IFT-3395/6390 Fundamentals of Machine Learning,
Professors : Guillaume Rabusseau / Ioannis Mitliagkas

Solution for Homework 3

This solution is based on the work submitted by some of your colleagues.

Question 1

A) We define the sign function as :

$$\text{sign}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$$

Using only indicator functions, the sign function can be written as :

$$\text{sign}(x) = \mathbb{1}_{x>0}(x) - \mathbb{1}_{x<0}(x)$$

B) The derivative of the relu function can be expressed as the following :

$$g'(x) = \begin{cases} 1, & \text{when } x > 0 \\ 0, & \text{when } x < 0 \end{cases}$$

g'(x) = Heaviside(x)

Note that the derivative does not exist at x=0.

C) By fundamental theorem of calculus :

$$\text{RELU}(x) = \int H(x')\, dx'$$

Also :

$$\text{ReLU}(x) = \begin{cases} xH(x) & \text{if} \quad x \neq 0 \\ 0 & \text{if} \quad x = 0 \end{cases}$$

$$\text{ReLU}(x) = \begin{cases} x - xH(-x) & \text{if} \quad x \neq 0 \\ 0 & \text{if} \quad x = 0 \end{cases}$$

D)

$$\text{For } x > 0 : \lim_{k\to\infty} a(x) = \lim_{k\to\infty} \frac{1}{1+e^{-kx}} = \frac{1}{1+e^{-\infty}} = 1$$
$$\text{For } x = 0 : \lim_{k\to\infty} a(0) = \lim_{k\to\infty} \frac{1}{1+e^{-k\cdot 0}} = \frac{1}{1+1} = \frac{1}{2}$$
$$\text{For } x < 0 : \lim_{k\to\infty} a(x) = \lim_{k\to\infty} \frac{1}{1+\underbrace{e^{-k\cdot x}}_{k\cdot x<0}} = \frac{1}{1+\infty} = 0$$
$$\lim_{k\to\infty} a(x) = \begin{cases} 1 & \text{si } x > 0 \\ 1/2 & \text{si } x = 0 \\ 0 & \text{si } x < 0 \end{cases} = H(x)$$

E)
Knowing that the sigmoid function is : $\sigma(x) = \frac{1}{1+e^{-x}}$, we can calculate its derrivative.

$$\begin{aligned}
\frac{\partial\sigma(x)}{\partial x} = \frac{\partial \frac{1}{1+e^{-x}}}{\partial x} &= \frac{\partial(1+e^{-x})^{-1}}{\partial x} \\
&= -(1+e^{-x})^{-2} \times -e^{-x} \quad \text{(Chain Rule)} \\
&= \frac{e^{-x}}{(1+e^{-x})^2} \\
&= \frac{1}{(1+e^{-x})}\frac{e^{-x}}{(1+e^{-x})} \\
&= \sigma(x)\frac{e^{-x}}{(1+e^{-x})} \\
&= \sigma(x)\left(\frac{1+e^{-x}-1}{(1+e^{-x})}\right) \\
&= \sigma(x)\left(\frac{1+e^{-x}}{(1+e^{-x})} - \frac{1}{(1+e^{-x})}\right) \\
&= \sigma(x)(1-\sigma(x))
\end{aligned}$$

let $i, j \in [\![1, n]\!] \times [\![1, n]\!]$

$$\begin{aligned}
\frac{d\sigma(x)_i}{dx_j} &= \frac{d}{dx_j}\left(\frac{1}{1+e^{-x_i}}\right) \\
&= \frac{-\delta_{ij}e^{-x_i}}{\left(1+e^{-x_i}\right)^2} \\
&= -\delta_{ij}\frac{1+e^{-x_i}-1}{\left(1+e^{-x_i}\right)^2} \\
&= \delta_{ij}\sigma(x)_i\left(1-\sigma(x)_i\right)
\end{aligned}$$

So,

$$\forall i, j \in [\![1, n]\!] \times [\![1, n]\!], i \neq j \Longrightarrow \frac{d\sigma(x)_i}{dx_j} = 0$$

thus

$$\forall i \in [\![1, n]\!], \quad \frac{d\sigma(x)_i}{dx_i} = \sigma(x)_i \left(1 - \sigma(x)_i\right)$$

the Jacobian matrix of σ is :

$$\frac{\partial \sigma(x)}{\partial x} = \text{diag}(\sigma(x)) \left(I_n - \text{diag}(\sigma(x))\right)$$

F)
Showing that $ln(\sigma(x)) = -softplus(-x)$

$$\begin{aligned} ln(\sigma(x)) &= ln\left(\frac{1}{1 + e^{-x}}\right) \\ &= ln(1) - ln(1 + e^{-x}) \quad \text{(Log properties)} \\ &= 0 - ln(1 + e^{-x}) = -softplus(-x) \end{aligned}$$

G)
Showing that softplus(x) - softplus(-x) = x :

$$\begin{aligned} softplus(x) - softplus(-x) &= ln(1 + e^{x}) - ln(1 + e^{-x}) \\ &= ln(1 + e^{x}) - ln(e^{-x}(1 + e^{x})) \\ &= ln\left(\frac{1 + e^{x}}{1 + e^{x}} \frac{1}{e^{-x}}\right) \\ &= ln\left(\frac{1}{e^{-x}}\right) \\ &= ln(e^{x}) \\ &= x \end{aligned}$$

And,

$$\begin{aligned} \frac{d}{dx}[\text{softplus}(x) - \text{ soffplus }(-x)] &= \frac{d}{dx} \cdot x \\ \frac{d}{dx} \text{ softplus }(x) &= 1 + \frac{d}{dx} \text{ softplus }(-x) \\ &= 1 - \frac{d}{dx}[- \text{ softplus }(-x)] \\ &= 1 - \frac{d}{dx} \ln(\sigma(x)) \\ &= 1 - \frac{1}{\sigma(x)} \cdot \sigma(x)[1 - \sigma(x)] \\ \frac{d}{dx} \text{softplus}(x) &= \sigma(x) \end{aligned}$$

H)
Showing that the Softmax is translation-invariant :

$$s(x+c)_i = \frac{e^{x_i+c}}{\sum_j e^{x_j+c}} = \frac{e^{x_i}e^c}{\sum_j e^{x_j}e^c} = \frac{e^c e^{x_i}}{e^c \sum_j e^{x_j}} = \frac{e^{x_i}}{\sum_j e^{x_j}} = s(x)_i$$

Therefore, the softmax function is translation invariant.

I) The softmax function is not invariant under multiplication, because when $-1 < x < 1$, multiplication will shrink $|x|$, whereas when $x > 1$ and $x < -1$, multiplcation will increase $|x|$. Multiplication would have the effect of making the softmax more certian of the most likely output, i.e., it would sharpen the distribution of probabilities computed by the softmax.

J)
Let's first calculate the partial derivative when i=j (is a diagonal element) :

$$\begin{aligned}
\frac{\partial \frac{e^{x_i}}{\sum_k e^{x_k}}}{\partial x_j} &= \frac{e^{x_i}\sum_k e^{x_k} - e^{x_j}e^{x_i}}{\left(\sum_k e^{x_k}\right)^2} \quad \text{(Quotient rule)} \\
&= \frac{e^{x_i}}{\sum_k e^{x_k}} \frac{\sum_k e^{x_k} - e^{x_j}}{\sum_k e^{x_k}} \\
&= S(x)_i \left(\frac{\sum_k e^{x_k}}{\sum_k e^{x_k}} - \frac{e^{x_j}}{\sum_k e^{x_k}} \right) \\
&= S(x)_i \left(1 - S(x)_j\right) \\
&= S(x)_i - S(x)_i S(x)_j
\end{aligned}$$

Now, let's calculate the partial derivative when $i \neq j$ (is a off-diagonal element) :

$$\begin{aligned}
\frac{\partial \frac{e^{x_i}}{\sum_k e^{x_k}}}{\partial x_j} &= \frac{0\sum_k e^{x_k} - e^{x_j}e^{x_i}}{\left(\sum_k e^{x_k}\right)^2} \quad \text{(Quotient rule)} \\
&= \frac{-e^{x_i}e^{x_j}}{\sum_k e^{x_k}} \\
&= -\frac{e^{x_i}}{\sum_k e^{x_k}} \frac{ex^j}{\sum_k e^{x_k}} \\
&= -S(x)_i S(x)_j
\end{aligned}$$

Therefore, $\frac{\partial S(x)_i}{\partial x_j} = S(x)_i \delta_{i=j} - S(x)_i S(x)_j$

k)

$$\frac{\partial S(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial S(\mathbf{x})_1}{\partial \mathbf{x}} \\ \frac{\partial S(\mathbf{x})_2}{\partial \mathbf{x}} \\ \vdots \\ \frac{\partial S(\mathbf{x})_n}{\partial \mathbf{x}} \end{bmatrix} = \begin{bmatrix} \frac{\partial S(\mathbf{x})_1}{\partial x_1} & \frac{\partial S(\mathbf{x})_1}{\partial x_2} & \cdots & \frac{\partial S(\mathbf{x})_1}{\partial x_n} \\ \frac{\partial S(\mathbf{x})_2}{\partial x_1} & \frac{\partial S(\mathbf{x})_2}{\partial x_2} & \cdots & \frac{\partial S(\mathbf{x})_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial S(\mathbf{x})_n}{\partial x_1} & \frac{\partial S(\mathbf{x})_n}{\partial x_2} & \cdots & \frac{\partial S(\mathbf{x})_n}{\partial x_n} \end{bmatrix}$$

$$= \begin{bmatrix} S(\mathbf{x})_1(1 - S(\mathbf{x})_1) & S(\mathbf{x})_1(0 - S(\mathbf{x})_2) & \dots & S(\mathbf{x})_1(0 - S(\mathbf{x})_n) \\ S(\mathbf{x})_2(0 - S(\mathbf{x})_1) & S(\mathbf{x})_2(1 - S(\mathbf{x})_2) & \dots & S(\mathbf{x})_2(0 - S(\mathbf{x})_n) \\ \vdots & \vdots & \ddots & \vdots \\ S(\mathbf{x})_n(0 - S(\mathbf{x})_1) & S(\mathbf{x})_n(0 - S(\mathbf{x})_2) & \dots & S(\mathbf{x})_n(1 - S(\mathbf{x})_n) \end{bmatrix}$$

$$= \underbrace{diag(S(\mathbf{x}))}_{n \times n} - \underbrace{\underbrace{S(\mathbf{x})}_{n \times 1} \underbrace{S(\mathbf{x})^T}_{1 \times n}}_{n \times n}$$

We assume that vectors are by default column vectors.

l)

$$\begin{aligned} \nabla_u \log S(\mathbf{x}(\mathbf{u}))_i &= \frac{\partial}{\partial u} \log \left(\frac{e^{\mathbf{x}(u)_i}}{\sum_{j=1}^n e^{\mathbf{x}(u)_j}} \right) \\ &= \frac{\partial}{\partial u} \log e^{\mathbf{x}(u)_i} - \frac{\partial}{\partial u} \log \left(\sum_{j=1}^n e^{\mathbf{x}(u)_j} \right) \\ &= \frac{\partial}{\partial u} \mathbf{x}(u)_i - \frac{\frac{\partial}{\partial u} \sum_{k=1}^n e^{\mathbf{x}(u)_k}}{\sum_{j=1}^n e^{\mathbf{x}(u)_j}} \\ &= \frac{\partial}{\partial u} \mathbf{x}(u)_i - \sum_{k=1}^n \frac{e^{\mathbf{x}(u)_k}}{\sum_{j=1}^n e^{\mathbf{x}(u)_j}} \frac{\partial}{\partial u} \mathbf{x}(u)_k \\ &= \nabla_u \mathbf{x}(u)_i - \sum_{k=1}^n S(\mathbf{x}(u))_k \cdot \nabla_u \mathbf{x}(u)_k \\ &= \nabla_u \mathbf{x}(u)_i - \mathbb{E}_j \left[\nabla_u \mathbf{x}(u)_j \right] \end{aligned}$$

m)

$$\begin{aligned}\nabla_u L(x,c) &= \nabla_u \sum_{i=1}^{K} -c_i \log y_i \\ &= -\nabla_u \log y_k \quad \text{with } c_k = 1 \\ &= -\nabla_u x_k + \mathbb{E}_j \nabla_u x_j \\ &= \mathbb{E}_j \nabla_u x_j - \nabla_u x_k \\ \nabla_u L(x,c) &= \sum_{j=1}^{k} y_j \nabla_u x_j - \nabla_u x_k\end{aligned}$$

Question 2

A) The dimension of $\mathbf{b}^{(1)}$ a vector of dimension $d_h \times 1$ which is equal to the number of neurons of the hidden layer.

The formula for the pre-activation vector can be expressed as (matrix form) - (where $\mathbf{P^{(0)}}(\mathbf{x})$ is the post activation of the first (input layer) layer) :

$$\mathbf{h}^a = \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}$$

To calculate the specific element $\mathbf{h}^a_j$, we get :

$$\mathbf{h}^a_j = \mathbf{b}^{(1)}_j + \sum_i \mathbf{W}^{(1)}_{j,i} x_i$$

And

$$\mathbf{h}^s = \phi(\mathbf{h}^a)$$

B) The dimension of $\mathbf{W^{(2)}}$ is of dimension $m \times d_h$, which is the dimension of the output layer times the dimension of the hidden layer. The dimension of $\mathbf{b^{(2)}}$ is a vector of size $m \times 1$.
The output layer before activation (pre-activation) is calculated (in matrix form) as :

$$\mathbf{o}^a = \mathbf{W}^{(2)}\mathbf{h}^s + \mathbf{b}^{(2)}$$

We can also write it in detailed form for $\mathbf{o}^a_k$ as :

$$\mathbf{o}^a_k = b^{(2)}_k + \sum_j \mathbf{W}^{(2)}_{k,j} h^s_j$$

C) We can write o_k^s as a function of o_k^a and o_j^a.

$$\begin{aligned} o_k^s &= \text{softmax}(\mathbf{o}^a)_k \\ &= \frac{\exp\left(o_k^a\right)}{\sum_{j=1}^m \exp\left(o_j^a\right)} \end{aligned}$$

All o_k^s are positive because the exponential function is stricly positive : $e^x > 0 \quad \forall x \in \mathbb{R}$. The softmax() function is a quotient of a exponential function (positive) by a sum of exponential functions (sum of positive values results in a positive value). The quotient of two positive values is also positive. Now let's prove that $\sum_k o_k^s = 1$.

$$\begin{aligned} \sum_k o_k^s &= \sum_k \text{softmax}(\mathbf{o}^a)_k \\ &= \sum_k \frac{\exp\left(o_k^a\right)}{\sum_j \exp\left(o_j^a\right)} \\ &= \frac{\exp\left(o_1^a\right)}{\sum_j \exp\left(o_j^a\right)} + \frac{\exp\left(o_2^a\right)}{\sum_j \exp\left(o_j^a\right)} + \dots \frac{\exp\left(o_m^a\right)}{\sum_j \exp\left(o_j^a\right)} \\ &= \frac{\sum_j \exp\left(o_j^a\right)}{\sum_j \exp\left(o_j^a\right)} \\ &= 1 \end{aligned}$$

The fact that $\sum_k o_k^s = 1$ is useful because it acts as a probability distribution over the different possible outcomes. By probability theory the sum of a discrete probability function must equals to 1.

D)

Let $\boldsymbol{x}$ be a 2-dimensional vector. One can represent a 2-class categorical probability using softmax :

$$\begin{aligned} P\left(\boldsymbol{x}_1\right) = S\left(\boldsymbol{x}_1\right) &= \frac{e^{x_1}}{e^{x_1} + e^{x_2}} = \frac{1}{1 + e^{x_2 - x_1}} \\ P\left(\boldsymbol{x}_2\right) = S\left(\boldsymbol{x}_2\right) &= \frac{e^{\boldsymbol{x}_2}}{e^{x_1} + e^{x_2}} = \frac{e^{\boldsymbol{x}_2 - \boldsymbol{x}_1}}{1 + e^{x_2 - x_1}} \end{aligned}$$

If we denote z as a scalar function of $\boldsymbol{x}$, i.e. $z = \boldsymbol{x}_2 - \boldsymbol{x}_1$, then

$$S\left(\boldsymbol{x}_1\right) = \frac{1}{1 + e^z} = \sigma(z)$$

$$S\left(\boldsymbol{x}_2\right) = \frac{e^z}{1 + e^z} = 1 - \sigma(z)$$

Hence $S(\boldsymbol{x}) = [\sigma(z), 1 - \sigma(z)]^\top$, where $z = \boldsymbol{x}_2 - \boldsymbol{x}_1$

E)

$$L_{MSE}\left(\sigma\left(\mathbf{o}^{a}\right),y\right)=\frac{1}{n}\sum_{i=1}^{n}\left(\sigma\left(\mathbf{o}_{i}^{a}\right)-y^{(i)}\right)^{2}$$

$$\begin{aligned}\frac{\partial L_{MSE}\left(\sigma\left(\mathbf{o}^{a}\right),y\right)}{\partial\mathbf{o}^{a}}&=\frac{\partial\left(\frac{1}{n}\sum_{i=1}^{n}\left(\sigma\left(\mathbf{o}^{a(i)}\right)-y^{(i)}\right)^{2}\right)}{\partial\mathbf{o}^{a}}\\&=\frac{1}{n}\sum_{i=1}^{n}\frac{\partial\left(\left(\sigma\left(\mathbf{o}^{a(i)}\right)-y^{(i)}\right)^{2}\right)}{\partial\mathbf{o}^{a(i)}}\\&=\frac{1}{n}\sum_{i=1}^{n}\frac{\partial\left(\sigma\left(\mathbf{o}^{a(i)}\right)-y^{(i)}\right)^{2}}{\partial\left(\sigma\left(\mathbf{o}^{a(i)}\right)-y^{(i)}\right)}\frac{\partial\left(\sigma\left(\mathbf{o}^{a(i)}\right)-y^{(i)}\right)}{\partial\sigma\left(\mathbf{o}^{a(i)}\right)}\frac{\partial\sigma\left(\mathbf{o}^{a(i)}\right)}{\partial\mathbf{o}^{a(i)}}\\&=\frac{1}{n}\sum_{i=1}^{n}2\left(\sigma\left(\mathbf{o}^{a(i)}\right)-y^{(i)}\right)\frac{\partial\sigma\left(\mathbf{o}^{a(i)}\right)}{\partial\mathbf{o}^{a(i)}}\end{aligned}$$

And :

$$\begin{aligned}\frac{\partial\sigma\left(\mathbf{o}^{a(i)}\right)}{\partial\mathbf{o}^{a(i)}}&=\partial\frac{\frac{1}{1+\exp\left(-\mathbf{o}^{a(i)}\right)}}{\partial\mathbf{o}^{a(i)}}\\&=\frac{\exp\left(-\mathbf{o}^{a(i)}\right)}{\left(1+\exp\left(-\mathbf{o}^{a(i)}\right)\right)^{2}}\\&=\frac{1}{1+\exp\left(-\mathbf{o}^{a(i)}\right)}\frac{\exp\left(-\mathbf{o}^{a(i)}\right)}{1+\exp\left(-\mathbf{o}^{a(i)}\right)}\\&=\frac{1}{1+\exp\left(-\mathbf{o}^{a(i)}\right)}\frac{1+\exp\left(-\mathbf{o}^{a(i)}\right)-1}{1+\exp\left(-\mathbf{o}^{a(i)}\right)}\\&=\sigma\left(\mathbf{o}^{a(i)}\right)\left(1-\sigma\left(\mathbf{o}^{a(i)}\right)\right)\end{aligned}$$

Finally :

$$\frac{\partial L_{MSE}\left(\sigma\left(\mathbf{o}^{a}\right),y\right)}{\partial\mathbf{o}^{a}}=\frac{1}{n}\sum_{i=1}^{n}2\left(\sigma\left(\mathbf{o}^{a(i)}\right)-y^{(i)}\right)\sigma\left(\mathbf{o}^{a(i)}\right)\left(1-\sigma\left(\mathbf{o}^{a(i)}\right)\right)$$

F)

$$L_{CE}\left(\sigma\left(\mathbf{o}^{a}\right),y\right)=\sum_{i=1}^{n}-\left(y^{(i)}\log\left(\sigma\left(\mathbf{o}^{a(i)}\right)\right)+\left(1-y^{(i)}\right)\log\left(1-\sigma\left(\mathbf{o}^{a(i)}\right)\right)\right)$$

$$
\begin{aligned}
\frac{\partial L_{CE}\left(\sigma\left(\mathbf{o}^{a}\right), y\right)}{\partial \mathbf{o}^{a}} &= \frac{\partial\left(\sum_{i=1}^{n}-\left(y^{(i)} \log\left(\sigma\left(\mathbf{o}^{a(i)}\right)\right)+\left(1-y^{(i)}\right) \log\left(1-\sigma\left(\mathbf{o}^{a(i)}\right)\right)\right)\right)}{\partial \mathbf{o}^{a}} \\
&= \sum_{i=1}^{n} \frac{\partial\left(-\left(y^{(i)} \log\left(\sigma\left(\mathbf{o}^{a(i)}\right)\right)+\left(1-y^{(i)}\right) \log\left(1-\sigma\left(\mathbf{o}^{a(i)}\right)\right)\right)\right)}{\partial \mathbf{o}^{a(i)}} \\
&= \sum_{i=1}^{n} \frac{\partial\left(-y^{(i)} \log\left(\sigma\left(\mathbf{o}^{a(i)}\right)\right)-\left(1-y^{(i)}\right) \log\left(1-\sigma\left(\mathbf{o}^{a(i)}\right)\right)\right)}{\partial \mathbf{o}^{a(i)}} \\
&= \sum_{i=1}^{n} \frac{\partial\left(-y^{(i)} \log\left(\sigma\left(\mathbf{o}^{a(i)}\right)\right)\right)}{\partial \mathbf{o}^{a(i)}}-\frac{\partial\left(\left(1-y^{(i)}\right) \log\left(1-\sigma\left(\mathbf{o}^{a(i)}\right)\right)\right)}{\partial \mathbf{o}^{a(i)}} \\
&= \sum_{i=1}^{n}\left(-y^{(i)}\left(1-\sigma\left(\mathbf{o}^{a(i)}\right)\right)+\left(1-y^{(i)}\right) \sigma\left(\mathbf{o}^{a(i)}\right)\right) \\
&= \sum_{i=1}^{n}\left(-y^{(i)}+\sigma\left(\mathbf{o}^{a(i)}\right) y^{(i)}+\sigma\left(\mathbf{o}^{a(i)}\right)-y^{(i)} \sigma\left(\mathbf{o}^{a(i)}\right)\right) \\
&= \sum_{i=1}^{n}\left(\sigma\left(\mathbf{o}^{a(i)}\right)-y^{(i)}\right)
\end{aligned}
$$

G) The derivative of the MSE is equivalent to the derivative of CE times $\sigma(1-\sigma)$. This term does not depend on the target, and vanishes when the output σ is near 0 or 1. In other words, when the predicted probability is around the extreme values (0 and 1), and especially when it does not correspond to the desired target, the MSE gradient will be very small which will hinder and considerably limit any useful correction towards the right target, corrections reflected in the term $\sigma - y$. Hence, the CE is a more appropriate choice for binary classification since it does not exhibit the same issue. H)

$$
L(x, y) = -\log o_y^s(x) = -\log\left(\frac{e^{o_y^a}}{\sum_{i=1} d_o e^{o_i^a}}\right) = -o_y^a + \log(\sum_{i=1}^{d_o} e^{o_i^a})
$$

I)

$$
\hat{R}(\theta) = \sum_{D} L(x^{(i)}, y^{(i)})
$$

The parameters θ are the two sets of W and b connecting the input layer to the hidden layer and the hidden layer to the output layer. In total, there are $n_{theta} = d_h d + d_h + d_o d_h + d_o$. The optimization problem is

$$
\arg\min_{\theta} \hat{R}(\theta)
$$

$$
\hat{R}(\theta) = \sum_{D} L(x^{(i)}, y^{(i)})
$$

The parameters θ are the two sets of W and b connecting the input layer to the hidden layer and the hidden layer to the output layer. In total, there are $n_{theta} = d_h d + d_h + d_o d_h + d_o$. The optimization problem is

$$\arg\min_\theta \hat{R}(\theta)$$

J)

$$\theta_{t+1} = \theta_t - \eta \nabla \hat{R}(\theta)$$

K)

$$\theta_{t+1} = \theta_t - \eta[\nabla \hat{R}(\theta) + 2\lambda\theta]$$

L) For this problem, we have to show that :

$$\frac{\partial L}{\partial \mathbf{o}^a} = \mathbf{o}^s - \text{onehot}_m(y)$$

Using the expression of L as a function of $\mathbf{o}_k^a$ We have :

$$L(\mathbf{o}^a, y) = -log(e^{\mathbf{o}_y^a}) + log(\sum_j e^{\mathbf{o}_j^a}) = log(\sum_j e^{\mathbf{o}_j^a}) - \mathbf{o}_y^a$$

When $k \neq y$:

$$\frac{\partial L}{\partial \mathbf{o_k^a}} = \frac{1}{\sum_j e^{o_j^a}} \frac{\partial \sum_j e^{o_j^a}}{\partial \mathbf{o_k^a}} - 0$$

$$= \frac{e^{o_k^a}}{\sum_j e^{o_j^a}} = o_k^s$$

When $k = y$:

$$\frac{\partial L}{\partial \mathbf{o_k^a}} = \frac{1}{\sum_j e^{o_j^a}} \frac{\partial \sum_j e^{o_j^a}}{\partial \mathbf{o_k^a}} - \frac{\partial o_y^a}{\partial o_y^a}$$

$$= \frac{e^{o_k^a}}{\sum_j e^{o_j^a}} - 1 = o_k^s - 1$$

Therefore, we can see that there is a minus one only when $k = y$, which means we can express :

$$\frac{\partial L}{\partial \mathbf{o}^a} = \mathbf{o}^s - \text{onehot}_m(y)$$

M)

$$\frac{\partial L}{\partial \mathbf{W}_{kj}^{(2)}} = \frac{\partial L}{\partial \mathbf{o}_k^a} \mathbf{h_j^s}$$

$$\frac{\partial L}{\partial \mathbf{b}_k^{(2)}} = \frac{\partial L}{\partial \mathbf{o}_k^a}$$

N)

$$\frac{\partial L}{\partial \mathbf{W}^{(2)}} = \frac{\partial L}{\partial \mathbf{o}^a}(\mathbf{h^s})^T$$

$$\frac{\partial L}{\partial \mathbf{b}^{(2)}} = \frac{\partial L}{\partial \mathbf{o}^a}$$

where $\frac{\partial L}{\partial \mathbf{o}^a}$ is a $d_o \times 1$ vector and h^s is a $d_h \times 1$ vector.

```
grad\_W2 = grad\_oa * hs.T \\
grad\_b2 = grad\_oa
```

O)
The partial derivative of the loss L with respect to the ouput of the neurons at the hidden layer is :

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{h}_j^s} &= \sum_{k=1}^{m} \frac{\partial L}{\partial \mathbf{o}_k^a} \frac{\partial \mathbf{o}_k^a}{\partial \mathbf{h}_j^s} \\
&= \sum_{k=1}^{m} \frac{\partial L}{\partial \mathbf{o}_k^a} \frac{\partial \left(b_k^{(2)} + \sum_j \mathbf{W}_{k,j}^{(2)} h_j^s \right)}{\partial \mathbf{h}_j^s} \\
&= \sum_{k=1}^{m} \frac{\partial L}{\partial \mathbf{o}_k^a} \mathbf{W}_{k,j}^{(2)}
\end{aligned}$$

P)

$$\frac{\partial L}{\partial \mathbf{h}_j^s} = (\mathbf{W}^{(2)})^T \frac{\partial L}{\partial \mathbf{o}^a}$$

```
grad_hs = w2.T * grad_oa
```

where the dimension of $(\mathbf{W}^{(2)})^T$ is of size $d_h \times m$, $\frac{\partial L}{\partial \mathbf{o}^a}$ is also a vector of size m $\times$ 1 and $\frac{\partial L}{\partial \mathbf{h}^s}$ is a vector of size $d_h \times 1$.

Q)

Now, calculating the partial derivative with respect to the. activation of the neurons at the hidden layer :

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{h}_j^a} &= \frac{\partial L}{\partial \mathbf{h}_j^s}\frac{\partial \mathbf{h}_j^s}{\partial \mathbf{h}_j^a} \\ &= \frac{\partial L}{\partial \mathbf{h}_j^s}\frac{\partial \phi(\mathbf{h}_j^a)}{\partial \mathbf{h}_j^a} \\ &= \frac{\partial L}{\partial \mathbf{h}_j^s}\sigma(\mathbf{h}_j^a)\end{aligned}$$

R) The gradient of the last equation can be written as :

$$\frac{\partial L}{\partial \mathbf{h}^a} = \frac{\partial L}{\partial \mathbf{h}^s}.diag(\sigma(\mathbf{h}^a))$$

The dimension of $\frac{\partial L}{\partial \mathbf{h}^s}$ is a vector of size $d_h \times 1$, (calcualted above).
The dimension of $\frac{\partial L}{\partial \mathbf{h}^a}$ is a vector of size $d_h \times 1$.
The dimension of $diag(\sigma(\mathbf{h}^a))$ is a matrix of size $d_h \times d_h$.

Question 3

a) Using the formula :

$$\text{out} = \left\lfloor \frac{\text{in} + 2p - d(k-1) - 1}{s} \right\rfloor + 1$$

Input layer :
$3 \times 128 \times 128$

Output 1st layer :
out - 1 : $\mathbf{32} \times \mathbf{64} \times \mathbf{64}$

Output 2nd layer :
out - 2 : $\mathbf{32} \times \mathbf{32} \times \mathbf{32}$

Output 3rd layer :
out - 3 : $\mathbf{32} \times \mathbf{64} \times \mathbf{64}$

b) the total number of parameters needed for last layer = $3 \times 3 \times 32 \times 64 =$ 18432

c)

$$\begin{aligned}[1,1,4,4,4,1,1] *_v [1/4,1/4,1/4] &= [1,1,4,4,4,1,1] * [1/4,1/4,1/4] \\ &= [3/2, 9/4, 3, 9/4, 3/2].\end{aligned}$$

$$\begin{aligned}[1,1,4,4,4,1,1] *_c [1/4,1/4,1/4] &= [0,0,1,1,4,4,4,1,1,0,0] * [1/4,1/4,1/4] \\ &= [1/4, 1/2, 3/2, 9/4, 3, 9/4, 3/2, 1/2, 1/4].\end{aligned}$$

$$\begin{aligned}[1,1,4,4,4,1,1] *_s [1/4,1/4,1/4] &= [0,1,1,4,4,4,1,1,0] * [1/4,1/4,1/4] \\ &= [1/2, 3/2, 9/4, 3, 9/4, 3/2, 1/2].\end{aligned}$$

d) The previous kernel smoothes the input : there are no significant jumps after the application of this convolution.