Provenance sur les réseaux de neurones

Paul LANDRIER

1 Exemples

1.1 Réseaux de neurones

On s'intéresse d'abord à des réseaux de neurones denses avec une fonction d'activation f.

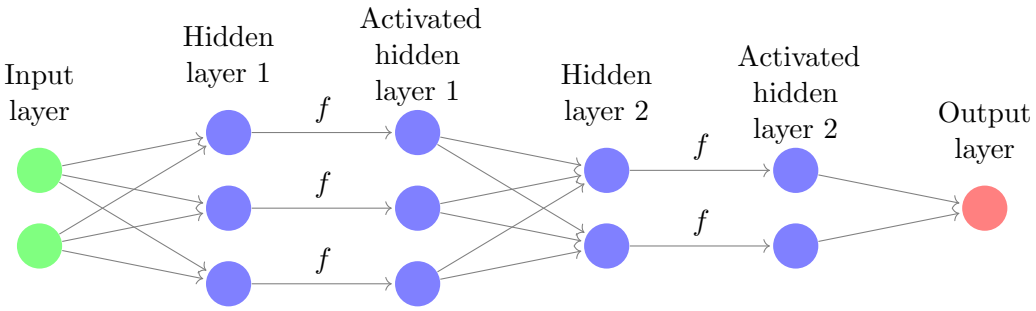

Figure 1: Réseau de neurone N

On appelle respectivement E, H_1, H_1', H_2, H_2' et o les vecteurs représentant les couches successives. On définit (A_1, B_1), (A_2, B_2) et (A_3, B_3) les couples matrices vecteurs tels que $H_1 = A_1E + B_1$, $H_2 = A_2H_1' + B_2$ et $o = A_3H_2' + B_3$.

En écrivant explicitement l'exécution générale, on obtient :

- $E = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$
- $H_1 = A_1E + B_1 = \begin{pmatrix} a^1_{1,1}x_1 + a^1_{1,2}x_2 + b^1_1 \\ a^1_{2,1}x_1 + a^1_{2,2}x_2 + b^1_2 \\ a^1_{2,1}x_1 + a^1_{2,2}x_2 + b^1_2 \end{pmatrix}$
- $H_1' = f(H_1) = \begin{pmatrix} f(a^1_{1,1}x_1 + a^1_{1,2}x_2 + b^1_1) \\ f(a^1_{2,1}x_1 + a^1_{2,2}x_2 + b^1_2) \\ f(a^1_{3,1}x_1 + a^1_{3,2}x_2 + b^1_3) \end{pmatrix}$
- $H_2 = A_2H_1' + B_2 =$
  $\begin{pmatrix} a^2_{1,1}f(a^1_{1,1}x_1 + a^1_{1,2}x_2 + b^1_1) + a^2_{1,2}f(a^1_{2,1}x_1 + a^1_{2,2}x_2 + b^1_2) + a^2_{1,2}f(a^1_{3,1}x_1 + a^1_{3,2}x_2 + b^1_3) + b^2_1 \\ a^2_{2,1}f(a^1_{1,1}x_1 + a^1_{1,2}x_2 + b^1_1) + a^2_{2,2}f(a^1_{2,1}x_1 + a^1_{2,2}x_2 + b^1_2) + a^2_{2,2}f(a^1_{3,1}x_1 + a^1_{3,2}x_2 + b^1_3) + b^2_2 \end{pmatrix}$
- $H_2' = f(H_2) =$
  $\begin{pmatrix} f(a^2_{1,1}f(a^1_{1,1}x_1 + a^1_{1,2}x_2 + b^1_1) + a^2_{1,2}f(a^1_{2,1}x_1 + a^1_{2,2}x_2 + b^1_2) + a^2_{1,2}f(a^1_{3,1}x_1 + a^1_{3,2}x_2 + b^1_3) + b^2_1) \\ f(a^2_{2,1}f(a^1_{1,1}x_1 + a^1_{1,2}x_2 + b^1_1) + a^2_{2,2}f(a^1_{2,1}x_1 + a^1_{2,2}x_2 + b^1_2) + a^2_{2,2}f(a^1_{3,1}x_1 + a^1_{3,2}x_2 + b^1_3) + b^2_2) \end{pmatrix}$
- $o = a^3_{11}f(a^2_{1,1}f(a^1_{1,1}x_1 + a^1_{1,2}x_2 + b^1_1) + a^2_{1,2}f(a^1_{2,1}x_1 + a^1_{2,2}x_2 + b^1_2) + a^2_{1,2}f(a^1_{3,1}x_1 + a^1_{3,2}x_2 + b^1_3) + b^2_1)$
  $+ a^3_{22}f(a^2_{2,1}f(a^1_{1,1}x_1 + a^1_{1,2}x_2 + b^1_1) + a^2_{2,2}f(a^1_{2,1}x_1 + a^1_{2,2}x_2 + b^1_2) + a^2_{2,2}f(a^1_{3,1}x_1 + a^1_{3,2}x_2 + b^1_3) + b^2_2)$

1.2 Graphes de calculs

Les graphes de calculs sont une autre façon de représenter l'exécution d'un réseau de neurones, en mettant en évidence la structure des calculs.

Commençons par un exemple. Nous définissons la fonction $f(x, y, z) = xe^{(x^2-y^2)z}$. Son graphe de calcul est représenté en Figure 1.2.

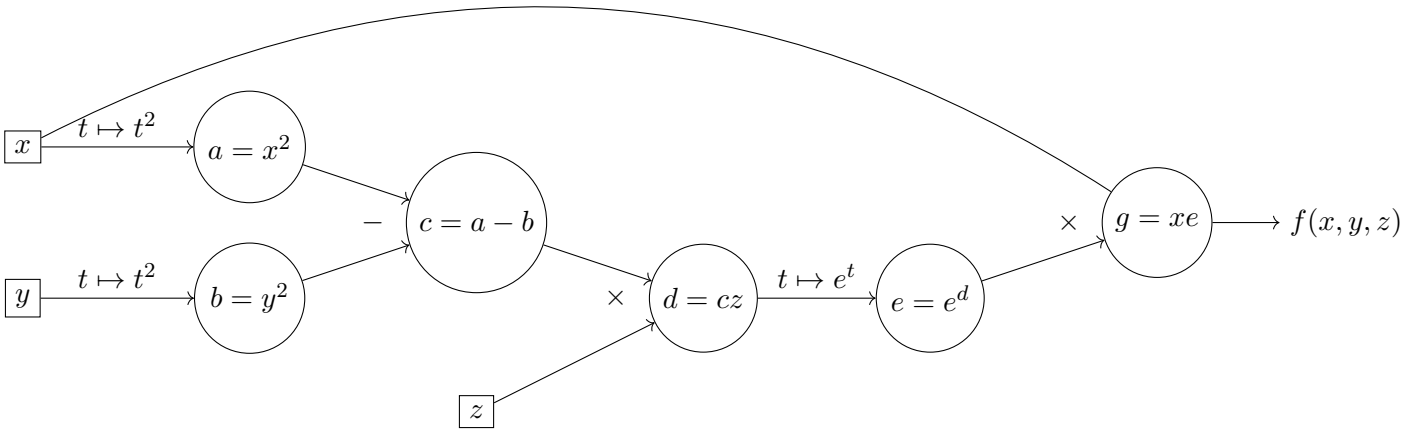

Figure 2: Graphe de calcul de la fonction f

Pour le réseau de neurones N, en considérant les opérations sur les vecteurs, cela donne :

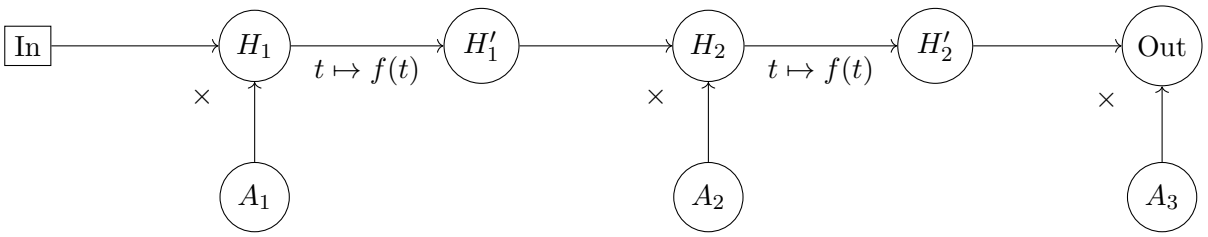

Figure 3: Le graphe de calcul du réseau de neurones N.

1.3 Graphes de calculs - *Backward pass*

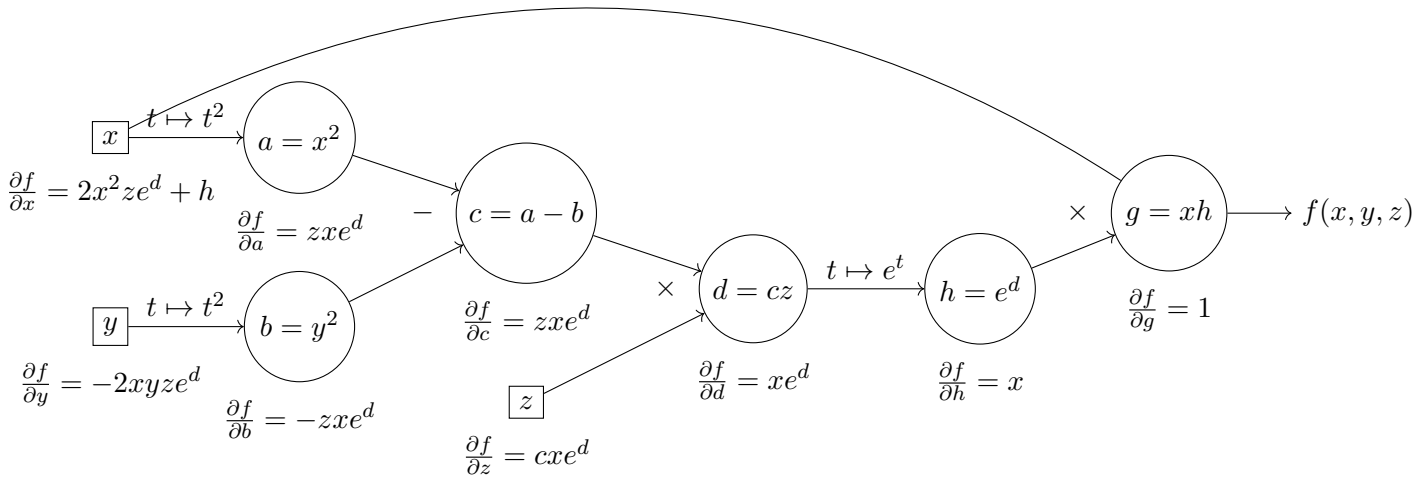

Figure 4: Exemple de *backward pass*.

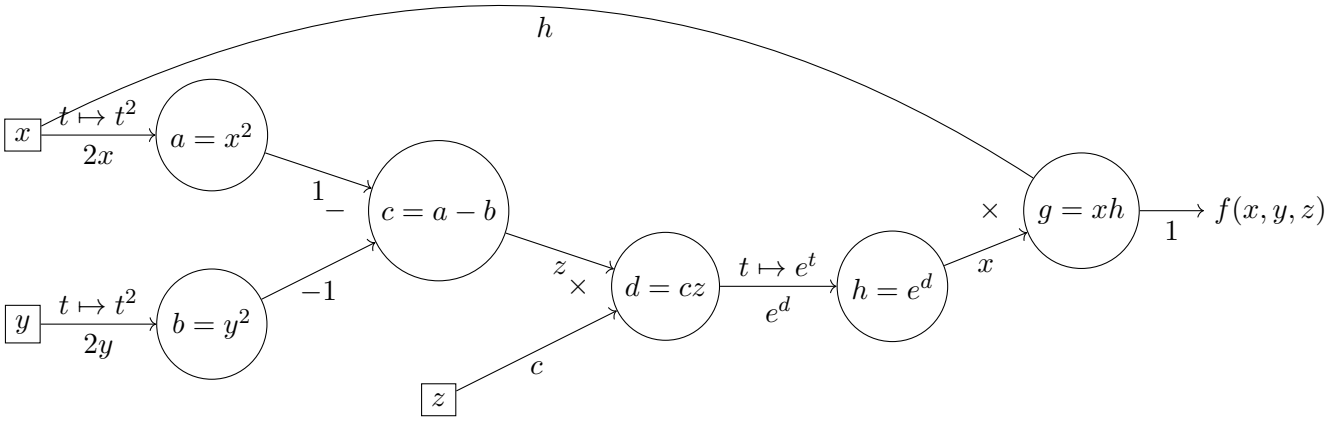

Figure 5: Exemple de *backward pass* avec les annotations sur les arrêtes.

Les graphes de calcul tels que représentés précédemment permettent de simuler une exécution du réseau de neurones ou de la fonction considérée. Cependant, leur intérêt ne se limite pas à cela puisqu'ils permettent aussi de calculer efficacement le gradient de la fonction considérée en remontant le graphe, c'est la *backward pass*.

L'idée repose sur la règle de la chaine :

$$\frac{\partial f}{\partial x} = \sum_{y \text{ child of } x} \frac{\partial y}{\partial x} \frac{\partial f}{\partial y} \tag{1}$$

Formellement, on voit d'abord f comme une fonction de g (la fonction identité) et on a donc $\frac{\partial f(g)}{\partial g} = 1$. Puis on voit g comme une fonction de x et e et on a $\frac{\partial f(g(x,e))}{\partial e} = x$, puis on voit e comme une fonction de d et ainsi de suite jusqu'à obtenir $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial y}$ et $\frac{\partial f}{\partial z}$ qui sont les trois coordonnées du gradient de f.

Le résultat de ces opérations sur le graphe de calcul est montré en figure 1.3. L'équation 1 nous dit que la dérivée partielle de f par rapport à une variable v est la provenance de v à f pour le semi-anneau des réels (ou des fonctions à valeur dans les réels).

2 Provenance

Nous souhaitons abstraire les méthodes précédentes à travers une strucuture algébrique adaptée, de façon analogue à la provenance par semi-anneaux dans les bases de données introduite dans [Green et al.].

2.1 Généralisation directe par les semi-anneaux

L'idée la plus directe consiste à remplacer les calculs sur les réels par des calculs dans un semi-anneau (une multiplication MV entre une matrice M et un vecteur V ne fait intervenir que des additions et multiplications d'éléments). Cependant, il est difficile d'intégrer les non-linéarités dans cette structure. On pourrait le faire pour ReLU en décidant que nous nous intéressons aux fonctions de la forme $x \mapsto \sigma(x)x$ où σ est à valeur dans $0, 1$ (dans le contexte des bases de données, σ est une sélection), mais cette restriction empêche de généraliser le processus à LRP.

2.2 Semi-anneau de fonctions

En regardant la structure du graphe de calcul, chaque nœud représente une fonction des données d'entrée. On peut donc essayer d'abstraire le fonctionnement en faisant appel à de l'algèbre différentielle.

On commence par définir une dérivation sur les semi-anneaux :

Definition 1. *Soit $(S, +, \cdot, 0, 1)$ un semi-anneau. Une dérivation sur S est une application $d : S \to S$ vérifiant, pour tout $x, y \in S$:*

- $d(a + b) = d(a) + d(b)$
- $d(ab) = d(a)b + ad(b)$

Cette définition implique que $d(0) = 0$ mais pas que $d(1) = 0$ (identité dans le semi-anneau des booléens). Est-ce qu'il faut ajouter cette hypothèse ? [littérature absente].

[Est-ce qu'on a envie que le semi-anneau commute ?] On définit donc une structure $(F, +, \cdot, \circ', 0, 1, \{\partial x, \partial y, \partial z\}$ vérifiant :

- $(F, +, \cdot, 0, 1)$ est un semi-anneau
- $\partial x, \partial y, \partial z$ sont des dérivations sur S
- $\partial x, \partial y, \partial z$ commutent : $\partial x \circ \partial y = \partial y \circ \partial x$,...
- $\circ'$ est une opération binaire et associative
- $1 \circ' f = 1$ et $0 \circ' f = 0$.
- Si $d \in \{\partial x, \partial y, \partial z\}$, $d(a \circ' b) = (d(a) \circ' b) \cdot d(b)$

L'idée est de dire que ce qu'il se passe au dessus des arrêtes c'est la composition $\circ'$ et ce qu'il se passe en dessous c'est la propagation avec la provenance et les ∂.

Problèmes : les dérivées partielles ne sont pas uniquement prises relativement à x, y, z, mais l'intérêt de la chose c'est précisément que l'on prend la dérivée d'une fonction dans un nœud par rapport à ses parents ; l'idée généralise assez naturellement les graphes de calculs mais l'application à LRP risque d'être artificielle ; la structure algébrique n'a pas l'air d'avoir été beaucoup utilisée avant.

2.3 Provenance fine

On cherche à raisonner au niveau le plus fin possible, c'est-à-dire sur les équations du paragraphe 1.1. On distingue trois grandes catégries d'objets. La première est constituée des éléments x_1, x_2 de l'input et des biais b, la deuxième est constituée des coefficients des matrices $a_{i,j}$ et la troisième est constituée de fonctions f. Nous appelons E le premier ensemble, A le deuxième et F le troisième.

Afin d'abstraire ce fonctionnement, il convient de trouver des structures adaptées pour ces trois catégories. Pour cela nous commençons par quelques observations :

- Les coefficients $a_{i,j}$ agissent sur les coefficients x_i, par une application $\left\{ \begin{array}{ccc} A \times E & \to & E \\ (a, x) & \mapsto & ax \end{array} \right.$
- L'ensemble E est muni d'une opération $+$.

2.4 Provenance grossière

On se fonde plutôt ici sur la présentation de la figure 1.2, qui correspond à la formule $Out = A_3 f(A_2 f(A_1 E))$ ou encore à l'écriture : $\mathbb{R}^2 \xrightarrow{\times A_1} \mathbb{R}^3 \xrightarrow{f} \mathbb{R}^3 \xrightarrow{\times A_2} \mathbb{R}^2 \xrightarrow{f} \mathbb{R}^2 \xrightarrow{A_3} \mathbb{R}$

Les matrices correspondent donc à des fonctions qui modifient la structure dans laquelle se trouvent les vecteurs et les fonctions modifient la valeur des vecteurs sans modifier la structure. De façon abstraite, on a donc $S_1 \xrightarrow{\varphi_1} S_2 \xrightarrow{\overline{f}} S_2 \xrightarrow{\varphi_2} S_3 \xrightarrow{\overline{f}} S_3 \xrightarrow{\varphi_3} S_4$

Problèmes :

- La modification de la structure complexifie significativement le processus

- Les vecteurs n'ont pas de structure algébrique évidente : sommer des vecteurs n'a pas nécessairement un sens dans notre contexte (la somme d'une image de 1 et d'une image de 2 ne veut plus rien dire), la dilatation par un scalaire n'est pas très éloquente non plus (mais peut-être moins absurde).
- Les matrices n'ont pas de structure algébrique évidente non plus (peut-être que les sommes/différences ont un sens lors des modifications manuelles des modèles.).
- Le problème de l'algébrisation des fonctions d'activation reste entier.