Devoir 2
Fondements de l'apprentissage machine IFT6390
Le lundi 4 décembre 2017

Noms : Léa-Marie Normandin, Zhibin Lu et Xiaocheng Liu

1 Partie théorique A (20 pts) : Relations et dérivées de quelques fonctions de base

1. On veut montrer que $\mathrm{sigmoid}(x) = \frac{1}{2}\left(\tanh\left(\frac{1}{2}x\right)+1\right)$.

$$\begin{aligned}
\frac{1}{2}\left(\tanh\left(\frac{1}{2}x\right)+1\right) &= \frac{1}{2}\left(\frac{\exp(\frac{1}{2}x)-\exp(-\frac{1}{2}x)}{\exp(\frac{1}{2}x)+\exp(-\frac{1}{2}x)}\right) \\
&= \frac{\exp(\frac{1}{2}x)-\exp(-\frac{1}{2}x)+\exp(\frac{1}{2}x)+\exp(-\frac{1}{2}x)}{\exp(\frac{1}{2}x)+\exp(-\frac{1}{2}x)} \\
&= \frac{\exp(\frac{1}{2}x)}{\exp(\frac{1}{2}x)+\exp(-\frac{1}{2}x)} \\
&= \frac{1}{1+\exp(-x)} \\
&= \mathrm{sigmoid}(x)
\end{aligned}$$

2. On doit montrer que $\ln \mathrm{sigmoid}(x) = -\mathrm{softplus}(-x)$.

$$\begin{aligned}
\ln \mathrm{sigmoid}(x) &= \ln \frac{1}{1+\exp(-x)} \\
&= -\ln(1+\exp(-x)) \\
&= -\mathrm{softplus}(-x)
\end{aligned}$$

3. On doit montrer que $\mathrm{sigmoid}'(x) = \mathrm{sigmoid}(x)\,(1-\mathrm{sigmoid}(x))$.

$$\mathrm{sigmoid}(x) = \frac{1}{1+\exp(-x)}$$

$$\begin{aligned}
\mathrm{sigmoid}'(x) &= -(1+\exp(-x))^{-2}(\exp(-x))(-1) \\
&= \frac{\exp(-x)}{(1+\exp(-x))^2} \\
&= \frac{1+\exp(-x)-1}{(1+\exp(-x))^2} \\
&= \frac{1}{1+\exp(-x)} - \frac{1}{(1+\exp(-x))^2} \\
&= \frac{1}{1+\exp(-x)}\left(1-\frac{1}{1+\exp(-x)}\right) \\
&= \mathrm{sigmoid}(x)\,(1-\mathrm{sigmoid}(x))
\end{aligned}$$

4. On doit montrer que $\tanh'(x) = 1-\tanh^2(x)$.

$$\begin{aligned}
\tanh'(x) &= \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}\right) \\
&= \frac{(\exp(x)+\exp(-x))^2-(\exp(x)-\exp(-x))^2}{(\exp(x)+\exp(-x))^2} \\
&= 1-\left(\frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}\right)^2 \\
&= 1-\tanh^2(x)
\end{aligned}$$

5. Écrivons la fonction $\text{sign}(x)$ en utilisant des fonctions indicatrices.

$$\text{sign}(x) = \mathbf{1}_{(0,+\infty)}(x) - \mathbf{1}_{(-\infty,0)}(x)$$

6. On cherche la dérivée de la fonction valeur absolue $\text{abs}(x)$.

$$\text{abs}(x) = x\mathbf{1}_{(0,+\infty)}(x) - x\mathbf{1}_{(-\infty,0)}(x)$$

$$\text{abs}'(x) = \mathbf{1}_{(0,+\infty)}(x) - \mathbf{1}_{(-\infty,0)}(x) = \text{sign}(x)$$

7. On cherche la dérivée de la fonction $\text{rect}(x)$.

$$\text{rect}(x) = x\mathbf{1}_{(0,\infty)}(x)$$

$$\text{rect}'(x) = \mathbf{1}_{(0,\infty)}(x)$$

8. Soit la norme L_2 d'un vecteur : $\|\boldsymbol{x}\|_2^2 = \sum_i \boldsymbol{x}_i^2$.

$$\begin{aligned}\frac{\partial\|\boldsymbol{x}\|_2^2}{\partial\boldsymbol{x}} &= \frac{\partial}{\partial\boldsymbol{x}}\sum_i \boldsymbol{x}_i^2 \\ &= \sum_i \frac{\partial}{\partial\boldsymbol{x}}\boldsymbol{x}_i^2\end{aligned}$$

On doit décomposer par composantes :

$$\begin{aligned}\frac{\partial\|\boldsymbol{x}\|_2^2}{\partial\boldsymbol{x}_j} &= \sum_i \frac{\partial}{\partial\boldsymbol{x}_j}\boldsymbol{x}_i^2 \\ &= 2\boldsymbol{x}_j\end{aligned}$$

Donc

$$\frac{\partial\|\boldsymbol{x}\|_2^2}{\partial\boldsymbol{x}} = (2\boldsymbol{x}_1, 2\boldsymbol{x}_2, \ldots, 2\boldsymbol{x}_d)$$

9. Soit la norme L_1 d'un vecteur : $\|\boldsymbol{x}\|_1 = \sum_i |\boldsymbol{x}_i|$

$$\begin{aligned}\frac{\partial\|\boldsymbol{x}\|_1}{\partial\boldsymbol{x}} &= \frac{\partial}{\partial\boldsymbol{x}}\sum_i |\boldsymbol{x}_i| \\ &= \sum_i \frac{\partial}{\partial\boldsymbol{x}}|\boldsymbol{x}_i|\end{aligned}$$

On doit décomposer par composantes :

$$\begin{aligned}\frac{\partial\|\boldsymbol{x}\|_1}{\partial\boldsymbol{x}_j} &= \sum_i \frac{\partial}{\partial\boldsymbol{x}_j}|\boldsymbol{x}_i| \\ &= \text{sign}(\boldsymbol{x}_j)\end{aligned}$$

Donc

$$\frac{\partial\|\boldsymbol{x}\|_1}{\partial\boldsymbol{x}} = (\text{sign}(\boldsymbol{x}_1), \text{sign}(\boldsymbol{x}_2), \ldots, \text{sign}(\boldsymbol{x}_d))$$

2 Partie théorique B (40 pts) : Calcul du gradient pour l'optimisation des paramètres d'un réseau de neurones pour la classification multiclasse

On a un ensemble de données $\mathrm{D}_n = \{(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \ldots, (\boldsymbol{x}^{(n)}, y^{(n)})\}$ avec $y^{(i)} \in \{1, \ldots, m\}$ et

$$\boldsymbol{x}^{(i)} = \begin{pmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_d^{(i)} \end{pmatrix} \in \mathbb{R}^d$$

1. Soit $\mathbf{W}^{(1)}$ la matrice $d_h \times d$ de poids

$$\mathbf{W}^{(1)} = \begin{pmatrix} \mathbf{W}_{11}^{(1)} & \mathbf{W}_{12}^{(1)} & \cdots & \mathbf{W}_{1d}^{(1)} \\ \mathbf{W}_{21}^{(1)} & \mathbf{W}_{22}^{(1)} & \cdots & \mathbf{W}_{2d}^{(1)} \\ \vdots & \vdots & & \vdots \\ \mathbf{W}_{d_h1}^{(1)} & \mathbf{W}_{d_h2}^{(1)} & \cdots & \mathbf{W}_{d_hd}^{(1)} \end{pmatrix}$$

Soit $\mathbf{b}^{(1)}$ le vecteur de biais caractérisant des connexions synaptiques allant de la couche d'entrée à la couche cachée.

$$\mathbf{b}^{(1)} = \begin{pmatrix} \mathbf{b}_1^{(1)} \\ \mathbf{b}_2^{(1)} \\ \vdots \\ \mathbf{b}_{d_h}^{(1)} \end{pmatrix} \in \mathbb{R}^{d_h}$$

On a donc que $\mathbf{b^{(1)}}$ est de dimension $d_h \times 1$.
La formule de calcul du vecteur d'activations des neurones de la couche cachée est

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

On obtient les composantes du vecteur $\mathbf{h}^{\boldsymbol{a}}$:

$$\mathbf{h}_j^a = \sum_{k=1}^{d} \mathbf{W}_{jk}^{(1)} \boldsymbol{x}_k^{(i)} + \boldsymbol{b}_j^{(1)}$$

Le vecteur des sorties des neurones de la couche cachée est

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

Note : On considère ici que la fonction rect est appliquée à chaque composante du vecteur.

2. Soit $\mathbf{W}^{(2)}$ la matrice de poids

$$\mathbf{W}^{(2)} = \begin{pmatrix} \mathbf{W}_{11}^{(2)} & \mathbf{W}_{12}^{(2)} & \cdots & \mathbf{W}_{1d_h}^{(2)} \\ \mathbf{W}_{21}^{(2)} & \mathbf{W}_{22}^{(2)} & \cdots & \mathbf{W}_{2d_h}^{(2)} \\ \vdots & \vdots & & \vdots \\ \mathbf{W}_{m1}^{(2)} & \mathbf{W}_{m2}^{(2)} & \cdots & \mathbf{W}_{md_h}^{(2)} \end{pmatrix}$$

Soit $\mathbf{b}^{(2)}$ le vecteur de biais caractérisant des connexions synaptiques allant de la couche cachée à la couche de sortie.

$$\mathbf{b}^{(2)} = \begin{pmatrix} \mathbf{b}_1^{(2)} \\ \mathbf{b}_2^{(2)} \\ \vdots \\ \mathbf{b}_m^{(2)} \end{pmatrix} \in \mathbb{R}^m$$

On a donc que $\mathbf{W}^{(2)}$ et $\mathbf{b}^{(2)}$ sont de dimensions $m \times d_h$ et $m \times 1$ respectivement.
La formule de calcul du vecteur d'activations des neurones de la couche sortie est

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

On obtient les composantes du vecteur $\mathbf{o}^a$:

$$\mathbf{o}_k^a = \sum_{b=1}^{d_h} \mathbf{W}_{kb}^{(2)}\mathbf{h}_b^s + \mathbf{b}_k^{(2)}$$

3. La sortie des neurones de sorties est donnée par

$$\mathbf{o}^s = \mathrm{softmax}(\mathbf{o}^a)$$

On a que

$$\mathrm{softmax}(\mathbf{o}_1^a, \mathbf{o}_2^a, \ldots, \mathbf{o}_m^a) = \frac{1}{\sum_{k'=1}^m \exp(\mathbf{o}_{k'}^a)}(\exp(\mathbf{o}_1^a), \exp(\mathbf{o}_2^a), \ldots, \exp(\mathbf{o}_m^a))$$

Donc

$$\mathbf{o}_k^s = \frac{\exp(\mathbf{o}_k^a)}{\sum_{k'=1}^m \exp(\mathbf{o}_{k'}^a)}$$

La fonction exponentielle est définie positive. De plus, le quotient de deux nombres positifs est positif. Donc $\mathbf{o}_k^s$ sont positifs.
Par ailleurs,

$$\sum_{k=1}^m \mathbf{o}_k^s = \frac{\exp(\mathbf{o}_1^a)}{\sum_{k'=1}^m \exp(\mathbf{o}_{k'}^a)} + \cdots + \frac{\exp(\mathbf{o}_m^a)}{\sum_{k'=1}^m \exp(\mathbf{o}_{k'}^a)} = \frac{\sum_{k'=1}^m \exp(\mathbf{o}_{k'}^a)}{\sum_{k'=1}^m \exp(\mathbf{o}_{k'}^a)} = 1$$

Ces deux observations sont très importantes puisqu'on considère les sorties comme étant des probabilités de classe.

4. Le calcul pour trouver la fonction de perte en fonction de $\mathbf{o}^a$.

$$\begin{aligned}
L(\boldsymbol{x}, y) &= -\log \mathbf{o}_y^s(\boldsymbol{x}) \\
&= -\log\left(\frac{\exp(\mathbf{o}_y^a)}{\sum_{k'=1}^m \exp(\mathbf{o}_{k'}^a)}\right) \\
&= -\left(\log\exp(\mathbf{o}_y^a) - \log\left(\sum_{k'=1}^m \exp(\mathbf{o}_{k'}^a)\right)\right) \\
&= \log\left(\sum_{k'=1}^m \exp(\mathbf{o}_{k'}^a)\right) - \mathbf{o}_y^a
\end{aligned}$$

5. On cherche $\hat{\mathrm{R}}$.

$$f_\theta(\boldsymbol{x}^{(i)}) = \mathrm{softmax}(\mathbf{o}^a)$$

$$\hat{\mathrm{R}}\left(f_\theta, \mathrm{D}_n\right) = \frac{1}{n}\sum_{i=1}^n L(f_\theta(\boldsymbol{x}^{(i)}), y^{(i)}) = \frac{1}{n}\sum_{i=1}^n \left(\log\left(\sum_{k'=1}^m \exp(\mathbf{o}_{k'}^a)\right) - \mathbf{o}_{y^{(i)}}^a\right)$$

$$\hat{\mathrm{R}}_\lambda\left(f_\theta, \mathrm{D}_n\right) = \hat{\mathrm{R}}\left(f_\theta, \mathrm{D}_n\right) + \lambda\Omega(\theta)$$

L'ensemble des paramètres du réseau est

$$\theta = \{\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(2)}\}$$

Le nombre de paramètres scalaires est

$$n_\theta = d_h d + d_h + m d_h + m$$

Le problème d'optimisation est

$$\theta^* = \underset{\theta}{\mathrm{argmin}}\hat{\mathrm{R}}_\lambda\left(f_\theta, \mathrm{D}_n\right)$$

6. On utilise la technique de descente de gradient.
   Pseudo-code :
   Initialiser les paramètres aléatoirement.
   Mise à jour des paramètres itérativement :

$$\theta \leftarrow \theta - \eta \frac{\partial \hat{\mathrm{R}}_\lambda}{\partial \theta}$$

7. On sait que

$$L(\boldsymbol{x}, y) = \log\left(\sum_{k'=1}^{m} \exp(\mathbf{o}^a_{k'})\right) - \mathbf{o}^a_y$$

   On veut montrer que

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

   Pour $k \neq y$, on a

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{o}^a_k} &= \frac{\frac{\partial}{\partial \mathbf{o}^a_k}\sum_{k'=1}^{m} \exp(\mathbf{o}^a_{k'})}{\sum_{k'=1}^{m} \exp(\mathbf{o}^a_{k'})} \\ &= \frac{\exp(\mathbf{o}^a_k)}{\sum_{k'=1}^{m} \exp(\mathbf{o}^a_{k'})}\end{aligned}$$

   Pour $k = y$, on a

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{o}^a_y} &= \frac{\frac{\partial}{\partial \mathbf{o}^a_y}\sum_{k'=1}^{m} \exp(\mathbf{o}^a_{k'})}{\sum_{k'=1}^{m} \exp(\mathbf{o}^a_{k'})} - 1 \\ &= \frac{\exp(\mathbf{o}^a_y)}{\sum_{k'=1}^{m} \exp(\mathbf{o}^a_{k'})} - 1\end{aligned}$$

   De plus, $\mathrm{onehot}_m(y)$ est un vecteur de dimension $m \times 1$ où toutes les composantes sont égales à 0 sauf celle à la position y qui est égale à 1.
   On a donc que

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

8. L'expression du gradient en numpy est
   `grad_oa = os - y = fprop(train_data[i,:-1]) - onehot(train_data[i,-1],m)`
9. On cherche les gradients par rapport aux paramètres $\mathbf{W}^{(2)}$ et $\mathbf{b}^{(2)}$

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{W}^{(2)}_{kj}} &= \frac{\partial L}{\partial \mathbf{o}^a_k}\frac{\partial \mathbf{o}^a_k}{\partial \mathbf{W}^{(2)}_{kj}} \\ &= \frac{\partial L}{\partial \mathbf{o}^a_k}\mathbf{h}^s_j\end{aligned}$$

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{b}^{(2)}_{k}} &= \frac{\partial L}{\partial \mathbf{o}^a_k}\frac{\partial \mathbf{o}^a_k}{\partial \mathbf{b}^{(2)}_{k}} \\ &= \frac{\partial L}{\partial \mathbf{o}^a_k}\end{aligned}$$

10. La forme matricielle du calcul du gradient par rapport au paramètre $\mathbf{W}^{(2)}$ est

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

où $\frac{\partial L}{\partial \mathbf{o}^a}$ est de dimension $m \times 1$, $(\mathbf{h}^s)^T$ est de dimension $1 \times d_h$ et $\frac{\partial L}{\partial \mathbf{W}^{(2)}}$ est de dimension $m \times d_h$.
La forme matricielle du calcul du gradient par rapport au paramètre $\mathbf{b}^{(2)}$ est

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

où $\frac{\partial L}{\partial \mathbf{b}^{(2)}}$ et $\frac{\partial L}{\partial \mathbf{o}^a}$ sont de dimension $m \times 1$.

Les expressions correspondantes en numpy sont :

```
grad_b2 = grad_oa
grad_W2 = np.dot(grad_oa,hs.T)
```

11. Les dérivées partielles du coût L par rapport aux sorties des neurones de la couche cachée sont

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{h}^s_j} &= \sum_{k=1}^{m} \frac{\partial L}{\partial \mathbf{o}^a_k} \frac{\partial \mathbf{o}^a_k}{\partial \mathbf{h}^s_j} \\ &= \sum_{k=1}^{m} \frac{\partial L}{\partial \mathbf{o}^a_k} \mathbf{W}^{(2)}_{kj}\end{aligned}$$

12. La forme matricielle est

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

où $\frac{\partial L}{\partial \mathbf{h}^s}$ est de dimension $d_h \times 1$, $(\mathbf{W}^{(2)})^T$ est de dimension $d_h \times m$ et $\frac{\partial L}{\partial \mathbf{o}^a}$ est de dimension $m \times 1$.

L'expression correspondante en numpy est :

```
grad_hs = np.dot(W2.T,grad_oa)
```

13. Les dérivées partielles par rapport aux activations des neurones de la couche cachée sont :

$$\frac{\partial L}{\partial \mathbf{h}^a_j} = \frac{\partial L}{\partial \mathbf{h}^s_j} \frac{\partial \mathbf{h}^s_j}{\partial \mathbf{h}^a_j}$$

On sait que

$$\mathbf{h}^s_j = \mathrm{rect}(\mathbf{h}^a_j)$$

et

$$\frac{\partial \mathrm{rect}(z)}{\partial z} = \mathbf{1}_{z>0}$$

Donc

$$\frac{\partial L}{\partial \mathbf{h}^a_j} = \frac{\partial L}{\partial \mathbf{h}^s_j} \mathbf{1}_{\mathbf{h}^a_j>0}$$

14. Posons

$$\boldsymbol{v} = \begin{pmatrix} \mathbf{1}_{\mathbf{h}^a_1>0} \\ \vdots \\ \mathbf{1}_{\mathbf{h}^a_{d_h}>0} \end{pmatrix}$$

La forme matricielle/vectorielle est

$$\frac{\partial L}{\partial \mathbf{h}^a} = \frac{\partial L}{\partial \mathbf{h}^s} \odot \boldsymbol{v}$$

où l'opérateur $\odot$ est le produit d'Hadamard, c'est-à-dire une multiplication composante par composante.
On a que $\frac{\partial L}{\partial \mathbf{h}^a}$, $\frac{\partial L}{\partial \mathbf{h}^s}$ et $\boldsymbol{v}$ sont de dimension $d_h \times 1$.

L'expression correspondante en numpy est :

```
grad_ha = np.multiply(grad_hs,v)
```

15. On cherche les gradients par rapport aux paramètres $\mathbf{W}^{(1)}$ et $\mathbf{b}^{(1)}$

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{W}^{(1)}_{jk}} &= \frac{\partial L}{\partial \mathbf{h}^a_j}\frac{\partial \mathbf{h}^a_j}{\partial \mathbf{W}^{(1)}_{jk}} \\
&= \frac{\partial L}{\partial \mathbf{h}^a_j}\boldsymbol{x}^{(i)}_k
\end{aligned}$$

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{b}^{(1)}_{j}} &= \frac{\partial L}{\partial \mathbf{h}^a_j}\frac{\partial \mathbf{h}^a_j}{\partial \mathbf{b}^{(1)}_{j}} \\
&= \frac{\partial L}{\partial \mathbf{h}^a_j}
\end{aligned}$$

16. La forme matricielle du calcul du gradient par rapport au paramètre $\mathbf{W}^{(1)}$ est

$$\frac{\partial L}{\partial \mathbf{W}^{(1)}} = \frac{\partial L}{\partial \mathbf{h}^a}(\mathbf{x}^{(i)})^T$$

où $\frac{\partial L}{\partial \mathbf{h}^a}$ est de dimension $d_h \times 1$, $(\boldsymbol{x}^{(i)})^T$ est de dimension $1 \times d$ et $\frac{\partial L}{\partial \mathbf{W}^{(1)}}$ est de dimension $d_h \times d$.
La forme matricielle du calcul du gradient par rapport au paramètre $\mathbf{b}^{(1)}$ est

$$\frac{\partial L}{\partial \mathbf{b}^{(1)}} = \frac{\partial L}{\partial \mathbf{h}^a}$$

où $\frac{\partial L}{\partial \mathbf{b}^{(1)}}$ et $\frac{\partial L}{\partial \mathbf{h}^a}$ sont de dimension $d_h \times 1$.

Les expressions correspondantes en numpy sont :

```
grad_b1 = grad_ha
grad_W1 = np.dot(grad_ha,x.T)
```

17. Les dérivées partielles du coût L par rapport au vecteur d'entrée $\mathbf{x}$ sont :

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{x}^{(i)}_k} &= \sum_{j=1}^{d_h} \frac{\partial L}{\partial \mathbf{h}^a_j}\frac{\partial \mathbf{h}^a_j}{\partial \mathbf{x}^{(i)}_k} \\
&= \sum_{j=1}^{d_h} \frac{\partial L}{\partial \mathbf{h}^a_j}\mathbf{W}^{(1)}_{jk}
\end{aligned}$$

La forme matricielle est

$$\frac{\partial L}{\partial \mathbf{x}^{(i)}} = (\mathbf{W}^{(1)})^T\frac{\partial L}{\partial \mathbf{h}^a}$$

où $\frac{\partial L}{\partial \mathbf{x}^{(i)}}$ est de dimension $d \times 1$, $(\mathbf{W}^{(1)})^T$ est de dimension $d \times d_h$ et $\frac{\partial L}{\partial \mathbf{h}^a}$ est de dimension $d_h \times 1$.

18. On a que

$$\frac{\partial \mathcal{L}(\theta)}{\partial \mathbf{W}^{(1)}_{ij}} = \lambda_{11}\mathrm{sign}(\mathbf{W}^{(1)}_{ij}) + 2\lambda_{12}\mathbf{W}^{(1)}_{ij}$$

$$\frac{\partial \mathcal{L}(\theta)}{\partial \mathbf{W}^{(2)}_{ij}} = \lambda_{21}\mathrm{sign}(\mathbf{W}^{(2)}_{ij}) + 2\lambda_{22}\mathbf{W}^{(2)}_{ij}$$

$$\frac{\partial \mathcal{L}(\theta)}{\partial \mathbf{b}^{(1)}_{ij}} = 0$$

$$\frac{\partial \mathcal{L}(\theta)}{\partial \mathbf{b}^{(2)}_{ij}} = 0$$

et donc, sous forme matricielle,

$$\frac{\partial \mathcal{L}(\theta)}{\partial \mathbf{W}^{(1)}} = \lambda_{11}\text{sign}(\mathbf{W}^{(1)}) + 2\lambda_{12}\mathbf{W}^{(1)}$$

$$\frac{\partial \mathcal{L}(\theta)}{\partial \mathbf{W}^{(2)}} = \lambda_{21}\text{sign}(\mathbf{W}^{(2)}) + 2\lambda_{22}\mathbf{W}^{(2)}$$

$$\frac{\partial \mathcal{L}(\theta)}{\partial \mathbf{b}^{(1)}} = \mathbf{0}$$

$$\frac{\partial \mathcal{L}(\theta)}{\partial \mathbf{b}^{(2)}} = \mathbf{0}$$

Puisque $\tilde{R} = \hat{R} + \mathcal{L}(\theta)$, on a que

$$\frac{\partial \tilde{R}}{\partial \theta} = \frac{\partial \hat{R}}{\partial \theta} + \frac{\partial \mathcal{L}(\theta)}{\partial \theta}$$

où

$$\theta = \{\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(2)}\}$$

Donc

$$\frac{\partial \tilde{R}}{\partial \mathbf{W}^{(1)}} = \frac{1}{n}\sum_{i=1}^{n} \frac{\partial L}{\partial \mathbf{h}^{a}}(\mathbf{x}^{(i)})^T + \lambda_{11}\text{sign}(\mathbf{W}^{(1)}) + 2\lambda_{12}\mathbf{W}^{(1)}$$

$$\frac{\partial \tilde{R}}{\partial \mathbf{W}^{(2)}} = \frac{1}{n}\sum_{i=1}^{n} \frac{\partial L}{\partial \mathbf{o}^{a}}(\mathbf{h}^{s})^T + \lambda_{21}\text{sign}(\mathbf{W}^{(2)}) + 2\lambda_{22}\mathbf{W}^{(2)}$$

Note : On considère ici que la fonction sign est appliquée à chaque composante de la matrice.

$$\frac{\partial \tilde{R}}{\partial \mathbf{b}^{(2)}} = \frac{1}{n}\sum_{i=1}^{n} \frac{\partial L}{\partial \mathbf{o}^{a}}$$

$$\frac{\partial \tilde{R}}{\partial \mathbf{b}^{(1)}} = \frac{1}{n}\sum_{i=1}^{n} \frac{\partial L}{\partial \mathbf{h}^{a}}$$

Les gradients dépendent, entre autres, des $\mathbf{x}^{(i)}$ même s'ils n'apparaissent pas dans les formes condensées écrites ci-dessus.