



Descriptif:

Backpropagation (Rétropropagation)

Encadrant: **Hugo Bolloré**

Par: **Aicha Maaoui**

Date: 03/01/2022

Problématique

Le but de cette partie est la description des étapes de rétropropagation (Backpropagation) utilisée dans les classes *Convolution_layer*, *Pooling_layer* et *Softmax_layer* du project Réseau de Neurones Convolutifs.

BackpropagationGeneralités et définitions:**1. Définition de la Rétropropagaion (Backpropagation):**

la rétropropagation du gradient est utilisée pour entraîner un réseau de neurones. Elle met à jour les poids de chaque neurone, en allant de la dernière couche vers la première [1], comme illustré dans la figure (1).

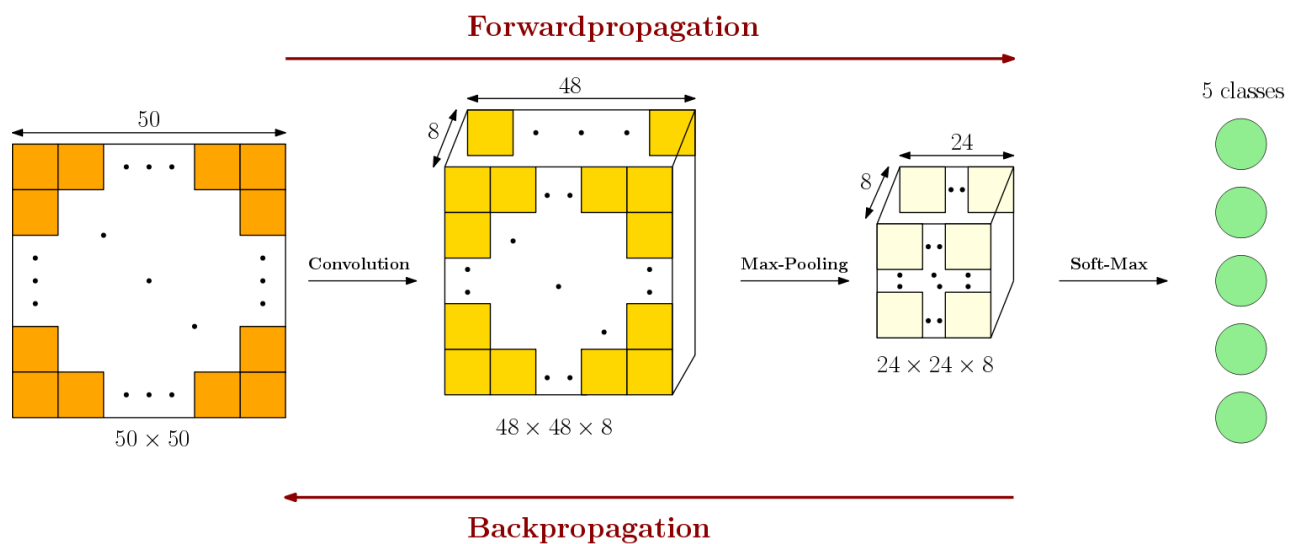


Figure 1: Forward-/ Back-propagation du réseau de neurones utilisé.

Appendix:

On considère un noeud z d'un réseau de neurones [2].

Règle de la chaîne (Chain Rule):

Il s'agit d'une règle de dérivation, qui permet de calculer les dérivées des fonctions composées.

- On pose dans le cas où z ne dépend que de a :

$$\begin{cases} a(t) = \text{fonction}(t) & \text{différentiable en } t \\ z(a(t)) = \text{fonction}(a(t)) & \text{différentiable en } a \end{cases} \quad (1)$$

Alors la dérivée de z par rapport à t dans le cas d'une dépendance unique de Z en t est:

$$\frac{\partial z}{\partial t} = \frac{\partial z}{\partial a} \times \frac{\partial a}{\partial t} \quad (2)$$

Cette procédure est illustrée dans la figure (1).

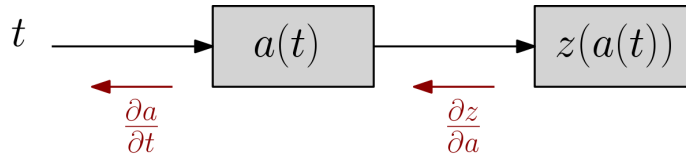


Figure 2: Règle de la chaîne appliquée à une fonction composée de dépendance simple.

- On pose dans le cas où z dépend que de a_1 et a_2 :

$$\begin{cases} a_1(t), a_2(t) = \text{fonction}(t) & \text{différentiable en } t \\ z(a_1(t), a_2(t)) = \text{fonction}(a_1(t), a_2(t)) & \text{différentiable en } a \end{cases} \quad (3)$$

Alors la dérivée de z par rapport à t dans le cas d'une dépendance unique de Z en t est:

$$\frac{\partial z}{\partial t} = \frac{\partial z}{\partial a_1} \times \frac{\partial a_1}{\partial t} + \frac{\partial z}{\partial a_2} \times \frac{\partial a_2}{\partial t} \quad (4)$$

Cette procédure est illustrée dans la figure (2).

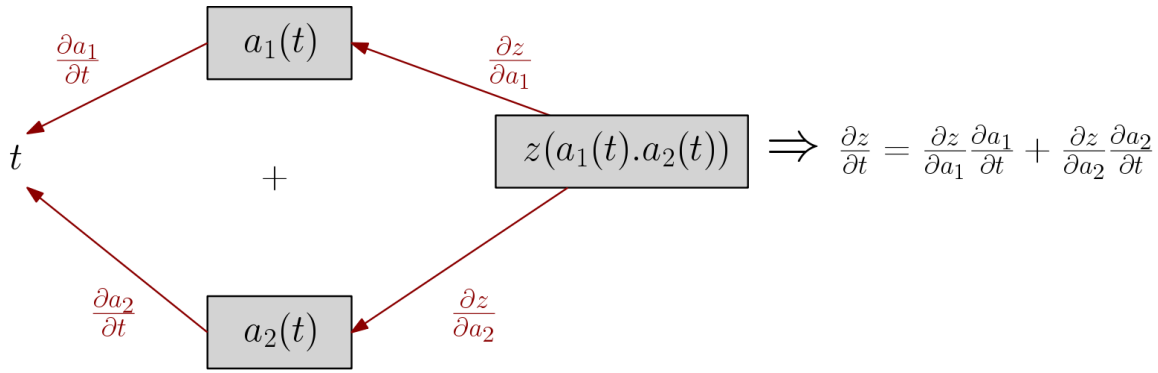


Figure 3: Règle de la chaîne appliquée à une fonction composée de dépendance multiple.

Gradient:

D'après [2], on définit:

- **Gradient Amont (Upstream gradient):** C'est le gradient que le noeud z reçoit lors de la rétropropagation (Backpropagation),
- **Gradients locaux (Local gradients):** C'est les gradients calculés par rapport aux entrées de z ,
- **Gradients Aval (Downstream gradients):** C'est le produit entre le Gradient Amont et Gradients Locaux .

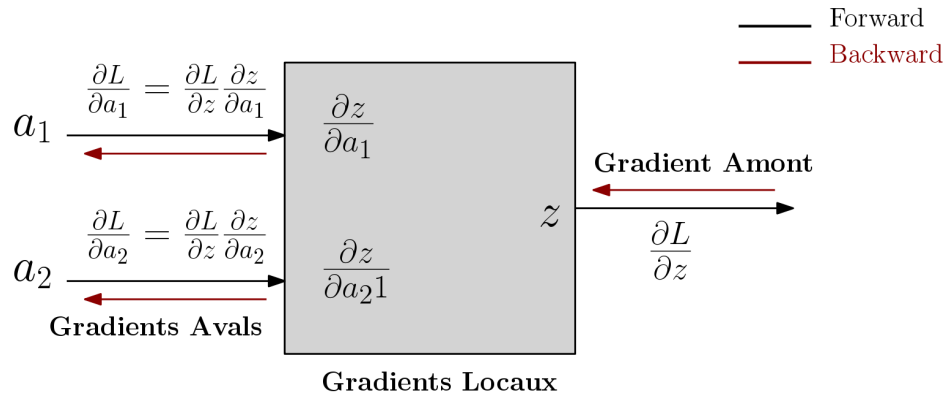


Figure 4: Formes de Gradients.

Règle de Quotient (Quotient Rule):

Soit deux fonctions f et g dépendantes de t . Alors, on a la dérivée du quotient suivant:

$$\frac{\partial}{\partial t} \left(\frac{f(t)}{g(t)} \right) = \frac{f'(t) \times g(t) - f(t) \times g'(t)}{g(t)^2} \quad (5)$$

References:

- [1] "Rétropropagation du gradient", Wikipedia, 2021, "https://fr.wikipedia.org/wiki/R%C3%A9tropropagation_du_gradient".
- [1] "Multi-Layer Neural Networks with Sigmoid Function", Nahua Kang, Towards Data Science, 2017, "<https://towardsdatascience.com/multi-layer-neural-networks-with-sigmoid-function-deep-learning-for-rookies-2-bf464f09eb7f>".
- [2] "Backpropagation in Fully Convolutional Networks", Giuseppe Pio Cannata, Towards Data Science, 2021, "<https://towardsdatascience.com/backpropagation-in-fully-convolutional-networks-fcns-1a13b75fb56a>".