

Une introduction à l'apprentissage profond

Houcine Senoussi

March 27, 2018

- 1 Introduction
- 2 Réseau convolutif (CNN)
 - Couche de convolution
 - Pooling
 - Couches fully connected
 - Récapitulatif
- 3 Conclusion
- 4 References

De quoi s'agit-il ?

- Profondeur d'un réseau neurone : nombre de niveaux.
- Une première définition : un réseau de neurones profond (**deep**) est un réseau ayant plus de niveaux qu'un réseau de neurones ordinaire (**shallow**).
- Question : à quoi servent ces niveaux supplémentaires ?
 - À représenter les données selon différents niveaux d'**abstraction**.
- Exemple de niveaux d'abstraction : une image peut être décrite par l'ensemble de ses pixels (niveau le plus bas), les groupes de pixels ayant une forme particulière, les arêtes et les objets (niveau(x) le(s) plus haut(s)).

De quoi s'agit-il ?-2

- Exemple d'application : Une image représentant un visage humain :
 - Le visage se compose des yeux, du nez, etc ...
 - Chaque partie 'simple' du visage se compose de groupes de pixels ayant une forme particulière.
 - Chaque forme se compose de pixels.
- D'où le complément de la définition : Dans une architecture profonde, chaque niveau permet un traitement non-linéaire de l'information issue des niveaux plus bas (en termes d'abstraction), c'est-à-dire plus proches de la couche d'entrée. Ce traitement consiste à créer de nouveaux objets correspondant à un niveau d'abstraction plus élevé.

De quoi s'agit-il ?-3

- Motivation/Intérêt :
 - Un nombre de niveaux réduit peut conduire à une explosion du nombre de noeuds par niveau.
 - Les humains apprennent d'abord des concepts simples, et les combinent ensuite pour représenter des concepts plus abstraits.
 - Cela est notamment dû à l'architecture 'profonde' du cerveau. Ceci est particulièrement vrai pour le cortex visuel qui contient une série d'aires, chacune contenant une représentation différente du champ visuel.

Types de réseaux profonds

- Plusieurs types de réseaux de neurones profonds existent. Parmi lesquels il y a :
 - les réseaux **récurrents** : ce sont des réseaux dans lesquels la sortie d'un neurone peut être connectée à l'entrée d'un neurone appartenant à un niveau inférieur. Autrement dit, ce sont des réseaux contenant des **cycles**.
 - Remarque : on oppose réseaux récurrents aux réseaux 'acycliques' ou 'feed-forward' :
 - les réseaux **convolutifs** (CNN, pour Convolutional Neural Network) : ainsi appelés car l'une des opérations principales pour passer d'un niveau à l'autre est la **convolution** (en l'occurrence entre un tableau à deux dimensions représentant un niveau du réseau et un tableau plus petit appelé filtre (filter) ou noyau (kernel)).

Principe

- L'idée de départ des CNN est de prendre en compte la **structure spatiale** des données.
 - Le principal domaine d'application de ce type de réseau étant le traitement d'images, les neurones du niveau d'entrée sont disposés sous la forme d'un **tableau à deux dimensions**.
- L'opération principale qui relie deux niveaux successifs est la **convolution**.
 - Cela donne lieu aux **couches de convolution** dont chaque neurone est connecté aux neurones d'un carré $t \times t$ de la couche précédente.
- Une autre opération importante est le **pooling**.
 - Une couche de pooling peut être considérée comme une copie sous-échantillonnée de la couche qui la précède.

Principe-2

- Le réseau se termine par une ou plusieurs couches entièrement connectées, telles qu'on en trouve dans les réseaux de neurones ordinaires.
- D'autres types de couches peuvent exister dans le réseau (voir plus bas).

Couche de convolution

- Étant donné un tableau T de taille $n \times n$ et un tableau W de taille $m \times m$, avec $m \leq n$, la convolution de T par W est le tableau $Conv$ défini comme suit :
 - $Conv$ est de taille $(n - m + 1) \times (n - m + 1)$.
 - $Conv(i, j) = \sum_{k=0}^{m-1} \sum_{l=0}^{m-1} W(k, l) * T(i + k, j + l)$.
- De manière plus visuelle, pour calculer $Conv(i, j)$:
 - Poser W au dessus de T de manière à faire coïncider $W(0, 0)$ avec $T(i, j)$.
 - Multiplier chaque élément de W par celui de T avec lequel il coïncide.
 - Additionner tous les produits.
- Le tableau W est appelé **filtre** ou **noyau** ou encore template.
- L'objectif de cette opération est en général est détecter des parties de l'image T semblables au template W .

Couche de convolution

- Une couche de convolution est définie à l'aide de la convolution de la couche qui la précède par un filtre contenant des poids $w_{k,l}$.
- Plus exactement, en notant y la sortie des neurones d'une couche de convolution et x celle de la couche précédente, nous avons:

$$y(i,j) = \sigma(b + \sum_{k=0}^{t-1} \sum_{l=0}^{t-1} w_{k,l} * x_{i+k,j+l})$$

- avec :
 - b est le biais,
 - $w_{k,l}$ est le poids associé au neurone $(i+k, j+l)$
 - σ est une fonction d'activation (par exemple : la fonction sigmoïde).

Couche de convolution

- Il est important de noter que les coefficients $w_{k,l}$ et b sont communs à tous les neurones du tableau.
 - L'idée est donc de détecter le même motif à tous les emplacements du tableau contenant les entrées. (Receptive fields)
- En revanche, un niveau de convolution ne contient pas un seul tableau de neurones mais plusieurs. Nous appellerons ces tableaux des **feature maps**. Chacun de ces tableaux est défini à l'aide d'un filtre différent. Cela permet de détecter des motifs différents dans un même tableau d'entrée.
- Les couches de convolution ont donc besoin de définir les hyperparamètres suivants :
 - le nombre de filtres ainsi que la forme et la taille de chacun d'entre eux.

Pooling

- Les niveaux 'Pooling' arrivent en général à la suite des niveaux convolutionnels.
- Leur rôle est de '**simplifier**'/'**résumer**' l'information contenue dans les niveaux convolutionnels.

Pooling-2

- Plus précisément, le pooling consiste à :
 - regrouper les neurones d'un niveau convolutionnel par carré $p \times p$. Notons $Carr(i, j)$ le carré dont les neurones occupent les positions $\{(p * i, p * j), \dots, (p * (i + 1) - 1, p * (j + 1) - 1)\}$
 - Le niveau Pooling contient un tableau dont chaque neurone est connecté à un carré du niveau précédent et dont la valeur est définie par :
 - $P(i, j) = f(Carr(i, j))$.
 - f est la fonction qui sert à 'résumer' les neurones du carré $Carr(i, j)$. On prend souvent $f = \max$. Dans ce cas nous parlerons de Max-Pooling. Les autres fonctions utilisées sont : la moyenne, la racine carrée de la somme des carrés (L2-Pooling).
- Les couches Pooling ont donc besoin de définir les hyperparamètres suivants :
 - taille p de Pooling (en général prise égale à 2) et fonction de pooling.

Pooling-3

- 1 La figure suivante montre un exemple de Max-pooling

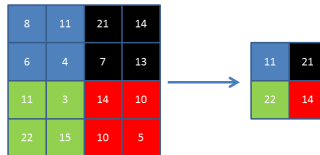


Figure: Pooling

Pooling-3

- 1 La figure suivante résume la succession entrée-couche convolutionnelle-couche pooling

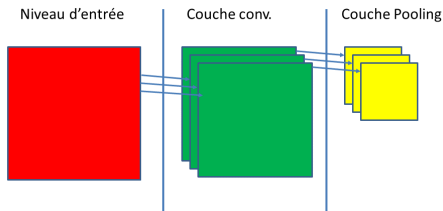


Figure: Succession des couches

Couches fully connected

- La sortie d'un réseau est assurée par une couche entièrement connectée (FC, fully connected) telle qu'on en trouve dans les réseaux de neurones ordinaires.
 - Un neurone d'une telle couche est connectée à tous les neurones de la couche précédente (en général une couche Pooling).
- Comme nouveaux paramètres, nous avons besoin de définir les poids w et les biais b associés à chaque neurone de sortie.

Architecture globale

- Un réseau CNN sera composé d'une couche d'entrée, suivie d'une couche de convolution suivie d'une couche Pooling elle même suivie d'une couche convolution suivie d'une couche pooling ... et se termine par une couche de sortie FC.
- Une couche de convolution et une couche Pooling successives contiennent le même nombre de tableau.
- Nous pouvons donc noter un réseau CNN de la manière suivante :

$$\bullet \text{ Input} \rightarrow (\text{Cnv} \rightarrow \text{Pool}) \times p1 \dots \rightarrow (\text{Cnv} \rightarrow \text{Pool}) \times pn \rightarrow \text{FC}$$

Avantages

- Prise en compte de la structure spatiale des données
- Grâce au fait qu'un neurone est en général connecté à un petit nombre de neurones du niveau précédent et que les poids et les biais sont les mêmes dans chaque tableau, les CNN réduisent le nombre de paramètres à calculer (par rapport à celui d'un réseau ordinaire fully connected).
- **Exercice** : Comparer le nombre de poids et biais à calculer dans un réseau ordinaire et dans un CNN.

Remarque

- Les CNN peuvent aussi comporter deux autres types de couches dont le but est d'améliorer les performances :
 - Exemple : les couches **ReLU** (pour Rectified Linear Units) qui sont insérées entre les couches convolutionnelles et les couches Pooling. L'objectif étant de 'corriger' les sorties des neurones par exemple en prenant le max entre 0 et la valeur de la sortie.

choix des paramètres et apprentissage

- Pour définir complètement un réseau CNN nous devons :
 - choisir le nombre de filtres et leurs formes pour chaque couche de convolution. Ces choix dépendent du type d'images à traiter.
 - Exemple : pour la reconnaissance des chiffres manuscrits, on a l'habitude de prendre des filtres de tailles 5×5
 - Choisir la taille des carrés ainsi que la fonction à utiliser dans chaque couche Pooling.
 - Appliquer la descente du gradient pour l'apprentissage des différents coefficients et biais, comme pour les réseaux de neurones ordinaires.

Conclusion

- Nous avons introduit l'apprentissage profond à travers l'un des réseaux les plus utilisés : Le réseau de neurones convolutionnel.

Bibliographie

- M.A. Nielsen, "Neural networks and deep learning".
Determination Press, 2015.
- Cours du CNAM (RCP209)
- Wikipedia.
- Livre : Deep learning methods and applications , Li Dend and
Don Yu.