M2 Informatique (VMI) – IFLCM010 – Image

Cours 1

Apprentissage profond & réseaux de neurones : Introduction

Enseignants : Camille Kurtz & Sylvain Lobry
xxx.yyy@u-paris.fr

21 septembre 2023

Université Paris Cité

Des éléments de ce cours sont empruntés à Geoffrey Daniel - CEA/Irfu/DAp

Plan

1. Aide à la décision et reconnaissance d'images
2. Réseaux de neurones (rappels)
3. Réseau neuronal convolutif (CNN)
4. D'autres familles de réseaux
5. TP

Plan

1. Aide à la décision et reconnaissance d'images
2. Réseaux de neurones (rappels)
3. Réseau neuronal convolutif (CNN)
4. D'autres familles de réseaux
5. TP

Principe

- Méthodes d'aide à la décision appliquées à la reconnaissance d'images
- La **reconnaissance des formes** = ensemble de techniques pour identifier des motifs à partir de données (par exemple des images) afin de **prendre une décision** dépendant de la **catégorie attribuée à ce motif**
- **Branche de l'IA** (apprentissage automatique)
- Forme ? très général, pas seulement **forme géométrique** mais plutôt motifs qui peuvent être de natures très variées : **contenu visuel** (code barre, visage, empreinte digitale) ou sonore, images médicales ou satellitaires.

Nombreuses approches

Approches globales *versus* approches locales

Approche globale : chaîne de traitement

Méthode globale

- **Caractériser une forme** et extraire des paramètres caractéristiques (*features*) de l'objet et les comparer par **classification ou mise en correspondance** à une base d'apprentissage
- Par cette méthode, difficile d'extraire plusieurs formes de la même image sans pré-traitement

Approche orientée "objet/région"

Approche locale : chaîne de traitement

Méthode à partir de point(s) d'intérêt

- Extraire des **points caractéristiques d'objets** comme les coins puis extraire des caractéristiques au voisinage de ce point
- Avec ces caractéristiques, il est possible d'extraire plusieurs objets et de faire la **reconnaissance de ceux-ci via un classifieur**

Extraction de points caractéristiques SIFT

Approche locale : chaîne de traitement

Étant donné un objet connu (à gauche) décrit par des points caractéristiques, recherche de cet objet dans une nouvelle image (à droite) où le contenu est inconnu

Approche locale : chaîne de traitement

Étant donné un objet connu (à gauche) décrit par des points caractéristiques, recherche de cet objet dans une nouvelle image (à droite) où le contenu est inconnu

Descripteurs de textures

Texture ?

- **Définition :** une structure spatiale constituée par l'organisation de primitives (ou motifs de base) ayant chacune un aspect aléatoire
- Une région issue de la segmentation peut être décrite par l'**analyse de sa texture**
- L'analyse de texture renvoie des informations sur l'**arrangement spatial des couleurs** ou des intensités dans tout ou partie de cette image

Local Binary Pattern (LBP)

Calcul du LBP :

1. Divisez l'image en cellules
2. Pour chaque pixel d'une cellule, comparez le pixel à chacun de ses 8 voisins. Suivre les pixels le long d'un cercle.
3. Quand la valeur du pixel central est > à la valeur du voisin, écrivez "0". Sinon, écrivez "1".
4. Calculez l'histogramme de la fréquence de chaque "numéro" survenant
5. Concaténer les histogrammes de toutes les cellules. => Cela donne un vecteur de caractéristiques pour toute la fenêtre.

Apprentissage machine pour classifier les images

Apprentissage machine (machine learning)

- **Classification :** étiqueter chaque donnée (ici une région) décrite par un descripteur – une/plusieurs valeurs numériques) en l'associant à une classe

Classification non supervisée (clustering)

Définition

- On ne dispose que d'exemples, **pas d'étiquettes**, et que le nombre de classes et leur nature n'ont pas été prédéterminés
- **Aucun expert n'est requis**
- L'algorithme doit **découvrir par lui-même la structure cachée des données**
- Exemple : K-Means clustering
- Le système doit – dans l'espace des caractéristiques – **classer les données en groupe homogènes**
- C'est ensuite à l'opérateur d'associer chaque cluster à une sémantique (groupe 1 => lézard)

Classification supervisée

Définition

- Si les **classes sont prédéterminées et les exemples connus**, le **système apprend à classer** selon un modèle de classement
- **Un expert doit préalablement étiqueter des exemples**
- 2 phases :
  1. **phase hors ligne, dite d'apprentissage** : déterminer un modèle des données étiquetées
  2. **phase en ligne, dite de test** : prédire l'étiquette d'une nouvelle donnée, connaissant le modèle préalablement appris

Méthode des k plus proches voisins

Idée

- On dispose d'**une base de données d'apprentissage de N entrées** (par exemple des régions préalablement étiquetées en différentes classes)
- Pour estimer le label associé à une nouvelle entrée/région x, la méthode des k plus proches voisins consiste à **prendre en compte les k échantillons d'apprentissage dont l'entrée est la plus proche** de la nouvelle entrée x

Classification supervisée

Autres approches

- Arbre de décision
- Support Vector Machine
- Régression logistique
- Réseaux de neurones

et maintenant les réseaux profonds (deep learning)

Approches classiques vs approches profondes

Applications diverses

Deep Learning Trends @ ICLR 2016

En vision machine

Des progrès rendus possibles grâce aux données annotées

Nécessite beaucoup de données d'apprentissage !

Localisation Vs Detection

Plan

1. Aide à la décision et reconnaissance d'images
2. Réseaux de neurones (rappels)
3. Réseau neuronal convolutif (CNN)
4. D'autres familles de réseaux
5. TP

Réseaux de neurones (classiques vs profonds)

Réseaux de neurones : rappels (1/3)

Un réseau de neurones

- Des neurones répartis en **plusieurs couches connectées entre elles**
- S'utilise pour résoudre divers problèmes, ici la classification : **le réseau calcule à partir de l'entrée une probabilité** pour chaque classe. La classe attribuée correspond à celle de score le plus élevé
- Chaque couche reçoit en entrée des données et les renvoie transformées. Pour cela, elle calcule une combinaison linéaire puis applique éventuellement une fonction non-linéaire, appelée fonction d'activation. **Les coefficients de la combinaison linéaire définissent les paramètres (ou poids) de la couche**

Output of neuron = Y= f(w1. X1 + w2.X2 + b)

Réseaux de neurones : rappels (2/3)

Un réseau de neurones

- **Construit en empilant les couches** : la sortie d'une couche correspond à l'entrée de la suivante.
- Cet empilement de couches définit la sortie finale du réseau comme le résultat d'une fonction différentiable de l'entrée
- **La dernière couche calcule les probabilités finales** en utilisant pour **fonction d'activation** la fonction logistique (classification binaire) ou la fonction softmax (classification multi-classes)

Réseaux de neurones : rappels (3/3)

Un réseau de neurones

- Une **fonction de perte (loss function) est associée à la couche finale** pour calculer l'erreur de classification. Il s'agit en général de l'entropie croisée
- Les valeurs des poids des couches sont appris par **rétropropagation du gradient** : on calcule progressivement (pour chaque couche, en partant de la fin du réseau) les paramètres qui minimisent la fonction de perte régularisée. L'optimisation se fait avec une descente du gradient stochastique

Backpropagation
+
Weights Adjusted

Plan

1. Aide à la décision et reconnaissance d'images
2. Réseaux de neurones (rappels)
3. Réseau neuronal convolutif (CNN)
4. D'autres familles de réseaux
5. TP

Réseau neuronal convolutif (CNN)

Architecture

Un CNN se compose de deux types de neurones, agencés en couches traitant successivement l'information :

- les **neurones de traitement**, qui traitent une portion limitée de l'image au travers d'une fonction de convolution
- les **neurones de mise en commun** des sorties dits de pooling (totale ou partielle)

⇒ L'ensemble des sorties d'une couche de traitement permet de reconstituer une image intermédiaire, qui servira de base à la couche suivante

Réseau neuronal convolutif (CNN)

2 blocs principaux (end-to-end)

- **Extracteur de caractéristiques** : opérations de filtrage par convolutions, plusieurs filtres en //, les valeurs des dernières feature maps sont concaténées dans un vecteur
- **Prise de décision** : à partir des vecteurs de caractéristiques issus des filtres, apprentissage d'un modèle (réseau de neurones classique) pour prédire une classe. Les valeurs du vecteur en entrée sont transformées (avec plusieurs combinaisons linéaires) pour renvoyer un nouveau vecteur en sortie. Ce dernier vecteur contient autant d'éléments qu'il y a de classes : l'élément i représente la probabilité que l'image $\in$ à la classe i

$\Rightarrow$ Les paramètres des couches sont déterminés par rétropropagation du gradient : le loss (l'erreur) est minimisée lors de la phase d'entraînement. Ces paramètres désignent les caractéristiques des images

Couches de convolution

- Repérer la présence d'un ensemble de caractéristiques dans les images reçues en entrée
- **Filtrage par convolution** : le principe est de faire "glisser" une fenêtre représentant la feature sur l'image, et de calculer le produit de convolution entre la feature et chaque portion de l'image balayée. Une feature est alors vue comme un filtre
- La couche de convolution reçoit donc en entrée plusieurs images (issues des convolutions précédentes), et calcule la convolution de chacune d'entre elles avec chaque filtre

Apprentissage des caractéristiques

Couche de pooling

- Réduire la taille des images, tout en préservant leurs caractéristiques importantes
- Souvent placée entre deux couches de convolution : elle reçoit en entrée plusieurs feature maps, et applique à chacune d'entre elles l'opération de pooling
- On découpe l'image en cellules régulières, puis on garde au sein de chaque cellule la valeur maximale
- Réduire le nombre de paramètres et de calculs dans le réseau. $\Rightarrow$ améliore ainsi l'efficacité du réseau et on évite le sur-apprentissage

Couches de correction (ReLU)

- ReLU (Rectified Linear Units) désigne la fonction réelle non-linéaire définie par

$$ReLU(x) = max(0, x)$$

- Remplace donc toutes les valeurs négatives reçues en entrées par des zéros. Elle joue le rôle de fonction d'activation
- On permet au CNN de rester en bonne santé (mathématiquement parlant) en empêchant les valeurs apprises de rester coincer autour de 0 ou d'exploser vers l'infinie

Architecture finale

Nombre de paramètres

18 + 18 paramètres

Nombre de paramètres : LeNet – 5

Gradient-Based Learning Applied to Document Recognition paper by Y. Lecun, L. Bottou, Y. Bengio and P. Haffner (1998)

Number of parameters : 60 thousands

Nombre de paramètres : AlexNet

AlexNet is another classic CNN architecture from ImageNet Classification with Deep Convolutional Neural Networks paper by Alex Krizhevsky, Geoffrey Hinton, and Ilya Sutskever (2012)

Number of parameters : 60 millions

Nombre de paramètres : VGG-16

VGG-16 from Very Deep Convolutional Networks for Large-Scale Image Recognition paper by Karen Simonyan and Andrew Zisserman (2014).

Number of parameters : 138 millions

Entrainement du réseau

- Distinguer les ensembles (découpés en 60% – 20 % – 20 %) : (1) **d'apprentissage** (mettre au point le modèle), (2) de **validation** (apprentissage de méta-paramètres), (3) de **test** (évaluer les performances du modèle appris)
- La machine va passer en revue chaque données du jeux d'apprentissage ⇒ **une epoch** (plusieurs epochs pour l'entrainement)
- Comment **choisir les hyper-paramètres ?** (nombre d'epochs et **taille du batch**, i.e. le nombre d'images que l'on va passer avant la rétro-propagation) ⇒ après chaque epoch la machine va tester son apprentissage en regard du jeu de validation

```
Test loss: 0.2022169417142868
```

Nombreux CNN existants

An Analysis of Deep Neural Network Models for Practical Applications, 2017.

Transfert learning / Fine-tuning

Tranfer learning

- **Idée :** conserver l'extraction des caractéristiques apprise sur d'autres problématiques (par exemple sur ImageNet)
- Revient à conserver des couches de convolution apprises sur un problème similaire
- On ne change que la (ou éventuellement les) dernière(s) couche(s) d'identification

Plan

1. Aide à la décision et reconnaissance d'images
2. Réseaux de neurones (rappels)
3. Réseau neuronal convolutif (CNN)
4. D'autres familles de réseaux
5. TP

GAN, auto-encoders, etc.

Résultats actuels en vision par ordinateur

- Visual recognition :
  `https://visual-recognition-demo.mybluemix.net/`
- Face detection (Compact CNN cascade) :
  `https://www.youtube.com/watch?v=Ad6GxIR8EpU`
- Real time object detection using deep learning :
  `https://youtu.be/HJ58dbd5g8g?t=4m17s`

Plan

1. Aide à la décision et reconnaissance d'images
2. Réseaux de neurones (rappels)
3. Réseau neuronal convolutif (CNN)
4. D'autres familles de réseaux
5. TP

Notebook Python pour le TP

https://github.com/mchelali/ImageClassificationWithDeepLearning/blob/master/Image%20Classification.ipynb

Université de Paris

Master 2 VMI - TP RF : Convolutionnal Neural Network (CNN)

Prérequis pour ce TP : disposer d'un compte Google

- Télécharger ce projet GitHub sur votre machine locale
- Uploader ensuite ce projet dans un dossier lié à votre compte Google Drive
- Ouvrir Google Collab et rechercher votre projet
- Vous pouvez alors exécuter les différentes étapes sucessivement de ce notebook en local sur le serveur Google Collab, ce qui vous évite de disposer d'un GPU en local

La classification

La classification est un precessus qui prend une entrée une donnée (par exemple, une image et ses pixels) et qui repond par une decision en sortie (chien, chat, ...) ou la probabiltée de chacune des classes considérées.

Les réseaux de neurones convolutionnels (CNN)

Les CNN sont des réseaux profonds composés de différentes couches dans le but d'extraire (d'apprendre) des caractéristiques permettant de différencier les classes traitées.