

Convolutional neural network

Benjamin GUIGON

January 2021

1 Lien

Article écrit par Hakim Daif

<https://github.com/hakim-daif/PSBX>

2 Produit de convolution

Soient f et g deux suites de nombres réels, on définit leur produit de convolution par :

$$f \otimes g = \sum_{k=-\infty}^{+\infty} f(n-k) \cdot g(k)$$

Cette formule est assez simple car elle revient à faire la somme des produits de 2 suites dans le sens inverse l'une de l'autre.

Produit de convolution pour deux matrices :

Soient A et M deux matrices de dimensions 3×3 ,

$$M = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix}$$

$$M_{\text{retourné}} = \begin{pmatrix} m_{33} & m_{32} & m_{31} \\ m_{23} & m_{22} & m_{21} \\ m_{13} & m_{12} & m_{11} \end{pmatrix}$$

Le produit de convolution $A \otimes M$ s'obtient en sommant la multiplication terme par terme des deux matrices.

C'est tout simplement la multiplication de la matrice A inchangée et la matrice M qui à garder ses valeurs mais dans un ordre différent.

Si les matrices ne sont pas de taille égale, il faut centrer la 2ème matrice et transformer les coefficients latéraux par des 0 quand ils sont à l'extérieurs de la 1ère matrice.

3 Le pooling

Definition : Le pooling (regroupement de termes) consiste transformer une matrice en une matrice plus petite tout en essayant d'en garder les caractéristiques principales. Un pooling de taille k transforme une matrice de taille $n \times p$ en une matrice de taille k fois plus petite, c'est-à-dire de taille $n//k \times p//k$. Une sous-matrice de taille $k \times k$ de la matrice de départ produit un seul coefficient de la matrice d'arrivée.

Les pooling les plus récurrents sont souvent le *max* et *avg*. C'est à dire que l'on prend la valeur maximum ou la valeur moyennée de chaque sous matrice.

Le pooling permet d'éviter le sur-apprentissage en rentrant des matrices trop volumineuses dans le CNN.

4 Fonction d'activation

A chaque couche est associée une fonction d'activation. Cette dernière va servir à mimer le fonctionnement des neurones dans le cerveau humain.

Il existe quelques fonctions d'activation récurrentes :

- ReLU
- tanh
- Sigmoid
- etc...

Elles permettent toutes des effets différents, ReLU sert à converger plus rapidement, la sigmoïde permet d'être moins catégorique que tanh.

5 Architecture d'un CNN

5.1 Feature Learning

Cette partie permet de réduire le nombre de données. Elle est composée de 3 couches :

- Convolutional layer : application du filtre convolutif
- Activation layer : application de la fonction d'activation
- MaxPooling layer : application de la fonction de Pooling

5.2 Classification

C'est untraditionnel " Multi Layer Perceptron w , utilisant la fonction softmax en sortie.

La sortie se calcule :

$$Z(X_1, X_2, X_3 \dots) = X_1 * \omega_1 + X_2 * \omega_2 + \dots + X_n * \omega_n$$

$$Y = ReLU(z) = \max(0, z)$$

Ou max est en fait la fonction *softmax*.

6 Avis

C'est un document très interessant pour commencer à apprendre les réseaux de neurones. Les Réseaux de neurones sont très utiles donc il faut absolument les connaitre et comprendre leur fonctionnement pour tous Data scientist.

En plus de ca, il est très didactique et donne tous les éléments pour comprendre. La dernière étape est maintenant de se familiariser avec le CNN sur mnist pour bien comprendre toutes les subtilités.