IFT6135-H2019 - Representation learning

Assignment 1, Theoretical Part

Sanae Lotfi

Num de matricule (Poly) : 1968682
Num de matricule (Udem) : 20147309

17th Februray, 2019

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 Gradescope.*

Question 1 (4-4-4-2)**.** 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. Write your answer here.

1. We know that the rectified linear unit, defined as : $g(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$, is continuous and differentiable on $\mathbb{R}_+^*$ and $\mathbb{R}_-^*$.

   Its derivative on these two intervals is : $g'(x) = \begin{cases} 1 = H(x) & \text{if } x > 0 \\ 0 = H(x) & \text{if } x < 0 \end{cases}$.

   g is differentiable in $x = 0$ if its left derivative and right derivative in $x = 0$ have the same value. We have :

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

   and

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

   thus, g is not differentiable in $x = 0$.

   We conclude that g differentiable on $\mathbb{R}_+^*$ and $\mathbb{R}_-^*$ and its derivative is equal to the Heaviside step function on those two intervals.

2. We can define g using H as follows : $\forall x \in \mathbb{R}, g(x) = x\,H(x)$ or $\forall x \in \mathbb{R}, g(x) = \int_{-\infty}^{x} H(x)$.
3. To show that $H(x)$ can be well approximated by the sigmoid function $\sigma(x) = \frac{1}{1+e^{-kx}}$ for large k, it's enough to study the value of σ when k takes a large value :
   - For $x = 0$, $\sigma(x) = \frac{1}{2} = H(x)$,
   - For $x > 0$, $\sigma(x) \approx \frac{1}{1+0} \approx 1 = H(x)$ (because $\lim_{k\to\infty} \exp(-kx) = 0$, $x > 0$)).
   - For $x < 0$, $\sigma(x) \approx 0 = H(x)$ (because $\lim_{k\to\infty} \exp(-kx) = +\infty$ when $x < 0$).
4. By definition, we have : $F'[\phi] = \int_{\mathbb{R}} F'(x)\phi(x)dx$
   We use the integration by parts to write, for given a and b in $\mathbb{R}$ such that $a < b$:

$$\begin{aligned}\int_a^b F'(x)\phi(x)dx &= [F(x)\phi(x)]_a^b - \int_a^b F(x)\phi'(x)dx \\ &= F(b)\phi(b) - F(a)\phi(a) - \int_a^b F(x)\phi'(x)dx\end{aligned} \tag{1}$$

   Since ϕ has a compact support, we can calculate the limit of the above equation when a goes to $-\infty$ and b goes to $+\infty$. This gives us (since ϕ is null for negative and positive high values) :

$$\begin{aligned}F'[\phi] = \int_{\mathbb{R}} F'(x)\phi(x)dx &= \lim_{a\to-\infty}\lim_{b\to+\infty}\int_a^b F'(x)\phi(x)dx \\ = 0 - \int_{\mathbb{R}} F(x)\phi'(x)dx &= -\int_{\mathbb{R}} F(x)\phi'(x)dx\end{aligned} \tag{2}$$

   If we use this formula to calculate $H'[\phi]$, then we have (since ϕ is null for negative and positive high values) :

$$\begin{aligned}H'[\phi] &= -\int_{\mathbb{R}} H(x)\phi'(x)dx \\ &= -\int_{\mathbb{R}_+} \phi'(x)dx \\ &= -\Big[\phi(x)\Big]_0^{+\infty} \\ &= \phi(0)\end{aligned}$$

Question 2 (5-8-5-5)**.** Let x be an n-dimensional 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 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. Let i and j be in $\{1,..,n\}$. The function $S_i : \boldsymbol{x} \in \mathbb{R}^n \mapsto S_i(\boldsymbol{x}) \in \mathbb{R}$, such that $S_i(\boldsymbol{x}) = S(\boldsymbol{x})_i = \frac{e^{\boldsymbol{x}_i}}{\sum_k e^{\boldsymbol{x}_k}}$, is differentiable with respect to x_j and we have :

$$\begin{aligned}
\frac{dS(\boldsymbol{x})_i}{d\boldsymbol{x}_j} &= \frac{\frac{de^{x_i}}{d\boldsymbol{x}_j}\sum_k e^{\boldsymbol{x}_k} - e^{x_i}\frac{d\sum_k e^{\boldsymbol{x}_k}}{d\boldsymbol{x}_j}}{\left(\sum_k e^{\boldsymbol{x}_k}\right)^2} \\
&= \frac{\delta_{ij} e^{x_i}\sum_k e^{\boldsymbol{x}_k} - e^{x_i}e^{x_j}}{\left(\sum_k e^{\boldsymbol{x}_k}\right)^2} \\
&= \frac{e^{\boldsymbol{x}_i}}{\sum_k e^{\boldsymbol{x}_k}}\left(\delta_{ij} - \frac{e^{\boldsymbol{x}_j}}{\sum_k e^{\boldsymbol{x}_k}}\right) \\
&= S(\boldsymbol{x})_i\left(\delta_{ij} - S(\boldsymbol{x})_j\right)
\end{aligned} \tag{3}$$

2. Let i and j be in $\{1,..,n\}$. We have :

$$\begin{aligned}
\left(\frac{\partial S(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)_{ij} &= \frac{dS(\boldsymbol{x})_i}{d\boldsymbol{x}_j} \\
&= S(\boldsymbol{x})_i\left(\delta_{ij} - S(\boldsymbol{x})_j\right) \\
&= \begin{cases} S(\boldsymbol{x})_i S(\boldsymbol{x})_j & \text{if } i \neq j \\ S(\boldsymbol{x})_i S(\boldsymbol{x})_i - S(\boldsymbol{x})_i & \text{if } i = j \end{cases}
\end{aligned} \tag{4}$$

Thus, we have :

$$\begin{aligned}
\frac{\partial S(\boldsymbol{x})}{\partial \boldsymbol{x}} &= \begin{bmatrix} \frac{dS(\boldsymbol{x})_1}{d\boldsymbol{x}_1} & ... & \frac{dS(\boldsymbol{x})_1}{d\boldsymbol{x}_n} \\ .. & ... & .. \\ \frac{dS(\boldsymbol{x})_n}{d\boldsymbol{x}_1} & ... & \frac{dS(\boldsymbol{x})_n}{d\boldsymbol{x}_n} \end{bmatrix} \\
&= \begin{bmatrix} -S(\boldsymbol{x})_1 S(\boldsymbol{x})_1 & ... & -S(\boldsymbol{x})_1 S(\boldsymbol{x})_n \\ .. & ... & .. \\ -S(\boldsymbol{x})_n S(\boldsymbol{x})_1 & ... & -S(\boldsymbol{x})_n S(\boldsymbol{x})_n \end{bmatrix} + \begin{bmatrix} S(\boldsymbol{x})_1 & ... & 0 \\ .. & ... & .. \\ 0 & ... & S(\boldsymbol{x})_n \end{bmatrix} \\
&= -S(\boldsymbol{x})S^\top(\boldsymbol{x}) + diag(S(\boldsymbol{x}))
\end{aligned} \tag{5}$$

3. Let i and j be in $\{1,..,n\}$. We have :

$$
\begin{aligned}
\Big(\frac{\partial \sigma(\boldsymbol{x})}{\partial \boldsymbol{x}}\Big)_{ij} &= \frac{d(1/(1+e^{-x_i}))}{d\boldsymbol{x}_j} \\
&= \frac{-de^{-x_i}/d\boldsymbol{x}_j}{\left(1+e^{-x_i}\right)^2} \\
&= \frac{\delta_{ij}e^{-x_i}}{\left(1+e^{-x_i}\right)^2} \\
&= \delta_{ij}\sigma(x_i)\left(1-\sigma_i(x_i)\right) \\
&= \begin{cases} \sigma(x_i)\left(1-\sigma(x_i)\right) & \text{if } i=j \\ 0 & \text{if } i \neq j \end{cases}
\end{aligned} \tag{6}
$$

Thus, we have :

$$
\begin{aligned}
\frac{\partial \sigma(\boldsymbol{x})}{\partial \boldsymbol{x}} &= \begin{bmatrix} \sigma(x_1)\left(1-\sigma(x_1)\right) & ... & 0 \\ .. & ... & .. \\ 0 & ... & \sigma(x_n)\left(1-\sigma(x_n)\right) \end{bmatrix} \\
&= diag(\sigma(\boldsymbol{x}))\, diag(1_n - \sigma(\boldsymbol{x}))
\end{aligned} \tag{7}
$$

where 1_n is the vector of length n with 1 in all its rows.

4. - If $f(\boldsymbol{x}) = \sigma(\boldsymbol{x})$, then

$$\nabla_{\boldsymbol{x}} L = (\frac{\partial \sigma(\boldsymbol{x})}{\partial \boldsymbol{x}})^{\top} \nabla_{\boldsymbol{y}} L = diag(\sigma(\boldsymbol{x}))\, diag(1_n - \sigma(\boldsymbol{x})) \nabla_{\boldsymbol{y}} L$$

The multiplication $diag(1_n - \sigma(\boldsymbol{x}))\nabla_{\boldsymbol{y}} L$ is a $\mathcal{O}(n)$ operation, because it can be seen as an element-wise multiplication between $\sigma(\boldsymbol{x}) - 1$ and $\nabla_{\boldsymbol{y}} L$. This will result a vector that we can name $\mathbf{v}$, then $\nabla_{\boldsymbol{x}} L = diag(\sigma(\boldsymbol{x}))\,\mathbf{v}$ can be seen as an element-wise product as well between $\sigma(\boldsymbol{x})$ and v, which is also a $\mathcal{O}(n)$ operation. Thus, the entire multiplication is simplified to a $\mathcal{O}(n)$ operation.

- If $f(\boldsymbol{x}) = S(\boldsymbol{x})$, then

$$\nabla_{\boldsymbol{x}} L = (\frac{\partial S(\boldsymbol{x})}{\partial \boldsymbol{x}})^{\top} \nabla_{\boldsymbol{y}} L = (-S(\boldsymbol{x})S^{\top}(\boldsymbol{x}) + diag(S(\boldsymbol{x})))\nabla_{\boldsymbol{y}} L = diag(S(\boldsymbol{x}))\nabla_{\boldsymbol{y}} L - S(\boldsymbol{x})(S^{\top}(\boldsymbol{x})\nabla_{\boldsymbol{y}} L)$$

The multiplication $diag(S(\boldsymbol{x}))\nabla_{\boldsymbol{y}} L$ is a $\mathcal{O}(n)$ operation, because it can be seen as an element-wise multiplication between $S(\boldsymbol{x})$ and $\nabla_{\boldsymbol{y}} L$. The multiplication $(S^{\top}(\boldsymbol{x})\nabla_{\boldsymbol{y}} L)$ is also a $\mathcal{O}(n)$ operation since its a multiplication between two vectros. This multiplication gives a scalar. Thus the rest is a product between a scalar and a vector. In conclusion, the matrix-vector multiplication is simplified to a $\mathcal{O}(n)$ operation.

Question 3 (3-3-3-3)**.** 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. Let c be in $\mathbb{R}$ and i in $\{1, .., n\}$, we have :

$$\begin{aligned} S(\boldsymbol{x}+c)_i &= e^{\boldsymbol{x}_i+c} / \sum_j e^{\boldsymbol{x}_j+c} \\ &= e^{\boldsymbol{x}_i}\, e^c / \sum_j e^{\boldsymbol{x}_j}\, e^c \\ &= e^{\boldsymbol{x}_i} / \sum_j e^{\boldsymbol{x}_j} \\ &= S(\boldsymbol{x})_i \end{aligned} \tag{8}$$

We conclude that $S(\boldsymbol{x}+c) = S(\boldsymbol{x})$, thus softmax is translation-invariant.

2. To prove this, it's enough to find a counterexample.
In fact, for $n = 2$, let's take $x_1 = 1$ and $x_2 = 0$ with $c = 2$. We have $S_c(\boldsymbol{x})_1 = S(2\boldsymbol{x})_1 = e^{2*1}/(e^{2*1}+e^{2*0}) = e^2/(e^2+1) \approx 0.88$. However, $S(\boldsymbol{x})_1 = e^1/(e^1+e^0) = e/(e+1) =\approx 0.71$.
It is clear that $S(c\boldsymbol{x}) \neq S(\boldsymbol{x})$ in general (the following part shows that it is not true in general for any n and $c = 0$ or arbitrarily large).
Let's find the expression of $S(c\boldsymbol{x})_i$ for a given i in $\{1, .., n\}$ and $c \geq 0$:

$$\begin{aligned} S(c\boldsymbol{x})_i &= e^{c\boldsymbol{x}_i} / \sum_j e^{c\boldsymbol{x}_j} \\ &= e^{c\boldsymbol{x}_i} / \sum_j e^{c\boldsymbol{x}_j} \\ &= 1 / \sum_j e^{c(\boldsymbol{x}_j - \boldsymbol{x}_i)} \end{aligned} \tag{9}$$

- If $c = 0$, then $S(c\boldsymbol{x})_i = 1/n$. In this case, all the classes have the same probability and the transformation becomes not interesting (since our goal is to be able to find the class with the highest probability).
- If c is arbitrarily large, let $L = argmax\{\boldsymbol{x}_j, j \in \{1, .., n\}\}$ be the set of the arguments all the elements of the vector $\boldsymbol{x}$ that share the highest value. We put $k = card(L)$. We have that :

$$S(c\boldsymbol{x})_i = \begin{cases} \frac{1}{k} & \text{if } i \in L \\ 0 & i \notin L \end{cases} \tag{10}$$

We notice then that in this case, any little difference between the values of the vector $\boldsymbol{x}$ is translated by big changes in terms of probability. In case $k = 1$, we obtain that the element of the vector $\boldsymbol{x}$ with the highest value gets a probabiliy of 1 while all other elements get a probability of 0. This makes us lose a lot of information concerning to which extent the element with the highest value is superior to other elements.

3. Let $\boldsymbol{x}$ be a 2-dimentional vector. One can represent a 2-class categorical probability using softmax $S(\boldsymbol{x})$. We have :

$$\begin{aligned} S(\boldsymbol{x}) &= \begin{bmatrix} e^{\boldsymbol{x}_1}/e^{\boldsymbol{x}_1}+e^{\boldsymbol{x}_2} \\ e^{\boldsymbol{x}_2}/e^{\boldsymbol{x}_1}+e^{\boldsymbol{x}_2} \end{bmatrix} \\ &= \begin{bmatrix} 1/(1+e^{\boldsymbol{x}_2-\boldsymbol{x}_1}) \\ 1-1/(1+e^{\boldsymbol{x}_2-\boldsymbol{x}_1}) \end{bmatrix} \\ &= \begin{bmatrix} \sigma(\boldsymbol{x}_1-\boldsymbol{x}_2) \\ 1-\sigma(\boldsymbol{x}_1-\boldsymbol{x}_2) \end{bmatrix} \\ &= [\sigma(z), 1-\sigma(z)]^\top \end{aligned} \tag{11}$$

where $z = \boldsymbol{x}_1 - \boldsymbol{x}_2$.

4. Let $\boldsymbol{x}$ be a K-dimentional vector ($K \geq 2$). We can write :

- $S(\boldsymbol{x})_1 = e^{x_1}/\sum_{j=1}^K e^{x_j} = 1/1 + \sum_{j=2}^K e^{x_j - x_1}$,
- $\forall i \in \{2,..,K\}, S(\boldsymbol{x})_i = e^{x_i}/\sum_{j=1}^K e^{x_j} = e^{x_i-x_1}/1 + \sum_{j=2}^K e^{x_j-x_1}$

If we put $y_i = x_{i+1} - x_1, \forall i \in \{1,..,K-1\}$, then we have :

- $S(\boldsymbol{x})_1 = e^0/e^0 + \sum_{j=1}^{K-1} e^{y_j}$,
- $\forall i \in \{2,..,K\}, S(\boldsymbol{x})_i = e^{y_{i-1}}/e^0 + \sum_{j=1}^{K-1} e^{y_j}$

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

Question 4 (15)**.** 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. Let x be in $\mathbb{R}$, we have :

$$\begin{aligned} tanh(x) &= (e^x - e^{-x})/(e^x + e^{-x}) \\ &= 2 \times 1/(1+e^{-2x}) - 1 \\ &= 2\sigma(2x) - 1 \end{aligned} \tag{12}$$

Thus, $\sigma(x) = \frac{1}{2} tanh(\frac{x}{2}) + \frac{1}{2}$.

Let $\boldsymbol{x} \in \mathbb{R}^D$. We write then :

$$\begin{aligned}
y(x,\Theta,\sigma)_k &= \sum_{j=1}^{M}\omega_{kj}^{(2)}\sigma\left(\sum_{i=1}^{D}\omega_{ji}^{(1)}x_i+\omega_{j0}^{(1)}\right)+\omega_{k0}^{(2)} \\
&= \sum_{j=1}^{M}\left[\frac{\omega_{kj}^{(2)}}{2}tanh\left(\sum_{i=1}^{D}\frac{\omega_{ji}^{(1)}}{2}x_i+\frac{\omega_{j0}^{(1)}}{2}\right)+\frac{\omega_{kj}^{(2)}}{2}\right]+\omega_{k0}^{(2)} \\
&= \sum_{j=1}^{M}\left[\frac{\omega_{kj}^{(2)}}{2}tanh\left(\sum_{i=1}^{D}\frac{\omega_{ji}^{(1)}}{2}x_i+\frac{\omega_{j0}^{(1)}}{2}\right)\right]+\sum_{j=1}^{M}\frac{\omega_{kj}^{(2)}}{2}+\omega_{k0}^{(2)}
\end{aligned} \tag{13}$$

We conclude then 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 $x \in \mathbb{R}^D$, where :

- $\tilde{\omega}^{(1)} = \frac{1}{2}\omega^{(1)}$
- $\tilde{\omega}^{(2)}$ is defined such as : $\tilde{\omega}_{kj}^{(2)} = \begin{cases} \frac{1}{2}\omega_{kj}^{(2)} & \text{if } j \neq 0 \\ \sum_{l=1}^{M}\frac{\omega_{kl}^{(2)}}{2}+\omega_{k0}^{(2)} = \sum_{l=0}^{M}\frac{\omega_{kl}^{(2)}}{2}+\frac{\omega_{k0}^{(2)}}{2} & \text{if } j = 0 \end{cases}$

We can write Θ' in a more compact way using Θ :

$$\Theta' = \frac{1}{2}\Theta + \frac{1}{2}(Z, \omega^{(2)}B)$$

where Z is a zero matrix of the same dimensions as $\omega^{(1)}$ and B is a matrix of the same dimensions as $\omega^{(2)\top}$ with $B_{ij} = \begin{cases} 1 & \text{if } j = 1 \\ 0 & \text{otherwise} \end{cases}$ (the first column contains values equal to one and the rest of the matrix has values equal to 0).

An even more compact way consists of writing Θ' as follows :

$$\Theta' = \frac{1}{2}\Theta + \frac{1}{2}\Theta\tilde{B}$$

Where $\tilde{B}$ is a $(M + D + 2) \times (K + M)$ matrix, defined as follows :

$$\tilde{B}_{ij} = \begin{cases} 1 & \text{if } i \in \{D+2, .., D+M+2\} \text{ and } j = 1 \\ 0 & \text{otherwise} \end{cases}$$

Question 5 (2-2-2-2)**.** 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 $\boldsymbol{W}^{(1)} = [\boldsymbol{w}, \cdots, \boldsymbol{w}]^T$ 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. **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. **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. Let's 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 $(\boldsymbol{W}^{(2)}, \boldsymbol{b}^{(2)})$:

$$y(x)_k = \sum_{j=1}^{N-1} \omega_{kj}^{(2)} \phi \left(\sum_{i=1}^{n} \omega_{ji}^{(1)} x_i + b_j^{(1)} \right) + b_k^{(2)}$$

We can write it in a compact way :

$$y(\boldsymbol{x})_k = \boldsymbol{w}_k^{(2)\top} \phi \left(\boldsymbol{W}^{(1)} \boldsymbol{x} + \boldsymbol{b}^{(1)} \right) + b_k^{(2)}$$

where $\phi\left(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}\right)_j = \phi((\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)})_j) = \phi\left(\sum_{i=1}^n \omega_{ji}^{(1)} x_i + b_j^{(1)}\right)$, for $j \in \{1, .., N-1\}$ and $\boldsymbol{W}^{(2)} = [\boldsymbol{w}_1^{(2)}, \cdots, \boldsymbol{w}_m^{(2)}]^T$.

Thus, we can write (using the same convention for ϕ applied to a vector) :

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

2. The interpolation problem on the sample set $\mathcal{S} = \{\boldsymbol{x}^{(1)}, \cdots \boldsymbol{x}^{(N)}\} \subset \mathbb{R}^n$ is about solving the equation :

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

.

Thus, we need to solve the equations :

$$\begin{aligned}
& f(\boldsymbol{x}^{(i)}) = \boldsymbol{W}^{(2)} \phi \left(\boldsymbol{W}^{(1)} \boldsymbol{x}^{(i)} + \boldsymbol{b}^{(1)} \right) + \boldsymbol{b}^{(2)}, \quad \forall i \in \{1, .., N\} \\
\implies & f^\top(\boldsymbol{x}^{(i)}) = \phi \left(\boldsymbol{W}^{(1)} \boldsymbol{x}^{(i)} + \boldsymbol{b}^{(1)} \right)^\top \boldsymbol{W}^{(2)\top} + \boldsymbol{b}^{(2)\top}, \quad \forall i \in \{1, .., N\} \\
\implies & f^\top(\boldsymbol{x}^{(i)}) = [\phi \left(\boldsymbol{W}^{(1)} \boldsymbol{x}^{(i)} + \boldsymbol{b}^{(1)} \right)^\top, 1] \begin{bmatrix} \boldsymbol{W}^{(2)\top} \\ \boldsymbol{b}^{(2)\top} \end{bmatrix}, \quad \forall i \in \{1, .., N\}
\end{aligned} \tag{14}$$

where $\phi\left(\boldsymbol{W}^{(1)}\boldsymbol{x}^{(i)}+\boldsymbol{b}^{(1)}\right)_j=\phi((\boldsymbol{W}^{(1)}\boldsymbol{x}^{(i)}+\boldsymbol{b}^{(1)})_j)=\phi\left(\boldsymbol{w}^\top\boldsymbol{x}^{(i)}+b_j^{(1)}\right)$, for $j\in\{1,..,N-1\}$.
Thus, we need to solve the equation :

$$\boldsymbol{F}=[f(\boldsymbol{x}^{(1)}),\cdots,f(\boldsymbol{x}^{(N)})]^\top=\begin{bmatrix}\sigma\left(\boldsymbol{W}^{(1)}\boldsymbol{x}^{(1)}+\boldsymbol{b}^{(1)}\right)^\top & 1\\ . & .\\ . & .\\ \sigma\left(\boldsymbol{W}^{(1)}\boldsymbol{x}^{(N)}+\boldsymbol{b}^{(1)}\right)^\top & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{W}^{(2)\top}\\ \boldsymbol{b}^{(2)\top}\end{bmatrix}\tag{15}$$

We put :

$$\tilde{\boldsymbol{W}}^{(2)}=[\boldsymbol{W}^{(2)},\boldsymbol{b}^{(2)}]^\top\quad\text{and}\quad\boldsymbol{M}=\begin{bmatrix}\phi\left(\boldsymbol{W}^{(1)}\boldsymbol{x}^{(1)}+\boldsymbol{b}^{(1)}\right)^\top & 1\\ . & .\\ . & .\\ \phi\left(\boldsymbol{W}^{(1)}\boldsymbol{x}^{(N)}+\boldsymbol{b}^{(1)}\right)^\top & 1\end{bmatrix}$$

We can write the : $\boldsymbol{M}\tilde{\boldsymbol{W}}^{(2)}=\boldsymbol{F}$. Where $\boldsymbol{M}$ is an $N\times N$ matrix defined by :

$$M_{ij}=\begin{cases}\phi\left(\boldsymbol{W}^{(1)}\boldsymbol{x}^{(i)}+\boldsymbol{b}^{(1)}\right)_j=\phi\left(\boldsymbol{w}^\top\boldsymbol{x}^{(i)}+b_j^{(1)}\right), & \text{if } i\in\{1,..,N\}\text{ and } j\in\{1,..,N-1\}\\ 1, & \text{if } i\in\{1,..,N\}\text{ and } j=N\end{cases}$$

Conclusion : We showed 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 $\boldsymbol{M}$ are given by the expressions above.

3. **Proof with Relu activation.**
Let's assume that $\boldsymbol{x}^{(i)}$ are all distinct. and choose $\boldsymbol{w}$ such that $\boldsymbol{w}^\top\boldsymbol{x}^{(i)}$ are also all distinct (the existence of $\boldsymbol{w}$ was proved in assignment 0). Let $\boldsymbol{b}_j^{(1)}=-\boldsymbol{w}^\top\boldsymbol{x}^{(j)}+\epsilon$, where $\epsilon>0$. Let's find a value of ϵ such that $\boldsymbol{M}$ is triangular with non-zero diagonal elements.
We have from the previous question that :

$$\begin{aligned}\mathrm{M}_{ij}&=\begin{cases}\phi\left(\boldsymbol{w}^\top\boldsymbol{x}^{(i)}+b_j^{(1)}\right), & \text{if } i\in\{1,..,N\}\text{ and } j\in\{1,..,N-1\}\\ 1, & \text{if } i\in\{1,..,N\}\text{ and } j=N\end{cases}\\ &=\begin{cases}max\left(\boldsymbol{w}^\top\boldsymbol{x}^{(i)}+b_j^{(1)},0\right), & \text{if } i\in\{1,..,N\}\text{ and } j\in\{1,..,N-1\}\\ 1, & \text{if } i\in\{1,..,N\}\text{ and } j=N\end{cases}\\ &=\begin{cases}max\left(\boldsymbol{w}^\top\boldsymbol{x}^{(i)}-\boldsymbol{w}^\top\boldsymbol{x}^{(j)}+\epsilon,0\right), & \text{if } i\in\{1,..,N\}\text{ and } j\in\{1,..,N-1\}\\ 1, & \text{if } i\in\{1,..,N\}\text{ and } j=N\end{cases}\end{aligned}$$

Without loss of generality, we suppose an order on our vectors : $\boldsymbol{w}^\top\boldsymbol{x}^{(1)}>\ldots>\boldsymbol{w}^\top\boldsymbol{x}^{(N)}$ (the inequalities are strict since the values are distinct by choice of $\boldsymbol{w}$).

We put $\epsilon=min\{\boldsymbol{w}^\top\boldsymbol{x}^{(i)}-\boldsymbol{w}^\top\boldsymbol{x}^{(j)},\ i<j,\ i\text{ and }j\text{ in }\{1,..,N\}\}$.Using our hypothesis, we see that $\epsilon>0$. Furthermore, by definition of epsilon, we have for $i>j$ that : $\alpha_{ij}=\boldsymbol{w}^\top\boldsymbol{x}^{(i)}-\boldsymbol{w}^\top\boldsymbol{x}^{(j)}+\epsilon=\epsilon-(\boldsymbol{w}^\top\boldsymbol{x}^{(j)}-\boldsymbol{w}^\top\boldsymbol{x}^{(i)})\leq 0$. Thus, $M_{ij}=max(\alpha_{ij},0)=0$ for $i>j$ and $j\neq N$. The matrix $\boldsymbol{M}$ is upper triangular. The diagonal elements are given by :

$$M_{ii} = \begin{cases} max\left(\boldsymbol{w}^\top \boldsymbol{x}^{(i)} - \boldsymbol{w}^\top \boldsymbol{x}^{(i)} + \epsilon, 0\right) = max\,(\epsilon, 0) = \epsilon & \text{if } i \neq N \\ 1 & \text{if } i = N \end{cases} \tag{16}$$

Which means that for all $i \in \{1, .., N\}$, $M_{ii} > 0$. In other words, $\boldsymbol{M}$ is triangular with non-zero diagonal elements. Thus $\boldsymbol{M}$ is invertible and the matrix equation $\boldsymbol{M}\tilde{\boldsymbol{W}}^{(2)} = \boldsymbol{F}$ can be solved using the inverse of $\boldsymbol{M}$. Conclusion : the subset $\mathcal{S}$ being fixed as well as the vector $\boldsymbol{w}$, we can find the weights matrix $\tilde{\boldsymbol{W}}^{(2)} = \boldsymbol{M}^{-1}\boldsymbol{F}$ such that we interpolate f with y.

4. **Proof with sigmoid-like activations**.
Let's assume ϕ is continuous, bounded, $\phi(-\infty) = 0$ and $\phi(0) > 0$. We decompose $\boldsymbol{w}$ as $\boldsymbol{w} = \lambda\boldsymbol{u}$ and set $\boldsymbol{b}_j^{(1)} = -\lambda\boldsymbol{u}^\top\boldsymbol{x}^{(j)}$. Fixing $\boldsymbol{u}$, let's show that $\lim_{\lambda\to+\infty} \boldsymbol{M}$ is triangular with non-zero diagonal elements. We have :

$$\mathrm{M}_{ij} = \begin{cases} \phi\left(\boldsymbol{w}^\top\boldsymbol{x}^{(i)} + b_j^{(1)}\right), & \text{if } i \in \{1, .., N\} \text{ and } j \in \{1, .., N-1\} \\ 1, & \text{if } i \in \{1, .., N\} \text{ and } j = N \end{cases}$$
$$= \begin{cases} \phi\left(\lambda(\boldsymbol{u}^\top\boldsymbol{x}^{(i)} - \boldsymbol{u}^\top\boldsymbol{x}^{(j)})\right), & \text{if } i \in \{1, .., N\} \text{ and } j \in \{1, .., N-1\} \\ 1, & \text{if } i \in \{1, .., N\} \text{ and } j = N \end{cases}$$

The diagonal elements are given by :

$$M_{ii} = \begin{cases} \phi(0) & \text{if } i \neq N \\ 1 & \text{if } i = N \end{cases} \tag{17}$$

Which means that for all $i \in \{1, .., N\}$, $M_{ii} > 0$.

We take $\lambda > 0$.Just like the previous question, we assume (without loss of generality) an order on our vectors : $\boldsymbol{u}^\top\boldsymbol{x}^{(1)} > \ldots > \boldsymbol{u}^\top\boldsymbol{x}^{(N)}$ (the inequalities are strict since the values are distinct by choice of $\lambda\boldsymbol{u}$ and since the factor $\lambda > 0$, this doesn't change the inequality).

Thus, for $i > j$ and $j \neq N$, $\lim_{\lambda\to+\infty} M_{ij} = \phi(-\infty) = 0$, using the order hypothesis.
From the other side, we assume that the value of $\phi(+\infty) = l$ exists (some continuous and bounded functions don't have a finite limit in $+\infty$). We have, for $i < j$ and $j \neq N$, $\lim_{\lambda\to+\infty} M_{ij} = \phi(+\infty) = l$.

Conclusion : $\lim_{\lambda\to+\infty} \boldsymbol{M}$ is triangular with non-zero diagonal elements. Thus it is invertible and the matrix equation $\boldsymbol{M}\tilde{\boldsymbol{W}}^{(2)} = \boldsymbol{F}$ can be solved using the inverse of the limit matrix of $\boldsymbol{M}$ because

$\tilde{\boldsymbol{W}}^{(2)}$ and $\boldsymbol{F}$ do not depend on λ. In other words, the subset $\mathcal{S}$ being fixed as well as the vector $\boldsymbol{w}$, we can find the weights matrix $\tilde{\boldsymbol{W}}^{(2)} = \boldsymbol{M}^{-1}\boldsymbol{F}$ such that we interpolate f with y.

Question 6 (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 convolution, we will the use the expression $(x{*}k)_{ij} = \sum_{p,q} x_{i+p,j+q} k_{r_1-p,r_2-q}$ where $r_1 \times r_2$ is the size of the kernel.

- *full convolution :* consists of adding the maximum zero-padding such that the convolution product with the kernel still takes into account elements from the original matrix. In this case, the maximum possible zero-padding is 2. We obtain the following 1D matrix : $[1, 2, 5, 8, 6, 8]$.
- *valid convolution :* consists of adding no zero-padding and performing the classic convolution product between the original matrix and the kernel. We obtain the following 1D matrix : $[5, 8]$.
- *same convolution :* consists of adding the enough zero-padding such that the output of the convolution product has the same dimension as the original matrix. In this case, the necessary zero-padding is 1. We obtain the following 1D matrix : $[2, 5, 8, 6]$.

Question 7 (5-5)**.** 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. 1. The dimensionality of the feature map after the first layer is given by the relation : $o = \lfloor \frac{i+2p-(d(k-1)+1)}{s} \rfloor + 1$, where i is the size of the input, p is the padding, d is the dilation, k is the size of the kernel and s is the stride.

Thus : $o = \lfloor \frac{256+2\times 0-(1\times(8-1)+1)}{2} \rfloor + 1 = 125$ and the output shape of the first layer is $(64, 125, 125)$. In the same way we calculate the output of the next layers :

- The output shape of the second layer is $(64, 25, 25)$ (because $d = \lfloor \frac{125+2\times 0-5}{1} \rfloor + 1 = 25$).

- The output shape of the third layer is $(128, 24, 24)$ (because $d = \lfloor \frac{25+2\times 1-4}{1} \rfloor + 1 = 24$).

Thus the dimensionality (scalar) of the output of the last layer is : $128 \times 24 \times 24 = 73728$.

2. The number of parameters that are needed for the last layer is : $64 \times 4 \times 4 \times 128 = 131072$ parameters.

Question 8 (4-4-4)**.** 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. We base the answers of all the following questions on the following expression of the output shape of a convolutional layer : $o = \lfloor \frac{i+2p-(d(k-1)+1)}{s} \rfloor + 1$, where i is the size of the input, p is the padding, d is the dilation, k is the size of the kernel and s is the stride. The only reason we didn't give the details of the calculations is because they are very repetitive and simple

Here are the answers :

1. The output shape of the first layer is $(64, 32, 32)$.
   (a) Assuming $k = 8$ without dilation ($d = 1$), we can take : $p = 3$ and $s = 2$.
   (b) Assuming $d = 7$, and $s = 2$, we can take : $k = 2$ and $p = 3$.
2. The output shape of the second layer is $(64, 8, 8)$. Assume $p = 0$ and $d = 1$.
   (a) For pooling with non-overlapping window, we should have $k \leq s$, we can take : $k = 4$ and $s = 4$
   (b) The output shape if $k = 8$ and $s = 4$ is : $(64, 7, 7)$.
3. The output shape of the last layer is $(128, 4, 4)$.
   (a) Assuming we are not using padding ($p = 0$) or dilation ($d = 1$), we can take : $k = 5$ and $s = 1$.
   (b) Assuming $d = 2$, $p = 2$, we can take : $k = 5$ and $s = 1$.
   (c) Assuming $p = 1$, $d = 1$, we can take : $k = 4$ and $s = 2$.