

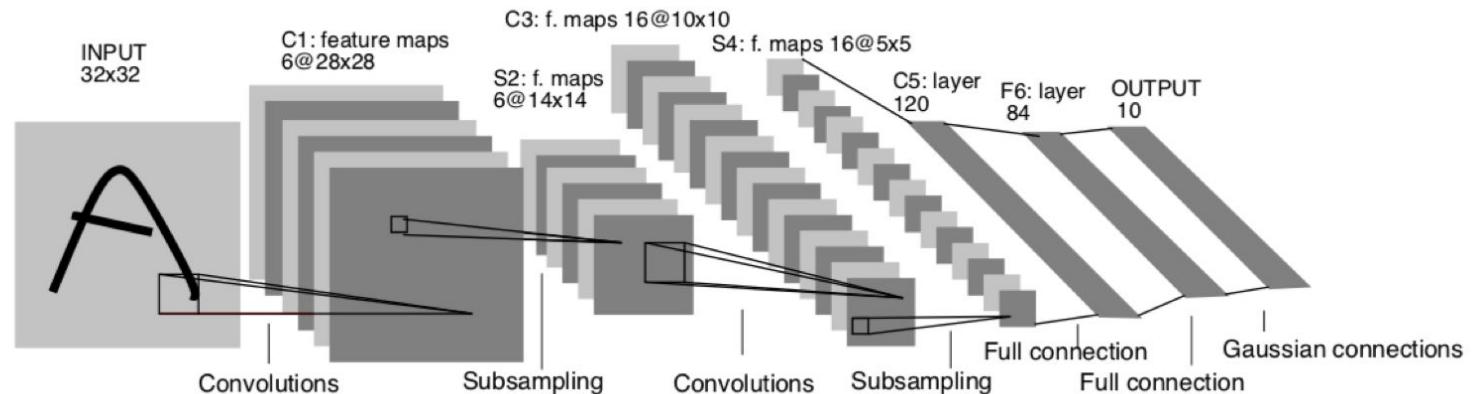
Réseaux de neurones profonds

Amira Barhoumi

amira.barhoumi@univ-grenoble-alpes.fr

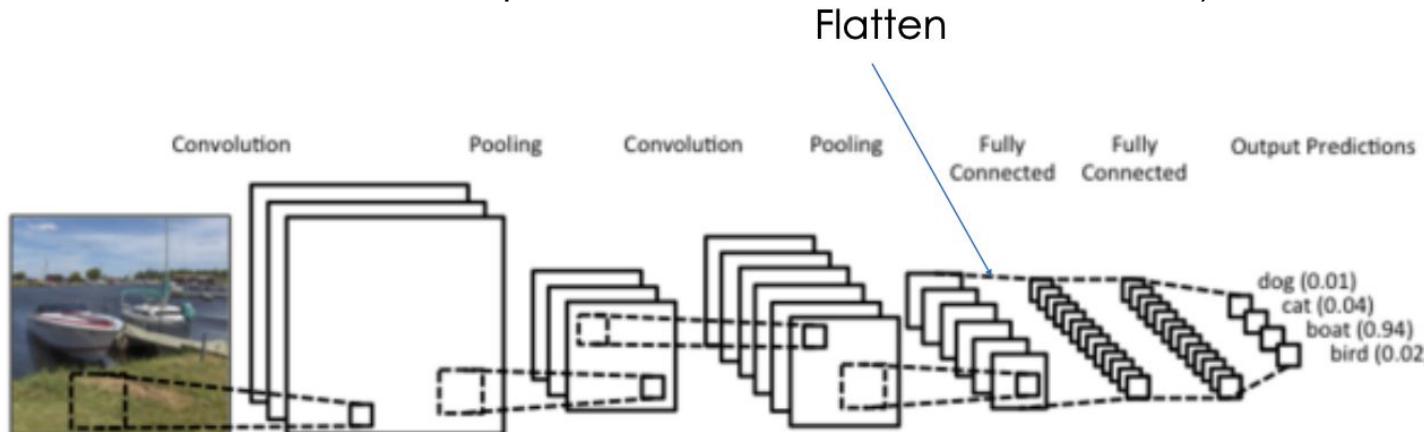
Introduction

- Réseaux de neurones convolutifs (CNN ou ConvNet)
- Succès dans un grand nombre d'applications: reconnaissance de l'écriture manuscrite, détection des objets, reconnaissance faciale, analyse de texte, etc.
- Premier réseau convolutif [LeNet](#) en 1990 dédié à la classification d'images de chiffres manuscrits



Introduction

- Motivations du CNN sur les images : corrélation locale des données, reconnaissance des formes, invariance par translation (les formes peuvent se trouver à différents endroits de l'image)
- CNN sont conçus pour traiter des données qui se présentent sous la forme de tableaux de valeurs en N dimensions, avec N strictement positif. Par exemple, une image couleur se compose de 3 tableaux 2D contenant les intensités de pixels dans les 3 canaux de couleurs RVB)



<https://towardsdatascience.com/build-your-own-convolution-neural-network-in-5-mins-4217c2cf964f>

Introduction

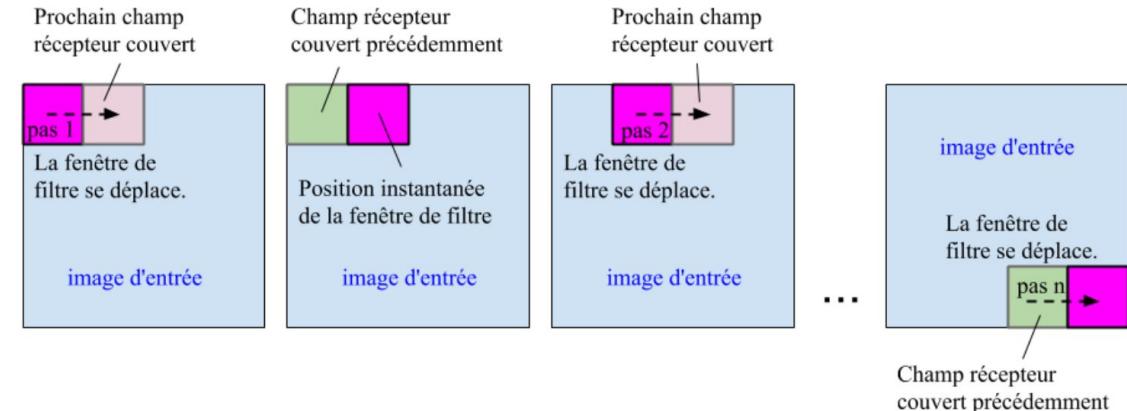
- Les approches d'apprentissage automatique standard nécessitent une définition au préalable d'un ensemble de descripteurs adéquats à la tâche de classification.
- Les performances de ces approches sont fortement conditionnées par la qualité de descripteurs pré-définis utilisés et leurs pertinences à la tâche.
- Contrairement aux approches d'apprentissage automatique standard basées sur des descripteurs pré-définis (phase de *features engineering*), un CNN est capable de définir et extraire, au moment de l'apprentissage, des descripteurs spécifiques adaptés à la tâche en question.

Introduction

- Types de données se présentent sous la forme de tableaux à multiples dimensions :
 - 1D pour les signaux et les séquences, y compris la langue
 - 2D pour images ou spectrogrammes audios
 - 3D pour les images vidéo ou volumétriques

Structure

- L'architecture d'un CNN typique est structurée en une série d'étapes. Les premières étapes sont composées de deux types de couches : couches convolutives et les couches de regroupement (*pooling*).
- La couche de convolution est l'élément central des réseaux neuronaux convolutifs. Son objectif est de détecter la présence de caractéristiques (*features*) dans les images d'entrée. Cela est réalisé grâce à un filtrage par convolution qui consiste à:
 - faire glisser une fenêtre représentative de la caractéristique sur l'image d'entrée
 - calculer le produit de convolution entre la caractéristique et chaque portion de l'image balayée.



CNN pour le texte

- Il existe une différence d'implémentation de CNN en fonction du domaine d'application. En effet, la différence de développement de réseaux de neurones convolutifs pour le traitement d'images et le traitement de langues réside principalement au moment de la convolution. Cette dernière permet d'extraire des descripteurs via des filtres.
 - En traitement d'images, l'extraction de descripteurs s'effectue sur des petites zones de l'image (entrée du réseau), où chaque filtre se déplace dans deux sens (horizontal et vertical) sur l'image.
 - En traitement de texte, le fonctionnement est différent. En effet, le filtre couvre une séquence de mots et ne se déplace que dans un seul sens.

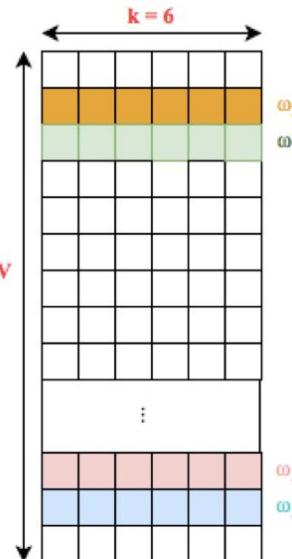
CNN pour le texte

- Entrée du CNN

- En traitement de texte, l'entrée du système représente une séquence de mots sous forme d'une structure unidirectionnelle.
- Exploiter les représentations vectorielles (*embeddings*) de mots pour transformer la séquence de mots en une grille de dimension $n \times k$, où n est le nombre de mots et k la dimension des représentations vectorielles de mots \Rightarrow obtenir une entrée *Embed* sous forme de grille bi-dimensionnelle (*lookup table*).

Ces embeddings constituent les paramètres de *Embed*. Ils sont mis à jour de deux manières différentes au moment de l'apprentissage du CNN:

- ne pas modifier les embeddings. \Rightarrow CNN *statique*
- les embeddings sont mis à jour à l'aide de l'algorithme de rétro-propagation \Rightarrow CNN *non statique*



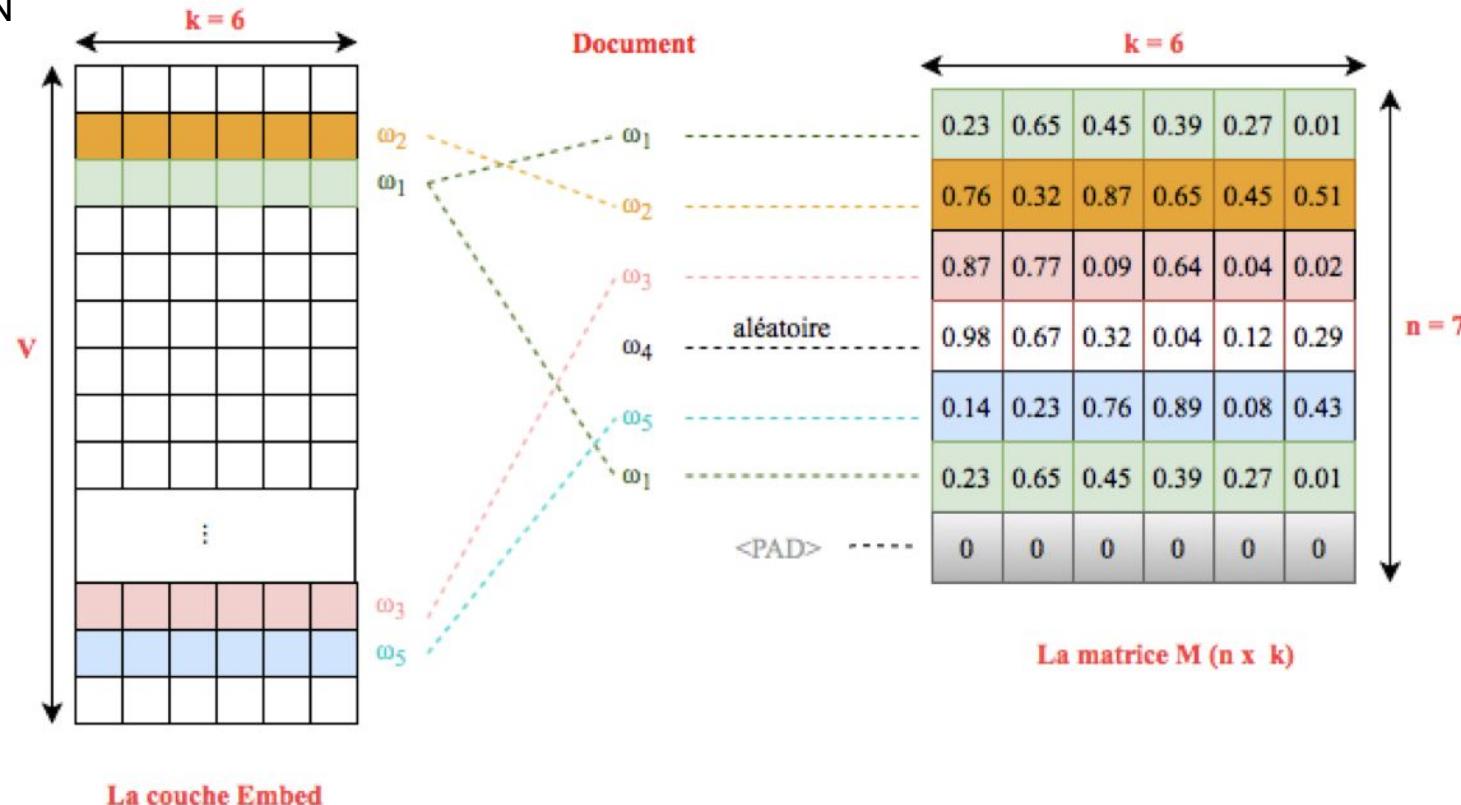
La couche Embed

CNN pour le texte

- Entrée du CNN
 - fixer n le nombre de mots dans le document d'entrée : la longueur du document.
 - Si la longueur du document est supérieure à n , alors il est nécessaire de tronquer les mots supplémentaires : c'est le tronquement (*truncating*).
 - Si la longueur l est inférieure à n , il est donc nécessaire d'ajouter $n-l$ fois un symbole spécial hors vocabulaire (par exemple: **<PAD>**): c'est le *padding*. Le vecteur correspondant à **<PAD>** dans la matrice M est initialisé à zéro. Il existe trois façons de faire: pre-padding/truncating, post-padding/truncating, ou sur les extrémités.

CNN pour le texte

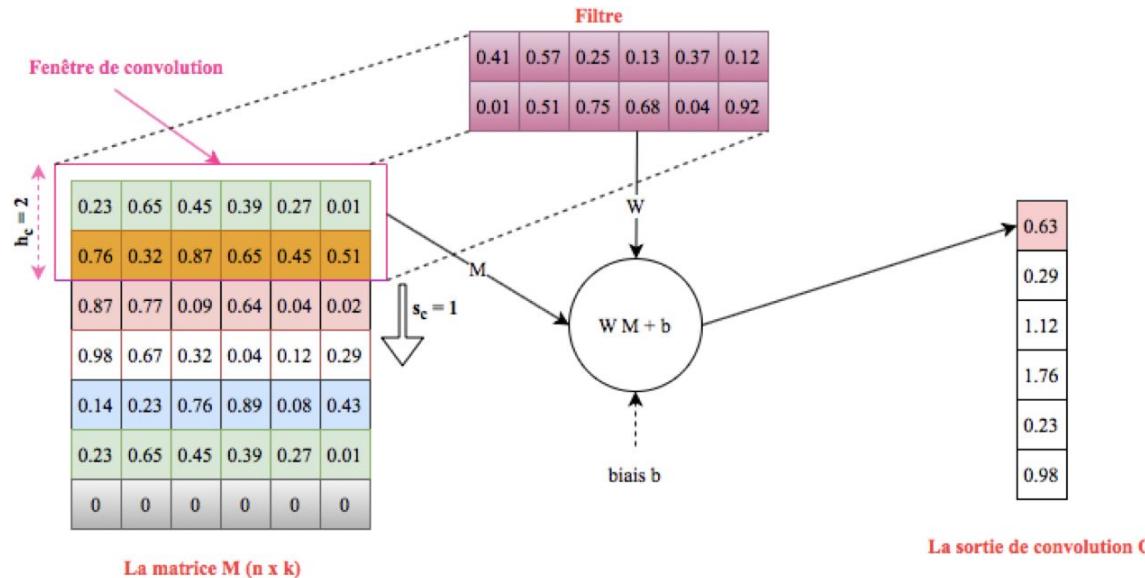
- Entrée du CNN



CNN pour le texte

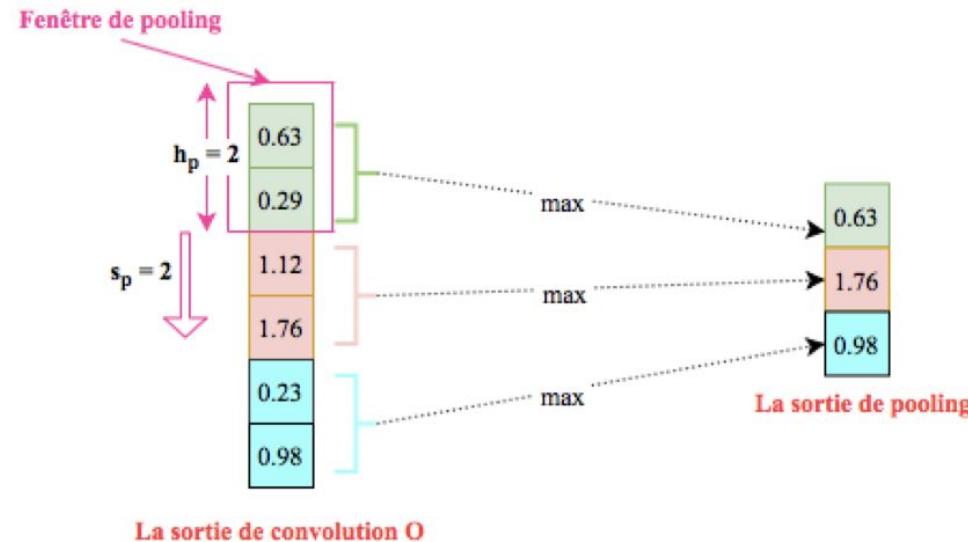
- Couche de convolution

- dimension 1 (en traitement de texte)
- opération mathématique fondamentale pour les CNN. Elle consiste à multiplier, ou *convoyer* la représentation matricielle M par une autre appelée matrice de convolution (ou filtre) pour produire une carte de caractéristiques (*feature map*).



CNN pour le texte

- Couche de pooling
 - dimension 1 (en traitement de texte)
 - opération permettant de réduire le nombre de paramètres en réduisant les dimensions des cartes de caractéristiques tout en gardant les informations les plus pertinentes.
 - 3 types de pooling : max-pooling, min-pooling et avg-pooling



CNN pour le texte

- Couches entièrement connectées (*fully connected FC*)
 - C'est une abstraction des entrées
 - La première couche FC après le pooling correspond à une concaténation des sorties de pooling.
 - Elle peut être éventuellement suivie par d'autres couches cachées. Les neurones de chaque couche FC sont liés à tous les neurones de la couche précédente et à tous ceux de la couche suivante.
 - Aucune connexion n'existe entre les neurones d'une même couche FC.

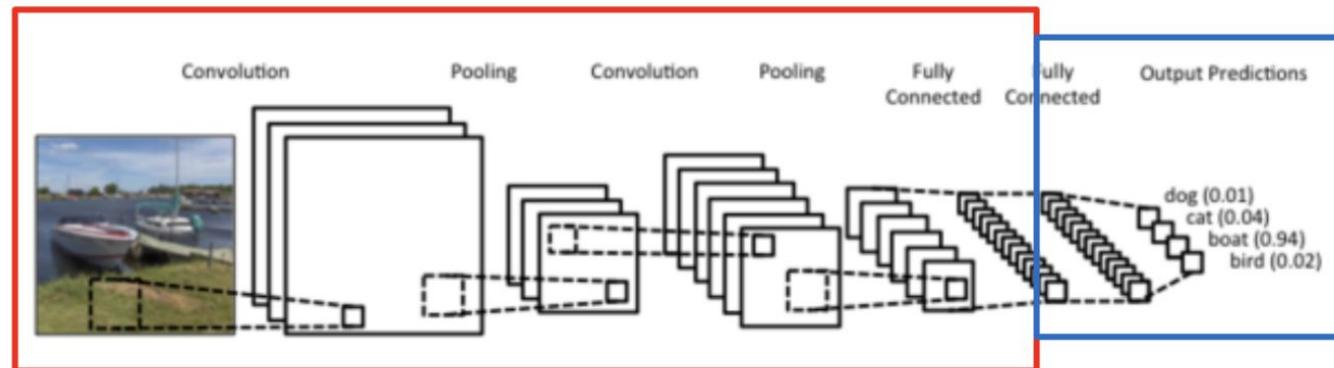
CNN pour le texte

- Couche de sortie
 - C'est une couche FC
 - Elle permet de prédire une classe ou une valeur continue selon la nature de la tâche : une régression ou une classification

CNN et transfer learning

- **Idée :** conserver l'extraction des caractéristiques apprises sur d'autres problématiques :
 - revient à conserver des couches de convolutions apprises sur un problème similaire
 - on ne change que les dernières couches FC

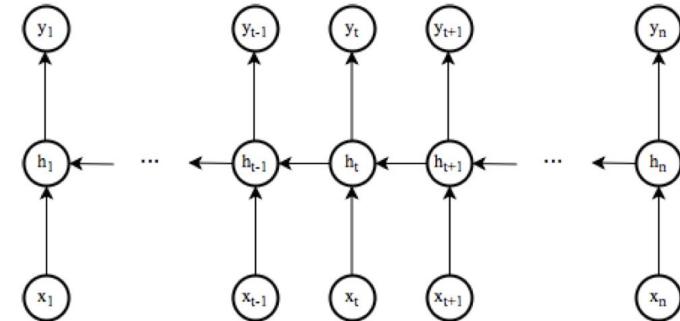
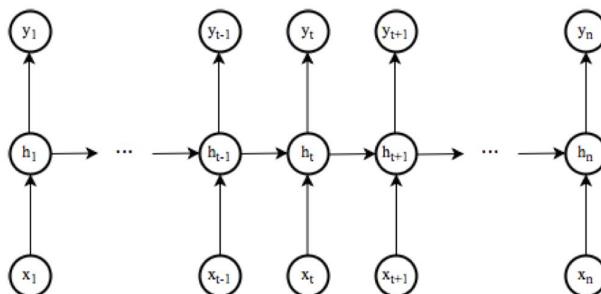
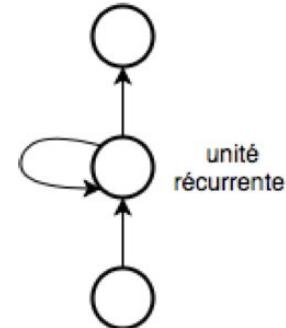
Fixée
(déjà
apprise)



On apprend
seulement ces
couches

- **Réseaux de neurones récurrents (RNN)**

- Problème : analyse de séquence (textes, enregistrements audio/vidéo)
- Ordre dans les données
- Les RNN prennent en input :
 - l'information actuelle
 - la sortie précédente
- Types: RNN avant, RNN arrière, LSTM, GRU



- **Auto-encodeurs**

- Problème : réduire la dimension des données
- Compression des données
- Considération des données comme inputs et outputs à la fois (pas de classe)
- Même nombre de neurones en couches d'entrée et de sortie

