

Document Deep Learning

Présentation du module

L'apprentissage profond et en anglais **deep learning** est un ensemble de méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données s'appuyant sur les réseaux de neurones artificiels. Ces techniques ont permis des progrès importants et rapides dans les domaines de l'analyse du signal sonore ou visuel et notamment de la reconnaissance faciale, de la reconnaissance vocale, de la vision par ordinateur, du traitement automatisé du langage. Dans les années 2000, ces progrès ont suscité des investissements privés, universitaires et publics importants, notamment de la part des GAFAM (Google, Apple, Facebook, Amazon, Microsoft.
)[wikipedia]

Objectifs

L'objectif de ce module est de vous donner un premier aperçu du monde du deep learning et les outils pour pouvoir appréhender les bases de celui-ci.

A la fin de ce module, le but est que vous soyez capable de:

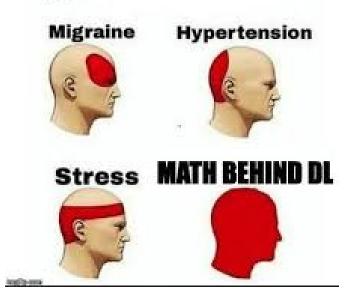
- Construire et implémenter un réseau de neurones
- Utiliser keras pour implémenter un reseaux de neurons et un réseau de neurons approfondi
- Utiliser un réseau de neurons convolutionnel (CNN) dans un cadre deep learning

Pré-requis

- Utilisation de python
- Utilisation des librairie sklearn, pandas
- Notions de machine learning

Projet étape 2 : Introduction aux réseaux de neurones convolutifs #2 (Jour 1)

Types of Headaches



Objectifs

- 1. Introduction CNN (Convolutional neural network) : architecture, couches, caractéristiques (convolutions), sortie
- 2. Construire un modèle CNN simple (2-3 couches)
- 3. Entraîner, évaluer et améliorer le modèle

Consignes

- 1. Nous allons commencer par un peu de lecture sur les réseaux de neurons convolutifs. Lisez attentivement la ressource R2.1.
 - Y a-t-il des concepts que vous n'avez pas compris ? Notez-les au tableau.
 - Les concepts seront vu ensemble avec le groupe.
- 2. Apres lecture de la ressource R2.1 et le tour de table, dessinez une architecture type d'un CNN à minimum 2 couches pour une image en entrée d'une taille de 32x32. Sur chaque couche notez les dimensions des tableaux et leur nom.
- 3. Suivez le tutorial de la ressource R2.2 jusqu'au chapitre "Using the same graph of layers to define multiple models"
 - O Implémentez le code donné dans l'exemple
 - O Vérifier son fonctionnement et visualisez quelques images de la base mnist
 - O Quelle est la précision de cet exemple ?
- 4. Et si on rendait notre réseau de neurones plus "deep" en rajoutant des couches convolutives (Conv2D) suivies de couches de pooling (MaxPooling2D). Quelle est la précision du modèle maintenant, Est-elle la même ?



Petit exemple pour vous aider dans la ressource R2.3

Deep Learning - Notions de base

5. Jouez sur les paramètres de votre choix pour améliorer l'estimation de ce CNN. Affichez les courbes de précision selon les itérations pour vos paramètres. Argumentez vos choix et si la prédiction peut être amélioré ou pas.



Trop easy ?! Essayez d'adapter votre CNN sur d'autres bases de données. Vous pouvez choisir parmi les milliers dispos ici :

https://www.kaggle.com/datasets

Ressources

- R2.1 https://medium.com/@CharlesCrouspeyre/comment-les-r%C3%A9seaux-de-neurones-%C3%A0-convolution-fonctionnent-b288519dbcf8
- R2.2 https://www.tensorflow.org/guide/keras/functional
- R2.3 https://becominghuman.ai/step-by-step-neural-network-tutorial-for-beginner-cc71a04eedeb

Ressources for parameter optimization

https://blog.floydhub.com/guide-to-hyperparameters-search-for-deep-learning-models/ https://towardsdatascience.com/simple-guide-to-hyperparameter-tuning-in-neural-networks-3fe03dad8594

Livrables

- ☐ L'implémentation du CNN avec keras
- ☐ Un fichier mémo ou vous placerez toutes les idées/les infos/ les graphs qui vous semblent pertinents

Projet étape 2 : Introduction aux réseaux de neurones convolutifs #2 (Jour 2)

Objectifs

- 4. Utilisation des modèles pré-entrainées en Deep Learning
- 5. Mettre en œuvre le Transfer Learning avec keras

Consignes

- 1. Lisez la ressource R2.4 jusqu'au chapitre "Transfer Learning" et suivez l'exemple donné dans le chapitre "Utilisation du VGG-16 pré-entrainé". Quel est le top 3 des classes les plus probables pour les images continues dans le dossier «Images_test »?
- 2. Testez le même module pour les modèles : MobileNetV2 et InceptionV3. Quels sont les résultats de ces modèles sur les images contenues dans le dossier.
- 3. Maintenant on va s'intéresser et appliquer tout ce qu'on a vu sur un challenge très connu dispo sur Kaggle "Cats vs Dogs challenge". Vous pouvez récupérer la base de données disponible dans le lien de la ressource R2.4. Le but sera de construire le meilleur réseau capable de reconnaitre si l'image montre un chat ou un chien.

Deep Learning - Notions de base

- a. Télécharger la base de données et charger la (R2.6). Explorez les données, vérifiez qu'il y a bien 25000 images, 12500 par catégorie. Créez une plus petite base de données que vous allez utiliser dans ce module : une base d'apprentissage (1500 à 2000 images), validation (500 images), test (500 images)
- b. Retravailler la base de données pour qu'on puisse appliquer notre CNN dessus : normalisez les images entre 0 et 1 ; transformer les noms des libellés en valeurs numériques ; affichez quelques images de chaque classe.
- 4. Maintenant, codez un CNN comme vu dans le module précèdent (vendredi). Vous pouvez choisir le nombre de couches que vous voulez, les tailles des convolutions ainsi que les méthodes d'optimisation que vous voulez. Quelle est la précision que vous obtenez ? Affichez les courbes d'erreur et de précision sur le set de test et de validation de chaque itération.
- 5. On va appliquer le concept de "Transfer Learning" sur notre base de données Cats vs Dogs. Vous pouvez choisir le modèle pré-entrainé que vous voulez parmi VGG16, MobileNetV2 ou InceptionV3 ; quelle est la précision du modèle choisi sur votre base de données test?
 - a. Le principe du *Transfer learning* est d'utiliser les connaissances acquises par un réseau de neurones lors de la résolution d'un problème afin d'en résoudre un autre plus ou moins similaire. On réalise ainsi un transfert de connaissances, l'intérêt n'est donc pas d'utiliser exactement le même modèle mais de l'adapter au nouveau jeu de données. Vous pouvez trouver les différentes stratégies pour faire cela dans la ressource R 2.4 (Chapitres « Apprenez à construire un CNN et gagnez du temps avec le Transfer Learning » et « Transfer Learning »). Appliquez les 3 types de stratégie. Quelle sont vos nouvelles performances ? Affichez pour chaque cas les courbes d'erreur et de précisions.

Ressources

 $R2.4\ \underline{https://openclassrooms.com/fr/courses/4470531\text{-}classez\text{-}et\text{-}segmentez\text{-}des\text{-}donnees\text{-}visuelles/}{5097666\text{-}tp\text{-}implementez\text{-}votre\text{-}premier\text{-}reseau\text{-}de\text{-}neurones\text{-}avec\text{-}keras}}$

R2.5 https://www.kaggle.com/c/dogs-vs-cats/data

R2.6 https://machinelearningmastery.com/how-to-load-large-datasets-from-directories-for-deep-learning-with-keras/

Livrables

- ☐ Mémo avec les réponses aux consignes 1 et 2.
- □ Support au choix pour les questions sur le challenge "Cats vs Dogs": doc ou jupyter notebook ou vous placerez toutes les idées/les infos/ les graphs qui vous semblent <u>pertinents</u>