TD - Gradients

1 Gradients

Question 1 (Dérivation des fonctions composées)**.**
On dit qu'une fonction $f : \mathbb{R} \to \mathbb{R}$ est dérivable en t si $\lim_{h\to 0} \frac{1}{h}\big(f(t+h) - f(t)\big)$ existe. Dans ce cas on note

$$f'(t) = \lim_{h\to 0} \frac{1}{h}\big(f(t+h) - f(t)\big) \ .$$

De manière équivalente, on peut écrire : il existe une fonction ϵ_f^t telle que

$$f(t+h) = f(t) + f'(t)h + h\epsilon_f^t(h) \ .$$

et $\lim_{h\to 0} \epsilon_f^t(h) = 0$.
Soit $f : \mathbb{R} \to \mathbb{R}$ et $g : \mathbb{R} \to \mathbb{R}$. Montrer que

$$(f \circ g)'(t) = f'(g(t)) \times g'(t)$$

Réponse On a $f(t+h) = f(t) + f'(t)h + h\epsilon_f^t(h)$ et $g(t+h) = g(t) + g'(t)h + h\epsilon_g^t(h)$. Ainsi,

$$\begin{aligned} f \circ g(t+h) &= f(g(t+h)) = f(g(t) + g'(t)h + h\epsilon_g^t(h)) \\ &= f(g(t)) + f'(g(t)) \times \big(g'(t)h + h\epsilon_g^t(h)\big) + \big(g'(t)h + h\epsilon_g^t(h)\big)\epsilon_f^{g(t)}\big(g'(t)h + h\epsilon_g^t(h)\big) \\ &= f(g(t)) + f'(g(t))g'(t)h + h\epsilon'(h) \end{aligned}$$

où $\epsilon'(h) = f'(g(t))\epsilon_g^t(h) + (g'(t) + \epsilon_g^t(h))\epsilon_f^{g(t)}\big(g'(t)h + h\epsilon_g^t(h)\big)$.
On bien le résultat demandé car $\lim_{h\to 0} \epsilon'(h) = 0$.

Question 2 (Matrice jacobienne)**.**
On dit qu'une fonction $f : \mathbb{R}^n \to \mathbb{R}$ est dérivable en x si il existe un vecteur $\nabla f(x) \in \mathbb{R}^n$ et une fonction ϵ_f^x tels que

$$f(x+h) = f(x) + \nabla f(x)^\top h + \|h\|\epsilon_f^x(h)$$

où $\lim_{h\to 0} \epsilon_f^x(h) = 0$.
On note les coordonnées de $\nabla f(x)$ de plusieurs manières :

$$(\nabla f(x))_i = \nabla_i f(x) = \frac{\partial f}{\partial x_i}(x) \ .$$

Il se trouve que $\nabla_i f(x)$ est égale à la i^{me} dérivée directionnelle :

$$\nabla_i f(x) = \lim_{t\to 0} \frac{f(x+te_i) - f(x)}{t} \ .$$

Soit $F : \mathbb{R}^n \to \mathbb{R}^m$ une fonction à valeurs vectorielles, c'est à dire que $F(x) = \begin{pmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{pmatrix}$.
On dit que F est dérivable en x si pour tout $i \in \{1, \dots, m\}$, f_i est dérivable en x :

$$f_i(x+h) = f_i(x) + \nabla f_i(x)^\top h + \|h\| \epsilon_{f_i}^x(h)$$

où $\lim_{h \to 0} \epsilon_{f_i}^x(h) = 0$.
On appelle matrice jacobienne de F en x la matrice qui concatène tous les gradients des f_i, c'est à dire

$$J_F(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(x) & \dots & \frac{\partial f_1}{\partial x_n}(x) \\ \vdots & & \vdots \\ \frac{\partial f_m}{\partial x_1}(x) & \dots & \frac{\partial f_m}{\partial x_n}(x) \end{bmatrix} .$$

Vérifier qu'avec cette notation, on a

$$F(x+h) = F(x) + J_F(x) h + o(\|h\|) .$$

Réponse Il suffit d'écrire la i^{me} coordonnée de $F(x+h)$.

Question 3 (Calculs de gradients)**.**

— $f_1(x) = \frac{1}{2}\|Ax - b\|_2^2$, A matrice de taille $m \times n$, $b \in \mathbb{R}^m$, $x \in \mathbb{R}^n$. Calculer le gradient de f_1 en x.
— $f_2(x) = Bx + c$, B matrice de taille $p \times n$, $c \in \mathbb{R}^p$, $x \in \mathbb{R}^n$. Calculer la Jacobienne de f_2 en x.
— $f_3(P, Q) = \frac{1}{2}\|M - PQ\|_F^2$, M matrice de taille $m \times n$, P matrice de taille $m \times k$ et Q matrice de taille $k \times n$. Calculer le gradient de f_3 en (P, Q).

Réponse

$$\begin{aligned} f_1(x+h) &= \frac{1}{2}\|A(x+h) - b\|_2^2 = \frac{1}{2}\|Ax - b\|_2^2 + (Ax-b)^\top A h + \frac{1}{2}\|Ah\|_2^2 \\ &= f_1(x) + (A^\top (Ax - b))^\top h + \|h\|_2 \epsilon_1(h) \end{aligned}$$

où $\lim_{h \to 0} \epsilon_1(h) = \lim_{h \to 0} \frac{\frac{1}{2}\|Ah\|_2^2}{\|h\|_2} = 0$. On a donc $\nabla f_1(x) = A^\top (Ax - b)$.
$f_2(x+h) = B(x+h) + c = f_2(x) + Bh$ donc $J_{f_2}(x) = B$.
Pour f_3, nous allons faire la preuve en calculant toutes les dérivées partielles.
On a $f_3(P, Q) = \frac{1}{2}\sum_{i=1}^m \sum_{j=1}^n (M_{i,j} - \sum_{l=1}^k P_{i,l} Q_{l,j})^2$.
En utilisant la notation $\delta_{ij} = 1$ si $i = j$ et $\delta_{ij} = 0$ si $i \neq j$, on trouve :

$$\begin{aligned} \frac{\partial f_3}{\partial P_{u,v}}(P, Q) &= \sum_{i=1}^m \sum_{j=1}^n \Big(M_{i,j} - \sum_{l=1}^k P_{i,l} Q_{l,j} \Big) \times \Big(- \sum_{l'=1}^k Q_{l',j} \delta_{iu} \delta_{l'v} \Big) \\ &= - \sum_{j=1}^n \Big(M_{u,j} - \sum_{l=1}^k P_{u,l} Q_{l,j} \Big) Q_{v,j} \end{aligned}$$

En reconstruisant la matrice, on trouve

$$\frac{\partial f_3}{\partial P}(P,Q) = -(M-PQ)Q^\top .$$

En faisant de même pour $\frac{\partial f_3}{\partial Q_{u,v}}(P,Q)$, on trouve

$$\begin{aligned}\frac{\partial f_3}{\partial Q_{u,v}}(P,Q) &= \sum_{i=1}^m \sum_{j=1}^n \Big(M_{i,j} - \sum_{l=1}^k P_{i,l}Q_{l,j}\Big) \times \Big(-\sum_{l'=1}^k P_{i,l'}\delta_{l'u}\delta_{jv}\Big) \\ &= -\sum_{i=1}^m \Big(M_{i,v} - \sum_{l=1}^k P_{i,l}Q_{l,v}\Big)P_{i,u}\end{aligned}$$

soit

$$\frac{\partial f_3}{\partial Q}(P,Q) = -P^\top(M-PQ) .$$

Question 4. Soient $F : \mathbb{R}^m \to \mathbb{R}^p$ et $G : \mathbb{R}^n \to \mathbb{R}^m$ deux fonctions dérivables. Montrer que pour tout i, j,

$$\frac{\partial (F\circ G)_j}{\partial x_i}(x) = \sum_{l=1}^m \frac{\partial F_j}{\partial y_l}(G(x))\frac{\partial G_l}{\partial x_i}(x) ,$$

et que cette formule est équivalente à

$$J_{F\circ G}(x) = J_F(G(x))J_G(x) .$$

Réponse On a pour tous x, y, h, $F_j(y+h) = F_j(y) + \nabla F_j(y)^\top h + \|h\|\epsilon^y_{F_j}(h)$
et $G(x+h) = G(x) + J_G(x)h + \|h\|\epsilon^x_G(h)$.
Ainsi, pour tout $t \in \mathbb{R}$,

$$\begin{aligned}&F_j(G(x+te_i)) = F_j(G(x) + tJ_G(x)e_i + |t|\epsilon^x_G(te_i) \\ &= F_j(G(x)) + \nabla F_j(G(x))\big(tJ_G(x)e_i + |t|\epsilon^x_G(te_i)\big) + \|tJ_G(x)e_i + |t|\epsilon^x_G(te_i)\|\epsilon^{G(x)}_{F_j}\big(tJ_G(x)e_i + |t|\epsilon^x_G(te_i)\big) \\ &= F_j(G(x)) + t\nabla F_j(G(x))J_G(x)e_i + |t|\epsilon'(t)\end{aligned}$$

où $\epsilon'(t)$ tend vers 0 quand t tend vers 0.
On a donc

$$\frac{\partial (F\circ G)_j}{\partial x_i}(x) = \nabla F_j(G(x))J_G(x)e_i = \sum_{l=1}^m \frac{\partial F_j}{\partial y_l}(G(x))\frac{\partial G_l}{\partial x_i}(x) .$$

En contruisant la matrice jacobienne à partir de ces dérivées partielles, on trouve bien la formule $J_{F\circ G}(x) = J_F(G(x))J_G(x)$.

2 Rétropropagation dans les réseaux de neurones

Considérons le modèle de réseau de neurones à 1 couche suivant :

$$y = f(w,x) = \sigma\Big(\sum_{i=1}^{H} w_i v_i\big(\sum_{j=1}^{N} w_{i,j}x_j\big)\Big)\ . \tag{1}$$

Dans cette formule :

- $x_1, \dots, x_N$ sont les observations.
- y est la sortie du modèle.
- Le nombre entier H est appelé nombre de neurones.
- σ et $v_1, \dots, v_H$ sont des fonctions fixées appelées fonctions d'activation. On supposera que ces fonctions sont dérivables. Un choix classique est $\sigma(z) = v_i(z) = \tanh(z)$.
- $w_1, \dots, w_H, w_{1,1}, \dots, w_{1,N}, w_{2,1}, \dots, w_{H,1}, \dots, w_{H,N}$ sont les paramètres du modèle. Il y en a $N \times H + H$.

Le but de cette partie du TD est de trouver une formule pour calculer le gradient de f par rapport à w, ce qui est la première étape pour implémenter une méthode de gradient. Cette formule est à la base des logiciels d'apprentissage de réseaux de neurones comme Tensorflow ou Keras.

Question 5. Écrire la fonction $f : \mathbb{R}^{NH+H} \times \mathbb{R}^N \to \mathbb{R}$ du modèle de réseau de neurones (1) comme une composition de fonctions plus simples de la forme suivante :

$$f(w,x) = \sigma \circ M\big(w, V \circ L(w,x)\big)\ .$$

Vous expliciterez les fonctions M, V et L en faisant attention à leur nombre de variables et à la dimension des images.

Réponse Let $L : \mathbb{R}^{H\times N} \times \mathbb{R}^N \mapsto \mathbb{R}^H, V : \mathbb{R}^H \mapsto \mathbb{R}^H$ and $M : \mathbb{R}^H \times \mathbb{R}^H \mapsto \mathbb{R}$ given by

$$\begin{aligned} L(w,x) &= [\sum_{j=1}^{N} w_{1,j}x_j, \dots, \sum_{j=1}^{N} w_{H,j}x_j] \\ V(y) &= [v_1(y_1), \dots, v_H(y_H)] \\ M(w,y) &= \sum_{i=1}^{H} w_i y_i. \end{aligned}$$

Thus clearly

$$f(w,x) = \sigma \circ M\big(w, V \circ L(w,x)\big) = \sigma\Big(\sum_{i=1}^{H} w_i v_i\big(\sum_{j=1}^{N} w_{i,j}x_j\big)\Big). \quad \square$$

Question 6. Calculer les jacobiennes de chacune des fonctions en jeu.

Réponse First we need to flatten the matrix of variables w_{ij}. Let $\mathbf{w} = (w_{1:}, w_{2:}, \ldots, w_{H:})$ be the concatenation of the rows of w_{ij}. With this in mind, the Jacobians are given by

$$\begin{aligned}
J_L(\mathbf{w}, x) &= \begin{bmatrix} x & 0 & \cdots & 0 & w_{1:} \\ 0 & x & 0 & \cdots & w_{2:} \\ \vdots & \ddots & & & \vdots \\ 0 & 0 & \cdots & x & w_{N:} \end{bmatrix} \\
J_V(y) &= \begin{bmatrix} v_1'(y_1) & 0 & \cdots & 0 \\ 0 & v_2'(y_2) & 0 & \cdots \\ \vdots & \ddots & & \\ 0 & 0 & \cdots & v_H'(y_H) \end{bmatrix} \\
J_M(w, y) &= \nabla M(w, y) = (y, w)^\top.
\end{aligned}$$

Question 7. Montrer que le gradient de f par rapport à w, que l'on notera $\nabla_w f$ peut s'écrire comme produit matriciel et somme des jacobiennes calculées à question précédente.

Réponse Using again $\mathbf{w} = (w_{1:}, w_{2:}, \ldots, w_{H:})$ we have that $f : (w, \mathbf{w}, x) \in \mathbb{R}^{H \times NH \times N} \to \mathbb{R}$. Consequently $\nabla_{w,\mathbf{w}} f(w, \mathbf{w}, x) \in \mathbb{R}^{NH+H}$. Omitting the arguments for brevity, we have that

$$\nabla_{w,\mathbf{w}} f = \sigma' \nabla M^\top \begin{bmatrix} I_N \\ J_V J_L \begin{bmatrix} I_{N\times H} \\ 0 \end{bmatrix} \end{bmatrix} = \sigma' \begin{bmatrix} y^\top \\ w_1 v_1' x \\ w_2 v_2' x \\ \vdots \\ w_H v_H' x \end{bmatrix}$$

Question 8. Évaluer le nombre d'opérations nécessaires pour calculer $\nabla_w f$ quand on commence par la couche d'entrée du réseau de neurones. On rappelle que pour calculer le produit matriciel $A \times B$ où A est de taille $n \times m$ et B de taille $m \times p$, il faut environ nmp opérations.

Réponse Another way of posing the question is, what is the complexity of calculating

$$\sigma' \nabla M_w^\top \left(J_V \left(J_L \begin{bmatrix} I_{N\times H} \\ 0 \end{bmatrix} \right) \right), \tag{2}$$

and

$$\sigma' \nabla M_y^\top I_N. \tag{3}$$

The cost of computing (3) is negligible in comparison to the cost of computing (2). Thus we will focus only on (2).
First, the cost of computing $J_L I_{N\times M}$ is $O(1)$. Let $A_1 = J_L I_{N\times M} \in \mathbb{R}^{H\times HN}$. If we ignore the structure of the Jacobians, the cost of computing $J_V A_1$ is $O(HN \times H^2) = O(H^3 N)$

Let $A_2 = J_V A_1 \in \mathbb{R}^{\in \mathbb{R}^{H \times HN}}$. The cost of computing $\sigma' \nabla M_w^\top A_2$ is $O(H^2 N)$ Thus the total is

$$O(H^3 N) + O(H^2 N) = O(H^3 N) \tag{4}$$

Question 9. Évaluer le nombre d'opérations nécessaires pour calculer $\nabla_w f$ quand on commence par la couches de sortie du réseau de neurones.

Réponse Computing $\sigma' M_w^\top J_V$ in (2) is $O(H^2 N)$. Let $a_1^\top = \sigma' M_w^\top J_V \in \mathbb{R}^H$. The cost of computing $a_1^\top J_L$ is $O(H^2 N)$. Thus the total cost of computing (2) backwards is

$$O(H^2 N) + O(H^2 N) = O(H^2 N).$$

Which is one order less in powers of H as compared to (4).