimensional vector $\boldsymbol{x}$ are all equal, then : $S(c\boldsymbol{x}) = S(\boldsymbol{x})$, and

$$S(c\boldsymbol{x})_i = S(\boldsymbol{x})_i = \frac{1}{n}$$

.

When $c = 0$, $e^{c\boldsymbol{x}_i} = e^{\boldsymbol{x}_i} = e^0 = 1$,

$$S(c\boldsymbol{x})_i = \frac{1}{\sum_1^n 1} = \frac{1}{n}$$

meaning that all element of $S(c\boldsymbol{x})$ are equal. If the element value of $S(c\boldsymbol{x})$ reflects the probability of several events, then it means all events have the same probability.
Now let's consider the situation when c is arbitrarily large and not all elements of $\boldsymbol{x}$ are equal. Suppose the n-dimensional vector $\boldsymbol{x}$ has k$(0 \leq k \leq n-1)$ largest elements with the maximal values are all x^* and their indices forming a collection $\mathcal{K}$:

$$x^* = \boldsymbol{x}_{k,k\in\mathcal{K}} = max\{\boldsymbol{x}_i,\ 0 \leq i \leq n-1\}$$

then,

$$\begin{aligned}
\lim_{c\to\infty} S(c\boldsymbol{x})_i &= \lim_{c\to+\infty} \frac{e^{c\boldsymbol{x}_i}}{\sum_j e^{c\boldsymbol{x}_j}} \\
&= \lim_{c\to+\infty} \frac{e^{c\boldsymbol{x}_i}/e^{cx^*}}{\left(\sum_j e^{c\boldsymbol{x}_j}\right)/e^{cx^*}} \\
&= \lim_{c\to+\infty} \frac{e^{c(\boldsymbol{x}_i - x^*)}}{\sum_j e^{c(\boldsymbol{x}_j - x^*)}} \\
&= \begin{cases} \frac{1}{k} & \text{if } i \in \mathcal{K} \\ 0 & \text{if } i \notin \mathcal{K} \end{cases}
\end{aligned}$$

To conclude, if $c = 0$, all elements of $S(c\boldsymbol{x})_i$ are equal; if c is arbitrarily large, only the largest element(s) has(or equally share) the value 1, other elements have the value 0.

3. If $\boldsymbol{x}$ is a 2-dimensional vector, let scalar :

$$z = f(\boldsymbol{x}) = \boldsymbol{x}_0 - \boldsymbol{x}_1$$

$$\begin{aligned}
S(\boldsymbol{x})_0 &= \frac{e^{\boldsymbol{x}_0}}{e^{\boldsymbol{x}_0} + e^{\boldsymbol{x}_1}} = \frac{1}{1 + e^{-(\boldsymbol{x}_0 - \boldsymbol{x}_1)}} = \sigma(\boldsymbol{x}_0 - \boldsymbol{x}_1) = \sigma(z) \\
S(\boldsymbol{x})_1 &= \frac{e^{\boldsymbol{x}_1}}{e^{\boldsymbol{x}_0} + e^{\boldsymbol{x}_1}} = 1 - \frac{e^{\boldsymbol{x}_0}}{e^{\boldsymbol{x}_0} + e^{\boldsymbol{x}_1}} = 1 - S(\boldsymbol{x})_0 = 1 - \sigma(z)
\end{aligned}$$

Therefore,

$$S(\boldsymbol{x}) = [S(\boldsymbol{x})_0, S(\boldsymbol{x})_1]^T = [\sigma(z), 1 - \sigma(z)]^T$$

4. If $\boldsymbol{x}$ is a K-dimensional vector, let constant $c = -\boldsymbol{x}_0$, and :

$$y_i = \boldsymbol{x}_i - \boldsymbol{x}_0,\ \text{where } 1 \leq i \leq K-1$$

As function $S(\boldsymbol{x})$ is translation-invariant, that is :$S(\boldsymbol{x} + c) = S(\boldsymbol{x})$, we have :

$$\begin{aligned}
S(\boldsymbol{x}) &= S(\boldsymbol{x} + c) \\
&= S(\boldsymbol{x} - \boldsymbol{x}_0) \\
&= S([\boldsymbol{x}_0 - \boldsymbol{x}_0, \boldsymbol{x}_1 - \boldsymbol{x}_0, \cdots, \boldsymbol{x}_{K-1} - \boldsymbol{x}_0)]^T) \\
&= S([0, y_1, y_2, \cdots, y_{K-1})]^T)
\end{aligned}$$

Question 4. 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 σ. 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 Θ' as a function of Θ.

Answer 4. First, we show that $\sigma(\boldsymbol{x})$ is a function of $\tanh(\boldsymbol{x})$

$$\begin{aligned}
\sigma(\boldsymbol{x}) &= \frac{1}{1+e^{-\boldsymbol{x}}} \\
&= \frac{1}{1+e^{-\frac{1}{2}\boldsymbol{x}-\frac{1}{2}\boldsymbol{x}}} \\
&= \frac{1}{1+\frac{e^{-\frac{1}{2}\boldsymbol{x}}}{e^{\frac{1}{2}\boldsymbol{x}}}} \\
&= \frac{e^{\frac{1}{2}\boldsymbol{x}}}{e^{\frac{1}{2}\boldsymbol{x}}+e^{-\frac{1}{2}\boldsymbol{x}}} \\
&= \frac{1}{2}\left(\frac{2\ e^{\frac{1}{2}\boldsymbol{x}}}{e^{\frac{1}{2}\boldsymbol{x}}+e^{-\frac{1}{2}\boldsymbol{x}}}\right) \\
&= \frac{1}{2}\left(\frac{e^{\frac{1}{2}\boldsymbol{x}}-e^{-\frac{1}{2}\boldsymbol{x}}}{e^{\frac{1}{2}\boldsymbol{x}}+e^{-\frac{1}{2}\boldsymbol{x}}}+1\right) \\
&= \frac{1}{2}\left(\tanh(\frac{1}{2}\boldsymbol{x})+1\right)
\end{aligned}$$

Then, for the 2-layer neural network $y : \mathbb{R}^D \to \mathbb{R}^K$:

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

As $\Theta' = (\tilde{\omega}^{(1)}, \tilde{\omega}^{(2)})$ and $\Theta = (\omega^{(1)}, \omega^{(2)})$, let :

$$\tilde{\omega}^{(1)} = \frac{1}{2} \omega^{(1)}$$

and for $1 \leq k \leq K$,

$$\tilde{\omega}_{kj}^{(2)} = \begin{cases} \frac{1}{2}\omega_{kj}^{(2)} & \text{if } 1 \leq j \leq M \\ \omega_{kj}^{(2)} + \sum_{i=1}^{M} \frac{1}{2}\omega_{ki}^{(2)} & \text{if } j = 0 \end{cases}$$

$$\begin{aligned} y(x, \Theta, \sigma)_k &= \sum_{j=1}^{M} \tilde{\omega}_{kj}^{(2)} \tanh\left(\sum_{i=1}^{D} \tilde{\omega}_{ji}^{(1)} x_i + \tilde{\omega}_{j0}^{(1)}\right) + \tilde{\omega}_{k0}^{(2)} \\ &= y(x, \Theta', \tanh)_k \end{aligned}$$

Question 5. Given $N \in \mathbb{Z}^+$, we want to show that for any $f : \mathbb{R}^n \to \mathbb{R}^m$ and any sample set $\mathcal{S} \subset \mathbb{R}^n$ of size N, there is a set of parameters for a two-layer network such that the output $y(\boldsymbol{x})$ matches $f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{S}$. That is, we want to interpolate f with y on any finite set of samples $\mathcal{S}$.

1. Write the generic form of the function $y : \mathbb{R}^n \to \mathbb{R}^m$ defined by a 2-layer network with $N-1$ hidden units, with linear output and activation function ϕ, in terms of its weights and biases $(\boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)})$ and $(\boldsymbol{W}^{(2)}, \boldsymbol{b}^{(2)})$.
2. In what follows, we will restrict $\boldsymbol{W}^{(1)}$ to be $\boldsymbol{W}^{(1)} = [\boldsymbol{w}, \cdots, \boldsymbol{w}]^\top$ for some $\boldsymbol{w} \in \mathbb{R}^n$ (so the rows of $\boldsymbol{W}^{(1)}$ are all the same). Show that the interpolation problem on the sample set $\mathcal{S} = \{\boldsymbol{x}^{(1)}, \cdots \boldsymbol{x}^{(N)}\} \subset \mathbb{R}^n$ can be reduced to solving a matrix equation : $\boldsymbol{M}\tilde{\boldsymbol{W}}^{(2)} = \boldsymbol{F}$, where $\tilde{\boldsymbol{W}}^{(2)}$ and $\boldsymbol{F}$ are both $N \times m$, given by

$$\tilde{\boldsymbol{W}}^{(2)} = [\boldsymbol{W}^{(2)}, \boldsymbol{b}^{(2)}]^\top \qquad \boldsymbol{F} = [f(\boldsymbol{x}^{(1)}), \cdots, f(\boldsymbol{x}^{(N)})]^\top$$

   Express the $N \times N$ matrix $\boldsymbol{M}$ in terms of $\boldsymbol{w}$, $\boldsymbol{b}^{(1)}$, ϕ and $\boldsymbol{x}^{(i)}$.
3. *3. **Proof with Relu activation.** Assume $\boldsymbol{x}^{(i)}$ are all distinct. Choose $\boldsymbol{w}$ such that $\boldsymbol{w}^\top\boldsymbol{x}^{(i)}$ are also all distinct (Try to prove the existence of such a $\boldsymbol{w}$, although this is not required for the assignment - See Assignment 0). Set $\boldsymbol{b}_j^{(1)} = -\boldsymbol{w}^\top\boldsymbol{x}^{(j)} + \epsilon$, where $\epsilon > 0$. Find a value of ϵ such that $\boldsymbol{M}$ is triangular with non-zero diagonal elements. Conclude. (Hint : assume an ordering of $\boldsymbol{w}^\top\boldsymbol{x}^{(i)}$.)
4. *4. **Proof with sigmoid-like activations**. Assume ϕ is continuous, bounded, $\phi(-\infty) = 0$ and $\phi(0) > 0$. Decompose $\boldsymbol{w}$ as $\boldsymbol{w} = \lambda\boldsymbol{u}$. Set $\boldsymbol{b}_j^{(1)} = -\lambda\boldsymbol{u}^\top\boldsymbol{x}^{(j)}$. Fixing $\boldsymbol{u}$, show that $\lim_{\lambda\to+\infty} \boldsymbol{M}$ is triangular with non-zero diagonal elements. Conclude. (Note that doing so preserves the distinctness of $\boldsymbol{w}^\top\boldsymbol{x}^{(i)}$.)

Answer 5.

1. for $0 \leq k \leq m-1$,

$$y(\boldsymbol{x})_k = \sum_{j=0}^{N-1} \boldsymbol{W}_{kj}^{(2)}\ \phi\left(\sum_{i=0}^{n-1} \boldsymbol{W}_{ji}^{(1)}\boldsymbol{x}_i + \boldsymbol{b}_j^{(1)}\right) + \boldsymbol{b}_k^2$$

   or the matrix form :

$$y(\boldsymbol{x}) = \boldsymbol{W}^{(2)}\phi\big(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}\big) + \boldsymbol{b}^{(2)}$$

2. Consider the interpolation problem on the sample set $\mathcal{S} = \{\boldsymbol{x}^{(1)}, \cdots \boldsymbol{x}^{(N)}\} \subset \mathbb{R}^n$ with N samples, let :

$$\boldsymbol{X} = [\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(N)}]^\top$$

and

$$\boldsymbol{Y} = [y(\boldsymbol{x}^{(1)}), y(\boldsymbol{x}^{(2)}), \cdots, y(\boldsymbol{x}^{(N)})]^\top$$

That $y(\boldsymbol{x})$ matches $f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{S}$ means :

$$\boldsymbol{Y} = [f(\boldsymbol{x}^{(1)}), f(\boldsymbol{x}^{(2)}), \cdots, f(\boldsymbol{x}^{(N)})]^\top = \boldsymbol{F}$$

$$\begin{aligned}\boldsymbol{F} &= \phi\left(\boldsymbol{X}(\boldsymbol{W}^{(1)})^\top + [\underbrace{\boldsymbol{b}^{(1)}, \cdots, \boldsymbol{b}^{(1)}}_{N \text{ times}}]^\top\right)(\boldsymbol{W}^{(2)})^\top + [\underbrace{\boldsymbol{b}^{(2)}, \cdots, \boldsymbol{b}^{(2)}}_{N \text{ times}}]^\top \\ &= \left[\phi([\boldsymbol{X}, 1] \cdot [\boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)}]^\top), 1\right] \cdot \left[\boldsymbol{W}^{(2)}, \boldsymbol{b}^{(2)}\right]^\top\end{aligned}$$

As $\tilde{\boldsymbol{W}}^{(2)} = [\boldsymbol{W}^{(2)}, \boldsymbol{b}^{(2)}]^\top$ and $\boldsymbol{W}^{(1)} = [\boldsymbol{w}, \cdots, \boldsymbol{w}]^\top,\ \boldsymbol{w} \in \mathbb{R}^n$, Let :

$$\boldsymbol{M} = \left[\phi([\boldsymbol{X}, 1] \cdot [\boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)}]^\top), 1\right]$$

Which means for $0 \leq i,\ j \leq N-1$:

$$\boldsymbol{M}_{ij} = \begin{cases} \phi\big(\boldsymbol{w}^\top \boldsymbol{x}^{(i)} + \boldsymbol{b}_j^{(1)}\big) & \text{for } 0 \leq j < N-1 \\ 1 & \text{for } j = N-1 \end{cases}$$

Obviously, $\boldsymbol{M}$ is $N \times N$. So we have :

$$\boldsymbol{F} = \boldsymbol{M}\tilde{\boldsymbol{W}}^{(2)}$$

*3. As $\boldsymbol{x}^{(i)}$ are distinct, and $\boldsymbol{w}^\top \boldsymbol{x}^{(i)} \in \mathbb{R}$ are also distinct, we can permutate $\boldsymbol{x}^{(i)}$ by sorting $\boldsymbol{w}^\top \boldsymbol{x}^{(i)}$, such that for all $0 \leq j < i \leq N-1$:

$$\boldsymbol{w}^\top \boldsymbol{x}^{(j)} > \boldsymbol{w}^\top \boldsymbol{x}^{(i)}$$

Let $\boldsymbol{b}_j^{(1)} = -\boldsymbol{w}^\top \boldsymbol{x}^{(j)} + \epsilon$, where $\epsilon > 0$, we have :

$$\boldsymbol{M}_{ij} = \begin{cases} \phi\big(\boldsymbol{w}^\top \boldsymbol{x}^{(i)} - \boldsymbol{w}^\top \boldsymbol{x}^{(j)} + \epsilon\big) & \text{for } 0 \leq j < N-1 \\ 1 & \text{for } j = N-1 \end{cases}$$

As ϕ is Relu activation function,

$$\begin{aligned} &\phi\big(\boldsymbol{w}^\top \boldsymbol{x}^{(i)} - \boldsymbol{w}^\top \boldsymbol{x}^{(j)} + \epsilon\big) = 0 \\ \Longleftrightarrow &\boldsymbol{w}^\top \boldsymbol{x}^{(i)} - \boldsymbol{w}^\top \boldsymbol{x}^{(j)} + \epsilon \leq 0 \\ \Longleftrightarrow &\epsilon \leq \boldsymbol{w}^\top \boldsymbol{x}^{(j)} - \boldsymbol{w}^\top \boldsymbol{x}^{(i)} \end{aligned}$$

Let,

$$0 < \epsilon \leq \min\{\boldsymbol{w}^\top \boldsymbol{x}^{(j)} - \boldsymbol{w}^\top \boldsymbol{x}^{(i)} \mid 0 \leq j < i \leq N-1\}$$

for $0 \leq i,\ j \leq N-1$, we have :

$$\boldsymbol{M}_{ij} = \begin{cases} 0 & 0 \leq j < i \leq N-1 \\ \epsilon & 0 \leq i = j < N-1 \\ \boldsymbol{w}^\top \boldsymbol{x}^{(i)} - \boldsymbol{w}^\top \boldsymbol{x}^{(j)} + \epsilon & 0 \leq i < j < N-1 \\ 1 & j = N-1 \end{cases}$$

indicating $\boldsymbol{M}$ is a triangular matrix with non-zero diagonal elements.

This proves that, for any $f : \mathbb{R}^n \to \mathbb{R}^m$ and any finite sample set $\mathcal{S} \subset \mathbb{R}^n$ of size N, there always exists a set of parameters for a two-layer network with $N-1$ neurons in hidden layer(with $\boldsymbol{W}^{(1)}$ specially chosen) and a ReLU activation function, such that the output $y(\boldsymbol{x})$ matches $f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{S}$.

*4. Re-use the ordering of $\boldsymbol{w}^\top \boldsymbol{x}^{(i)}$, which for all $0 \leq j < i \leq N-1$:

$$\boldsymbol{w}^\top \boldsymbol{x}^{(j)} > \boldsymbol{w}^\top \boldsymbol{x}^{(i)}$$

Given, $\boldsymbol{w} = \lambda \boldsymbol{u}, \boldsymbol{b}_j^{(1)} = -\lambda \boldsymbol{u}^\top \boldsymbol{x}^{(j)}, \phi(-\infty) = 0$ and $\phi(0) > 0$, for $0 \leq i,\ j \leq N-1$, we have :

$$\boldsymbol{M}_{ij} = \begin{cases} \phi\big(\lambda(\boldsymbol{u}^\top \boldsymbol{x}^{(i)} - \boldsymbol{u}^\top \boldsymbol{x}^{(j)})\big) & \text{for } 0 \leq j < N-1 \\ 1 & \text{for } j = N-1 \end{cases}$$

Similarly,

$$\lim_{\lambda \to \infty} \boldsymbol{M} = \begin{cases} \phi(-\infty) = 0 & 0 \leq j < i \leq N-1 \\ \phi(0) > 0 & 0 \leq i = j < N-1 \\ \phi(+\infty) > 0 & 0 \leq i < j < N-1 \\ 1 & j = N-1 \end{cases}$$

indicating it is a triangular matrix with non-zero diagonal elements.

This proves that, for any $f : \mathbb{R}^n \to \mathbb{R}^m$ and any finite sample set $\mathcal{S} \subset \mathbb{R}^n$ of size N, there always exists a set of parameters for a two-layer network with $N-1$ neurons in hidden layer and a sigmoid-like activation function, such that the output $y(\boldsymbol{x})$ matches $f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{S}$.

Question 6. Compute the *full*, *valid*, and *same* convolution (with kernel flipping) for the following 1D matrices : $\big[1, 2, 3, 4\big] * \big[1, 0, 2\big]$

Answer 6. To compute the result of a convolution with **kernel flipping** for 1D matrices, we can use the following formula :

$$\boldsymbol{S}(i) = (\boldsymbol{K} * \boldsymbol{X})(i) = \sum_{n=0}^{k-1} \boldsymbol{X}(i+n)\boldsymbol{K}(k-1-n)$$

where the input $\boldsymbol{X}$, in this question, is variant for different convolution patterns, the kernel $\boldsymbol{K}$, in this question, is $[1, 0, 2]$, and k, the size of kernel, is 3.

Let s, i, k, p, o represent the size of stride, input, kernel, zero-padding, and output size respectively, we have :

$$o = \lfloor \frac{i + 2p - k}{s} \rfloor + 1$$

For *full*, *valid* and *same* convolution in this question, $i = 4, s = 1$ and $k = 3$ hold.

1. for *full* convolution : $p = k - 1 = 2, o = 6,\ \boldsymbol{X} = [0, 0, 1, 2, 3, 4, 0, 0]$, and the convolution result

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

2. for *valid* convolution : $p = 0, o = 2,\ \boldsymbol{X} = [1, 2, 3, 4]$, and the convolution result

$$\boldsymbol{S} = [5, 8]$$

3. for *same* convolution : $p = 1, o = i = 4$, $\boldsymbol{X} = [0, 1, 2, 3, 4, 0]$, and the convolution result

$$\boldsymbol{S} = [2, 5, 8, 6]$$

Question 7. Consider a convolutional neural network. Assume the input is a colorful image of size 256×256 in the RGB representation. The first layer convolves 64 8×8 kernels with the input, using a stride of 2 and no padding. The second layer downsamples the output of the first layer with a 5×5 non-overlapping max pooling. The third layer convolves 128 4×4 kernels with a stride of 1 and a zero-padding of size 1 on each border.

1. What is the dimensionality (scalar) of the output of the last layer ?
2. Not including the biases, how many parameters are needed for the last layer ?

Answer 7.

For question1, the dimensionality of the output of the last(third) layer is $128 \times 24 \times 24 = 73728$; for question 2, 131072 parameters are need for the last layer.

In general, a CNN architecture follows the following rules :

1. dimensionality of input data are user-defined.
2. number of channels(#Channels) of a convolutional layer is free to define.
3. pooling downsampling operation has **no** parameters and doesn't change the number of channels.
4. the output size per channel of a convolutional layer o is determined by its input size per channel i, kernel size :k, number of zero-padding :p, and strides :s, of the convolutional operation :

$$o = \lfloor \frac{i + 2p - k}{s} \rfloor + 1$$

5. not including the biases, number of parameters (#Parameters) for a convolutional layer is :

$$\#\text{Parameters} = (\text{kernel width}) \times (\text{kernel height}) \times \#\text{Channels(input)} \times \#\text{Channels(output)}$$

6. A squared kernel for a non-overlapping max polling operation means $k = s$

Based on the above rules, we built the table describing the detail of the CNN architecture.

TABLE 1 – CNN architecture

Layer	Role	#Channels	Size per channel	#Parameters
Input	original data	3	(256,256)	0
First	convolution(k=8,s=2,p=0)	64	(125,125)	$8{\times}8 \times 3 \times 64 = 12288$
Second	pooling&down sampling(s=5)	64	(25,25)	0
Third	convolution(k=4,s=1,p=1)	128	(24,24)	$4{\times}4 \times 64 \times 128 = 131072$

Question 8. Assume we are given data of size $3 \times 64 \times 64$. In what follows, provide the correct configuration of a convolutional neural network layer that satisfies the specified assumption. Answer with the window size of kernel (k), stride (s), padding (p), and dilation (d, with convention $d = 1$ for no dilation). Use square windows only (e.g. same k for both width and height).

1. The output shape of the first layer is $(64, 32, 32)$.
   (a) Assume $k = 8$ without dilation.
   (b) Assume $d = 7$, and $s = 2$.
2. The output shape of the second layer is $(64, 8, 8)$. Assume $p = 0$ and $d = 1$.
   (a) Specify k and s for pooling with non-overlapping window.
   (b) What is output shape if $k = 8$ and $s = 4$ instead ?
3. The output shape of the last layer is $(128, 4, 4)$.
   (a) Assume we are not using padding or dilation.
   (b) Assume $d = 2$, $p = 2$.
   (c) Assume $p = 1$, $d = 1$.

Answer 8. Using square windows only, the following two formulas define the relationship of following hyper-parameters : input size(i), output size(o), kernel size(k), effective kernel size(k'), zero-paddings(p), strides(s), and dilations(d) :

$$o = \lfloor \frac{i + 2p - k'}{s} \rfloor + 1$$

$$k' = k + (k-1)(d-1)$$

where $i, o, s, k, k' \in \mathbb{N}$, and $p \in \mathbb{Z}^{\geq 0}$.

All of the sub-questions can be regarded as finding the possible combinations of some hyper-parameters given others. I will first show the relationship between parameters by a formula, then list all possible combinations if there are finite solutions, and some possible combinations followed by $\cdots$ if the solution has infinity combinations, such as by infinitely and meaninglessly increasing the number of zero-padding while still fit the formula. Combination(s) marked with a '$*$' indicates it is practically usable or preferred to be used in practice.

1. $i = 64$, output shape of $(64, 32, 32)$ means $o = 32$.
   (a) Given $k = 8, d = 1$(without dilation) :

$$k' = 8 + (8-1)(1-1) = 8$$

$$32 = \lfloor \frac{64 + 2p - 8}{s} \rfloor + 1 \Leftrightarrow \lfloor \frac{56 + 2p}{s} \rfloor = 31$$

   s, p could be the following (* indicates the preferred configuration(s) for a CNN network, applied to all the following) :
      i. $s = 2, p = 3$ *
      ii. $s = 3, p = 18$
      iii. $s = 4, p = 34$
      iv. $\cdots$
   (b) Given $d = 7, s = 2$, then :

$$\lfloor \frac{64 + 2p - k'}{2} \rfloor = 31$$

$$k' = k + (k-1)(d-1) = 7k - 6$$

   p, k could be the following :

   i. $k = 1$ $(k' = 1), p = 0$ *
   ii. $k = 2$ $(k' = 8), p = 3$ *
   iii. $k = 3$ $(k' = 15), p = 7$ *
   iv. $k = 4$ $(k' = 22), p = 10$
   v. $\cdots$

2. $o = 8, i = 32, p = 0, d = 1$
   (a) For polling with non-overlapping window, it should be $k = s$,

$$8 = \lfloor \frac{32 - k}{k} \rfloor + 1 \Rightarrow k = 4$$

   Therefore :

$$k = 4, s = 4$$

   (b) if $k = 8(k' = 8,\ \text{as}\ d = 1), s = 4$, the output size should be :

$$o = \lfloor \frac{32 + 2 \times 0 - 8}{4} \rfloor + 1 = 7$$

   Therefore, the output shape could be $(64, 7, 7)$ if the number of channels maintains 64.

3. Given $i = 8, o = 4$
   (a) Given $p = 0, d = 1(k' = k)$

$$4 = \lfloor \frac{8 - k}{s} \rfloor + 1 \Rightarrow \lfloor \frac{8 - k}{s} \rfloor = 3$$

   Here are the following possible configurations :
   i. $s = 1, k = 5$ *
   ii. $s = 2, k = 2$ *

   (b) Given $d = 2(k' = 2k - 1), p = 2$

$$4 = \lfloor \frac{8 + 4 - (2k - 1)}{s} \rfloor + 1 \Rightarrow \lfloor \frac{13 - 2k}{s} \rfloor = 3$$

   Here are the following possible configurations :
   i. $k = 1(k' = 1), s = 3$ *
   ii. $k = 2(k' = 3), s = 3$ *
   iii. $k = 3(k' = 5), s = 2$ *
   iv. $k = 5(k' = 9), s = 1$ *

   (c) Given $p = 1, d = 1(k' = k)$

$$4 = \lfloor \frac{8 + 2 - k}{s} \rfloor + 1 \Rightarrow \lfloor \frac{10 - k}{s} \rfloor = 3$$

   Here are the following possible configurations :
   i. $s = 1, k = 7$ *
   ii. $s = 2, k = 4$ *
   iii. $s = 2, k = 3$ *
   iv. $s = 3, k = 1$ (not often used)