

□ **Rétropropagation du gradient** – La rétropropagation du gradient (en anglais *backpropagation*) est une méthode destinée à mettre à jour les coefficients d'un réseau de neurones en comparant la sortie obtenue et la sortie désirée. La dérivée par rapport au coefficient  $w$  est calculée à l'aide du théorème de dérivation des fonctions composées, et s'écrit de la manière suivante :

$$\frac{\partial L(z,y)}{\partial w} = \frac{\partial L(z,y)}{\partial a} \times \frac{\partial a}{\partial z} \times \frac{\partial z}{\partial w}$$

Ainsi, le coefficient est actualisé de la manière suivante :

$$w \leftarrow w - \eta \frac{\partial L(z,y)}{\partial w}$$

□ **Actualiser les coefficients** – Dans un réseau de neurones, les coefficients sont actualisés comme suit :

- Étape 1 : Prendre un groupe d'observations appartenant aux données du training set.
- Étape 2 : Réaliser la propagation avant pour obtenir le loss correspondant.
- Étape 3 : Effectuer une rétropropagation du loss pour obtenir les gradients.
- Étape 4 : Utiliser les gradients pour actualiser les coefficients du réseau.

□ **Dropout** – Le dropout est une technique qui est destinée à empêcher le sur-ajustement sur les données de training en abandonnant des unités dans un réseau de neurones. En pratique, les neurones sont soit abandonnés avec une probabilité  $p$  ou gardés avec une probabilité  $1 - p$ .

## 3.2 Réseaux de neurones convolutionnels

□ **Pré-requis de la couche convolutionnelle** – Si l'on note  $W$  la taille du volume d'entrée,  $F$  la taille de la couche de neurones convolutionnelle,  $P$  la quantité de zero padding, alors le nombre de neurones  $N$  qui tient dans un volume donné est tel que :

$$N = \frac{W - F + 2P}{S} + 1$$

□ **Normalisation de batch** – C'est une étape possédant les paramètres  $\gamma, \beta$  qui normalise le batch  $\{x_i\}$ . En notant  $\mu_B, \sigma_B^2$  la moyenne et la variance de ce que l'on veut corriger au batch, ceci est fait de la manière suivante :

$$x_i \leftarrow \gamma \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta$$

Cela est normalement effectué après une couche fully-connected/couche convolutionnelle et avant une couche de non-linéarité et a pour but de permettre un taux d'apprentissage plus grand et de réduire une dépendance trop forte à l'initialisation.

## 3.3 Réseaux de neurones récurrents

□ **Types de porte** – Voici les différents types de porte que l'on rencontre dans un réseau de neurones récurrent typique :