Devoir 2 et Devoir 3

Fondements de l'apprentissage machine - Aaron Courville

Boumedienne Boukharouba, Farzin Faridfar

December 4, 2017

1 PARTIE THÉORIQUE A

1.1

$$sigmoid(x) = \frac{1}{2}\big(tanh(\frac{1}{2}x) + 1\big)$$

$$\begin{aligned}
\frac{1}{2}\big(tanh(\frac{1}{2}x) + 1\big) &= \frac{1}{2}\big(\frac{e^{x/2} - e^{-x/2}}{e^{x/2} + e^{-x/2}} + 1\big) \\
&= \frac{1}{2}\big(\frac{e^{x/2} - e^{-x/2}}{e^{x/2} + e^{-x/2}} + \frac{e^{x/2} + e^{-x/2}}{e^{x/2} + e^{-x/2}}\big) \\
&= \frac{1}{2}\big(\frac{e^{x/2} - e^{-x/2} + e^{x/2} + e^{-x/2}}{e^{x/2} + e^{-x/2}}\big) \\
&= \frac{1}{2}\big(\frac{2e^{x/2}}{e^{x/2} + e^{-x/2}}\big) \\
&= \big(\frac{e^{x/2}}{e^{x/2} + e^{-x/2}}\big) \\
&= \big(\frac{e^{-x/2}}{e^{-x/2}} \times \frac{e^{x/2}}{e^{x/2} + e^{-x/2}}\big) \\
&= \big(\frac{1}{1 + e^{-x}}\big) = sigmoid(x) \quad \blacksquare
\end{aligned}$$

1.2

$$\ln(sigmoid(x)) = -softplus(-x)$$

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

1.3

$sigmoid'(x) = sigmoid(x)\big(1 - sigmoid(x)\big)$

$$\begin{aligned}
\frac{\partial}{\partial x}\big(\frac{1}{1+e^{-x}}\big) &= \frac{-e^{-x}}{(1+e^{-x})^2} \\
&= \frac{e^{-x}}{(1+e^{-x})^2} \\
&= \frac{1-1+e^{-x}}{(1+e^{-x})^2} \\
&= \frac{1+e^{-x}}{(1+e^{-x})^2} - \frac{1}{(1+e^{-x})^2} \\
&= \frac{1}{(1+e^{-x})} - (sigmoid(x))^2 \\
&= sigmoid(x) - (sigmoid(x))^2 \\
&= sigmoid(x)\big(1 - sigmoid(x)\big) \quad \blacksquare
\end{aligned}$$

1.4

$tanh'(x) = 1 - tanh^2(x)$

On a: $\big(\frac{f}{g}\big)' = \frac{f'.g - f.g'}{g^2}$

$$\begin{aligned}
\frac{\partial}{\partial x}\big(\frac{e^x - e^{-x}}{e^x + e^{-x}}\big) &= \frac{(e^x + e^{-x})(e^x + e^{-x}) - (e^x - e^{-x})(e^x - e^{-x})}{(e^x + e^{-x})^2} \\
&= \frac{(e^x + e^{-x})^2 - (e^x - e^{-x})^2}{(e^x + e^{-x})^2} \\
&= \frac{(e^x + e^{-x})^2}{(e^x + e^{-x})^2} - \frac{(e^x - e^{-x})^2}{(e^x + e^{-x})^2} = 1 - tanh^2(x) \quad \blacksquare
\end{aligned}$$

1.5

$$sign(x) = (1_{f(x)\geq 0} - 1_{f(x)\leq 0}) \quad \blacksquare$$

1.6

$$abs(x) = \begin{cases} x, & x \geq 0 \\ -x, & \text{sinon} \end{cases}$$

$$abs'(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}$$

$$abs'(x) = sign(x) \quad \blacksquare$$

1.7

$$rect(x) = (1_{x>0})x = \begin{cases} x, & x \geq 0 \\ 0, & \text{sinon} \end{cases}$$

$$rect'(x) = \begin{cases} 1, & x > 0 \\ 0, & \text{sinon} \end{cases}$$

$$rect'(x) = 1_{x>0}(x) \quad \blacksquare$$

1.8

$$\|x\|_2^2 = \sum_i x_i^2$$

$$\frac{\partial \|x\|_2^2}{\partial x} = \frac{\partial}{\partial x} \sum_i x_i^2 = \sum_i \frac{\partial}{\partial x_i} x_i^2 = \sum_i 2x_i = 2 \sum_i x_i \quad \blacksquare$$

1.9

$$\|x\|_1 = \sum_i |x_i|$$

$$\frac{\partial \|x\|_1}{\partial x} = \frac{\partial}{\partial x} \sum_i |x_i| = \sum_i \frac{\partial}{\partial x_i} |x_i| = \sum_i sign(x_i) \quad \blacksquare$$

2 PARTIE THÉORIQUE B

2.1

$b^{(1)} \in \mathbb{R}^{\mathrm{d_h}}$.

$$w^{(1)} = \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} & \dots & w_{1d}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} & \dots & w_{2d}^{(1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{d1}^{(1)} & w_{d2}^{(1)} & w_{d3}^{(1)} & \dots & w_{d_h d}^{(1)} \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}, b^{(1)} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(1)} \\ \vdots \\ b_{d_h}^{(1)} \end{bmatrix} \quad \blacksquare$$

Expression matricielle: $\quad ha = \left\langle w^{(1)}.x \right\rangle + b^{(1)} \quad \blacksquare$

Expression élémentaire:

$$ha = \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} & \dots & w_{1d}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} & \dots & w_{2d}^{(1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{d1}^{(1)} & w_{d2}^{(1)} & w_{d3}^{(1)} & \dots & w_{d_h d}^{(1)} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} + \begin{bmatrix} b_1^{(1)} \\ b_2^{(1)} \\ \vdots \\ b_{d_h}^{(1)} \end{bmatrix} = \begin{bmatrix} \sum_i^d w_{1i}^{(1)}.x_i + b_1^{(1)} \\ \sum_i^d w_{2i}^{(1)}.x_i + b_2^{(1)} \\ \vdots \\ \sum_i^d w_{d_h i}^{(1)}.x_i + b_{d_h}^{(1)} \end{bmatrix}$$

$$\rightarrow ha_j = \sum_i^d w_{ji}^{(1)}.x_i + b_j^{(1)} \quad \blacksquare$$

Expression matricielle: $\quad hs = RELU(ha(x)) \quad \blacksquare$

Expression élémentaire:

$$hs = RELU\left(\begin{bmatrix} \sum_j^d w_{1j}^{(1)}.x_j + b_1^{(1)} \\ \sum_j^d w_{2j}^{(1)}.x_j + b_2^{(1)} \\ \vdots \\ \sum_j^d w_{d_h j}^{(1)}.x_j + b_{d_h}^{(1)} \end{bmatrix} \right) = \begin{bmatrix} max(0, \sum_j^d w_{1j}^{(1)}.x_j + b_1^{(1)}) \\ max(0, \sum_j^d w_{2j}^{(1)}.x_j + b_2^{(1)}) \\ \vdots \\ max(0, \sum_j^d w_{d_h j}^{(1)}.x_j + b_{d_h}^{(1)}) \end{bmatrix}$$

$$\rightarrow hs_j = max(0, \sum_i^d w_{ji}^{(1)}.x_i + b_j^{(1)}) \quad \blacksquare$$

$ha \in \mathbb{R}^{\mathrm{d_h}}$, $hs \in \mathbb{R}^{\mathrm{d_h}}$.

2.2

$w^{(2)} \in \mathbb{R}^{\mathrm{m} \times \mathrm{d_h}}, \mathrm{b}^{(2)} \in \mathbb{R}^{\mathrm{m}}$.

Expression matricielle: $\quad oa = \left\langle w^{(2)}.hs \right\rangle + b^{(2)} \quad \blacksquare$

Expression élémentaire:

$$oa = \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} & \dots & w_{1d_h}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} & \dots & w_{2d_h}^{(1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{m1}^{(1)} & w_{m2}^{(1)} & w_{m3}^{(1)} & \dots & w_{md_h}^{(1)} \end{bmatrix} . \begin{bmatrix} hs_1 \\ hs_2 \\ \vdots \\ hs_{d_h} \end{bmatrix} + \begin{bmatrix} b_1^{(2)} \\ b_2^{(2)} \\ \vdots \\ b_m^{(2)} \end{bmatrix} = \begin{bmatrix} \sum_i^{d_h} w_{1i}^{(1)}.hs_i + b_1^{(1)} \\ \sum_i^{d_h} w_{2i}^{(1)}.hs_i + b_2^{(1)} \\ \vdots \\ \sum_i^{d_h} w_{mi}^{(1)}.hs_i + b_m^{(1)} \end{bmatrix}$$

$$\rightarrow oa_k = \sum_i^{d_h} \left(w_{ki}^{(2)}.hs_i \right) + b_k^{(2)} \quad \blacksquare$$

$oa \in \mathbb{R}^{\mathrm{m}}$.

2.3

Expression matricielle:

$$os = softmax(oa) \quad \blacksquare$$

$os \in \mathbb{R}^{\mathrm{m}}$.
Expression élémentaire:

$$os_k = \frac{e^{oa_k}}{\sum_i^m e^{oa_i}} \quad \blacksquare$$

La fonction exponentielle est une fonction positive. Par conséquente, le résultat de $softmax$ est toujours positif.

$$\begin{aligned} \sum_k^m os_k = \sum_k^m softmax(oa_k) &= \sum_k^m \left(\frac{e^{oa_k}}{\sum_i^m e^{oa_i}} \right) \\ &= \frac{e^{oa_1}}{\sum_i^m e^{oa_i}} + \frac{e^{oa_2}}{\sum_i^m e^{oa_i}} + \ldots + \frac{e^{oa_m}}{\sum_i^m e^{oa_i}} \\ &= \frac{e^{oa_1} + e^{oa_2} + \ldots + e^{oa_m}}{\sum_i^m e^{oa_i}} \\ &= \frac{\sum_i^m e^{oa_i}}{\sum_i^m e^{oa_i}} = 1 \quad \blacksquare \end{aligned}$$

Comme les os_k sont positifs et somment à 1 ($os_k \in [0, 1]$), on peut les interpréter comme la probabilité qu'un exemple x, dans l'ensemble de donnée, appartient à une classe k. Ainsi, on pourrait utiliser le résultat de $softmax$ pour classifier les données pour une classification multi-classe.

2.4

$$\begin{aligned} L(x, y) = -\log os_y(x) &= -\log softmax(oa_y) \\ &= -\log \left(\frac{e^{oa_y}}{\sum_i^m e^{oa_i}} \right) \quad \blacksquare \end{aligned}$$

2.5

Risque empirique:

$$\begin{aligned}\widehat{R} &= \frac{1}{n}\sum_{t=1}^{n} L(x^{(t)}, y^{(t)}) \\ &= -\frac{1}{n}\sum_{t=1}^{n} \log\left(\frac{e^{oa_y}}{\sum_i^m e^{oa_i}}\right) \quad \blacksquare\end{aligned}$$

$$\begin{aligned}\theta &= \{w^{(1)}, b^{(1)}, w^{(2)}, b^{(2)}\} \\ n_{w^{(1)}} &= d_h \times d,\ n_{b^{(1)}} = d_h \\ n_{w^{(2)}} &= m \times d_h,\ n_{b^{(2)}} = m \\ \rightarrow n_\theta &= d_h \times d + d_h + m \times d_h + m\end{aligned}$$

Pour chacun des paramètres, on utilise la méthode de descente de gradient pour trouver la valeur qui minimise le risque. On met à jour les paramètres sur l'ensemble d'entraînement et pour un nombre maximal d'itérations:

$$\begin{aligned}\underset{\theta}{argmin}\widehat{R} &= \underset{\theta}{argmin}\frac{1}{n}\sum_{t=1}^{n} L(x^{(t)}, y^{(t)}) \\ &= -\underset{\theta}{argmin}\frac{1}{n}\sum_{t=1}^{n} \log\left(\frac{e^{oa_y}}{\sum_i^m e^{oa_i}}\right) \quad \blacksquare\end{aligned}$$

$$\frac{\partial \widehat{R}}{\partial \theta} = 0 \text{ (Analytique)}$$

$$\theta \leftarrow \theta - \eta\frac{\partial \widehat{R}}{\partial \theta} \text{ (Descente de gradient)}$$

Où η est le taux d'apprentissage. Ensuite en utilisant les paramètres optimisés, on fait la prédiction pour l'ensemble de test avec notre technique linéaire/non-linéaire choisie.

2.6

Data: Les paramètres $\theta = \{w^{(1)}, b^{(1)}, w^{(2)}, b^{(2)}\}$
Result: Les paramètres θ optimisés
Initialiser les paramètres θ;
for *N epochs* **do**
  for *chaque batch de K exemples* **do**
    for *chaque exemple de données x et y dans ce batch* **do**

$$\frac{\partial L}{\partial \theta} \leftarrow \frac{\partial}{\partial \theta} L(x, y) + \lambda \frac{\partial \phi}{\partial \theta}$$
$$\theta \leftarrow \theta - \eta \frac{\partial L}{\partial \theta}$$

    end
  end
end

Algorithm 1: technique de descente de gradient pour minimiser le risque empirique

2.7

On a:

$$L(x, y) = -\log os_y(x) = -\log softmax(oa_y) = -\log\Big(\frac{e^{oa_y}}{\sum_i^m e^{oa_i}}\Big) \quad \blacksquare$$

Pour la simplification on remplace $\sum_i^m e^{oa_i}$ par $\sum$, ainsi on aura:

$$\frac{\partial L}{\partial oa_k} = -\frac{1}{\frac{e^{oa_y}}{\sum}}\Big(\frac{\partial}{\partial oa_k}\frac{e^{oa_y}}{\sum}\Big)$$

Nous avons:

$$(\frac{f}{g})' = \frac{f'.g - f.g'}{g^2}, \quad \frac{f}{g} = \frac{e^{oa_y}}{\sum}$$
$$f' = \begin{cases} 0, & k \neq y \\ e^{oa_k}, & \text{sinon} \end{cases} \quad \blacksquare$$
$$g' = \frac{\partial}{\partial oa_k}\sum_i^m e^{oa_i} = e^{oa_k} \text{ pour tout i} \quad \blacksquare$$

Ainsi, pour le numérateur dans l'équation de dérive de la perte on aura:

$$\begin{aligned}
\frac{\partial}{\partial oa_k}\frac{e^{oa_y}}{\sum} &= \begin{cases} \frac{e^{oa_k}\sum - e^{oa_k}.e^{oa_k}}{\sum^2} & \text{, si k = y} \\ -\frac{e^{oa_y}.e^{oa_k}}{\sum^2} & \text{sinon} \end{cases} \\
&= \begin{cases} \frac{e^{oa_k}}{\sum}\frac{\sum}{\sum} - \frac{e^{oa_k}.e^{oa_k}}{\sum^2} & \text{, si k = y} \\ -\frac{e^{oa_y}}{\sum}.\frac{e^{oa_k}}{\sum} & \text{sinon} \end{cases} \\
&= \begin{cases} softmax(oa_k)(1 - softmax(oa_k)) & \text{,si k = y} \\ -softmax(oa_y).softmax(oa_k) & \text{,sinon} \end{cases} \quad \blacksquare
\end{aligned}$$

Maintenant on retourne au calcul de la dérivée de gradient de perte:

$$\begin{aligned}
\frac{\partial L}{\partial oa_k} &= -\frac{\frac{\partial}{\partial oa_k}\frac{e^{oa_y}}{\sum}}{\frac{e^{oa_k}}{\sum}} = -\frac{\frac{\partial}{\partial oa_k} softmax(oa_y)}{softmax(oa_y)} \\
&= \begin{cases} -\frac{softmax(oa_k)(1-softmax(oa_k))}{softmax(oa_k)} & \text{,si k = y} \\ \frac{softmax(oa_y).softmax(oa_k)}{softmax(oa_y)} & \text{,sinon} \end{cases} \\
&= \begin{cases} softmax(oa_k) - 1 & \text{,si k = y} \\ softmax(oa_k) & \text{,sinon} \end{cases} \\
&= \begin{cases} os_k - 1 & \text{,si k = y} \\ os_k & \text{,sinon} \end{cases} \quad \blacksquare
\end{aligned}$$

On sait que $onehot_m(y)$ et un vecteur de longueur m qui à 0 pour tous ces éléments sauf pour l'élément y qui est égal à 1. Si on regarde la dernière équation obtenue on peut voir que la dérivée de la perte est égal à o^s pour tout $k \neq y$ et seulement pour $k = y$ on fait la soustraction de 1. Par conséquente, nous sommes capable de convertir l'équation élémentaire obtenue dans la forme vectoriel:

$$\begin{aligned}
\frac{\partial L}{\partial oa} &= (os_1, os_2, \ldots, os_y, \ldots, os_m) - (0, 0, \ldots, 1, \ldots, 0) \\
&= os - onehot_m(y) \quad \blacksquare
\end{aligned}$$

$$\frac{\partial L}{\partial oa} \in \mathbb{R}^{\mathrm{m}}.$$

2.8

Pour un vecteur oa et un y correspondant on aura:

```
softmax = np.exp(oa)/np.sum(np.exp(oa))
onehot = np.ones((m,1))
onehot[y] = 1
grad_oa = softmax - onehot
```

2.9

$$\frac{\partial L}{\partial w_{kj}^{(2)}} = \frac{\partial L}{\partial oa_k}\frac{\partial oa_k}{\partial w_{kj}^{(2)}} \ , \ \frac{\partial L}{\partial b_k^{(2)}} = \frac{\partial L}{\partial oa_k}\frac{\partial oa_k}{\partial b_k^{(2)}} \ , \ \frac{\partial L}{\partial oa_k} = o^s - onehot_m(y) \quad \blacksquare$$

$$oa_k = \sum_j^{d_h} \left(w_{kj}^{(2)}.hs_j\right) + b_k^{(2)} \quad \blacksquare$$

$$\begin{aligned}\frac{\partial oa_k}{\partial w_{kj}^{(2)}} &= \frac{\partial}{\partial w_{kj}^{(2)}} \sum_j^{d_h} \left(w_{kj}^{(2)}.hs_j\right) + \frac{\partial}{\partial w_{kj}^{(2)}} b_k^{(2)} \\ &= \frac{\partial}{\partial w_{kj}^{(2)}} \sum_j^{d_h} \left(w_{kj}^{(2)}.hs_j\right) = hs_j \quad \blacksquare\end{aligned}$$

$$\frac{\partial oa_k}{\partial b_k^{(2)}} = \frac{\partial}{\partial b_k^{(2)}} \sum_j^{d_h} \left(w_{kj}^{(2)}.hs_j\right) + \frac{\partial}{\partial b_k^{(2)}} b_k^{(2)} = 1 \quad \blacksquare$$

Maintenant, on combine les dérivées enchaînée:

$$\begin{aligned}\frac{\partial L}{\partial w_{kj}^{(2)}} &= \frac{\partial L}{\partial oa_k}\frac{\partial oa_k}{\partial w_{kj}^{(2)}} \\ &= (os_k - onehot_m(k))(hs_j) \quad \blacksquare \\ \frac{\partial L}{\partial b_k^{(2)}} &= \frac{\partial L}{\partial oa_k}\frac{\partial oa_k}{\partial b_k^{(2)}} \\ &= (os_k - onehot_m(k)) \quad \blacksquare\end{aligned}$$

2.10

$$\begin{aligned}\frac{\partial L}{\partial w^{(2)}} &= \frac{\partial L}{\partial oa}\frac{\partial oa}{\partial w^{(2)}} \\ &= (os - onehot_m(k))(hs)^T \quad \blacksquare\end{aligned}$$

$$\begin{aligned}\frac{\partial L}{\partial b_k^{(2)}} &= \frac{\partial L}{\partial oa_k}\frac{\partial oa_k}{\partial b_k^{(2)}} \\ &= (os - onehot_m(k)) \quad \blacksquare\end{aligned}$$

$$\frac{\partial L}{\partial oa} \in I\!R^m, \quad hs \in I\!R^{d_h} \rightarrow \frac{\partial L}{\partial w^{(2)}} \in I\!R^{m \times d_h} \quad et \quad \frac{\partial L}{\partial b^{(2)}} \in I\!R^m.$$

```
grad_w2 = np.dot(grad_oa, hs.T)
grad_b2 = grad_oa
```

2.11

$$\frac{\partial L}{\partial hs_j} = \sum_k^m \frac{\partial L}{\partial oa_k} \frac{\partial oa_k}{\partial hs_j}\ ,\ \frac{\partial L}{\partial oa_k} = os_k - onehot_m(k) \quad \blacksquare$$

$$oa_k = \sum_i^{d_h} \left(w_{ki}^{(2)}.hs_i\right) + b_k^{(2)} \quad \blacksquare$$

$$\begin{aligned} \frac{\partial oa_k}{\partial hs_j} &= \frac{\partial}{\partial hs_j} \sum_i^{d_h} \left(w_{ki}^{(2)}.hs_i\right) + \frac{\partial}{\partial hs_j} b_k^{(2)} \\ &= w_{kj}^{(2)} \quad \blacksquare \end{aligned}$$

$$\begin{aligned} \frac{\partial L}{\partial hs_j} &= \sum_k^m \frac{\partial L}{\partial oa_k} \frac{\partial oa_k}{\partial hs_j} \\ &= \sum_k^m \left((os_k - onehot_m(k)).w_{kj}^{(2)} \right) \quad \blacksquare \end{aligned}$$

2.12

$$\frac{\partial L}{\partial hs} = \frac{\partial L}{\partial oa} \frac{\partial oa}{\partial hs} = (w^{(2)})^T . \frac{\partial L}{\partial oa} \quad \blacksquare$$

$$\frac{\partial L}{\partial oa} \in I\!R^m, \quad w^{(2)} \in I\!R^{m \times d_h} \rightarrow \frac{\partial L}{\partial hs} \in I\!R^{d_h}.$$

```
grad_hs = np.dot(w2.T, grad_oa)
```

2.13

$$\frac{\partial L}{\partial ha_j} = \frac{\partial L}{\partial hs_k}\frac{\partial hs_k}{\partial ha_j}\ ,\ \frac{\partial L}{\partial hs_k} = \sum_k^m \frac{\partial L}{\partial oa_k}.w_{kj}^{(2)} \quad \blacksquare$$

$$hs_j = rect(ha_j)$$

$$\frac{\partial hs_k}{\partial ha_j} = rect'(ha_j) = \begin{cases} 1, & ha_j > 0 \\ 0, & \text{sinon} \end{cases}$$

$$= 1_{(ha_j>0)} \text{ (la fonction indicatrice)} \quad \blacksquare$$

On combine les dérivées enchaînées:

$$\frac{\partial L}{\partial ha_j} = \begin{cases} \sum_k^m \frac{\partial L}{\partial oa_k}.w_{kj}^{(2)} & ha_j > 0 \\ 0 & \text{sinon} \end{cases}$$

$$= \Big(\sum_k^m \frac{\partial L}{\partial oa_k}.w_{kj}^{(2)}\Big).1_{(ha_j>0)} \quad \blacksquare$$

2.14

$$\frac{\partial L}{\partial ha} = \begin{cases} (w^{(2)})^T \frac{\partial L}{\partial oa} & ha > 0 \\ 0 & \text{sinon} \end{cases}$$

$$= \frac{\partial L}{\partial oa}.w^{(2)} * [1_{ha_1>0}, \ldots, 1_{ha_{dh}>0}]^T$$

$$= \frac{\partial L}{\partial hs} * [1_{ha_1>0}, \ldots, 1_{ha_{dh}>0}]^T$$

Où $*$ est le produit élémentaire entre les deux vecteurs.

$$\frac{\partial L}{\partial hs} \in I\!R^{d_h}, \quad 1_{(ha>0)} \in I\!R^{d_h} \rightarrow \frac{\partial L}{\partial ha} \in I\!R^{d_h}$$

```
grad_ha = np.where(self.ha > 0, 1 , 0)*grad_hs
```

2.15

$$\frac{\partial L}{\partial w_{kj}^{(1)}} = \frac{\partial L}{\partial ha_k}\frac{\partial ha_k}{\partial w_{kj}^{(1)}}\ ,\ \frac{\partial L}{\partial b_k^{(1)}} = \frac{\partial L}{\partial ha_k}\frac{\partial ha_k}{\partial b_k^{(1)}}$$

$$\frac{\partial L}{\partial ha_k} = \sum_{k=1}^m \frac{\partial L}{\partial oa_k}.w_{kj}^{(1)}.1_{(ha_j>0)}, \quad ha_k = \sum_{j=1}^n \left(w_{kj}^{(1)}.x_j\right) + b_k^{(1)} \quad \blacksquare$$

$$\frac{\partial ha_k}{\partial w_{kj}^{(1)}} = \frac{\partial}{\partial w_{kj}^{(1)}} \sum_{j=1}^{n} \left(w_{kj}^{(1)}.x_j\right) + \frac{\partial}{\partial w_{kj}^{(1)}} b_k^{(1)} = x_j$$
$$\rightarrow \frac{\partial L}{\partial w_{kj}^{(1)}} = \frac{\partial L}{\partial ha_k} x_j \quad \blacksquare$$

$$\frac{\partial ha_k}{\partial b_k^{(1)}} = \frac{\partial}{\partial b_k^{(1)}} \sum_{j=1}^{n} \left(w_{kj}^{(1)}.x_j\right) + \frac{\partial}{\partial b_k^{(1)}} b_k^{(1)} = 1$$
$$\rightarrow \frac{\partial L}{\partial b_k^{(1)}} = \frac{\partial L}{\partial ha_k} \quad \blacksquare$$

2.16

$$\frac{\partial L}{\partial w^{(1)}} = \frac{\partial L}{\partial ha}.x^T \quad \blacksquare$$
$$\frac{\partial L}{\partial b^{(1)}} = \frac{\partial L}{\partial ha} \quad \blacksquare$$
$$\frac{\partial L}{\partial w_{kj}^{(1)}} \in IR^{d_h \times d}, \frac{\partial L}{\partial b_k^{(1)}} \in IR^{d_h}.$$

```
grad_w1 = np.dot(grad_ha, x.T)
grad_b1 = grad_ha
```

2.17

Expression élémentaire:

$$\frac{\partial L}{\partial x_j} = \sum_{k}^{d_h} \frac{\partial L}{\partial ha_k} \frac{\partial ha_k}{\partial x_j} \ , \ ha_k = w_{kj}^{(1)} x_j + b_k$$

$$\frac{\partial ha_k}{\partial x_j} = \frac{\partial}{\partial x_j}\left(w_{kj}^{(1)} x_j + b_k\right) = w_{kj}^{(1)}$$

$$\frac{\partial L}{\partial x_j} = \sum_{k}^{d_h} \frac{\partial L}{\partial ha_k}.w_{kj}^{(1)} \quad \blacksquare$$

Expression matricielle:

$$\frac{\partial L}{\partial x_j} = (w^{(1)})^T.\frac{\partial L}{\partial ha} \quad \blacksquare$$

$$\frac{\partial L}{\partial ha} \in IR^{d_h}, w^{(1)} \in IR^{d_h \times d}, \frac{\partial L}{\partial x_j} \in IR^{d}.$$

2.18

L'addition des hyper-paramètes de la régularisation aura un effet sur les gradients de la fonction de perte par rapport aux paramètres de poids, $w1$ et $w2$. Ainsi, on aura:

$$\tilde{R} = \widehat{R} + \mathcal{L}(\theta)$$

$$\begin{aligned}
\frac{\partial \tilde{R}}{\partial w^{(l)}} &= \frac{\partial \widehat{R}}{\partial w^{(l)}} + \frac{\partial \mathcal{L}(\theta)}{\partial w^{(l)}}, \quad l \in \{1, 2\} \\
&= \frac{\partial \widehat{R}}{\partial w^{(l)}} + \frac{\partial}{\partial w^{(l)}} \Big(\sum_{l=1}^{2} (\lambda_{l1} \|w^{(l)}\| + \lambda_{l2} \|w^{(l)}\|^2) \Big) \\
&= \frac{\partial \widehat{R}}{\partial w^{(l)}} + \sum_{l=1}^{2} (\lambda_{l1} + 2\lambda_{l2} \|w^{(l)}\|) \\
&= \frac{1}{n} \sum_{i}^{n} \Big(\frac{\partial L_i}{\partial w^{(l)}} \Big) + \sum_{l=1}^{2} (\lambda_{l1} + 2\lambda_{l2} \|w^{(l)}\|)
\end{aligned}$$

Le calcul de $\frac{\partial L_i}{\partial w^{(l)}}$ est déjà fait dans les questions précédents.

3 PARTIE PRATIQUE

L'explication générale pour exécuter le code de la partie pratique:

1. Toutes les questions de cette partie sont répondues dans le fichier `main.py`. Vous pourriez exécuter ce fichier sur un terminal GNU/Linux avec la commande:
   `$ python3 main.py` (Vous pourriez trouver ce fichier dans le dossier *pratique* de fichier zip).

2. Les réponses de toutes les questions, sauf pour la question 9, seront affiché dans le terminal en exécutant la commande mentionnée au-dessus.

3. L'affichage de surfaces de décision demandée pour la questions 5 est en commentaire pour éviter la perte de temps pendant l'exécution. Les graphiques sont stockés dans le dossier *graphiques*. Cependant, pour voir les plots pour cette questions vous pourriez enlever la commentaire de la ligne 138 du fichier `main.py`.

4. Comme l'exécution de la question 9 prend beaucoup de temps, cette questions est mise en commentaire.

   (a) Les résultat d'exécution de cette questions est stocké dans le fichier `study_log.txt`.

(b) Si vous voulez exécuter le code de cette partie, vous pourriez sortir les codes des lignes 267 jusqu'à 284 du fichier `main.py` de commentaire.

(c) Aussi, pour qu'il prend moins de temps, vous pourriez diminuer la valeur du hyper-paramète `epoch` en ligne 273 du même fichier.

(d) Finalement, pour lire le fichier des données *MNIST*, nous étions obligé d'ajouter l'argument `encoding='latin-1'` dans la méthode `pickle.load()` (ligne 48 de fichier `main.py`) pour que la lecture se fait sur une machine Debian alors que sur Windows, cet argument donne une erreur. S'il vous plaît, si jamais dans l'exécution de cette question une erreur d'encoding s'affiche, vous devriez tout simplement, enlever cet argument.

```
data = pickle.load(f)       # (à la place de ligne dessous)
data = pickle.load(f, encoding='latin1')
```

5. Vous pourriez trouver les graphiques pour les question 5 et 10 dans le dossier *graphiques* dans le dossier *pratique* de fichier zip envoyé.