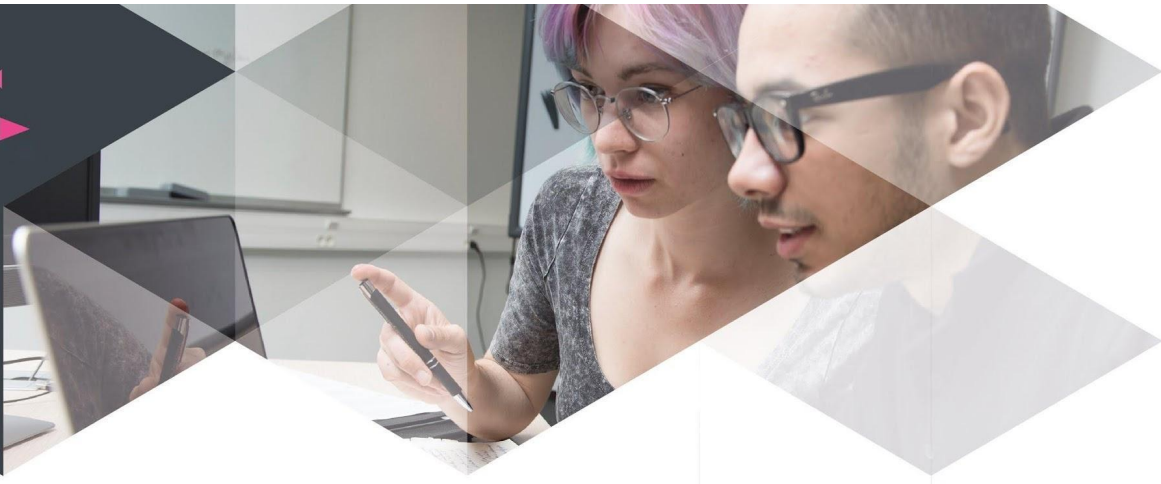




le
campus
numérique
in the ALPS



Deep Learning

1. Introduction aux réseaux de neurones convolutifs #2 (Jour 2)

Objectifs:

1. Utilisation des modèles pré-entraînés en Deep Learning
2. Mise en œuvre de Transfer Learning avec keras

Consignes:

1. Maintenant on va s'intéresser et appliquer tout ce qu'on a vu sur une nouvelle base de données. Vous pouvez récupérer la base de données disponible sur le drive. Le but sera de construire le meilleur réseau capable donner les meilleures performances selon le jeu de données choisi: <https://www.kaggle.com/datasets>.
ex : <https://www.kaggle.com/hasinisadunikasilva/skincancerdetectiondcnn>
 - a. Télécharger la base de données et charger là, utilisez la méthode donnée dans la Ressource 2.6 . Combien d'images avez-vous dans les datasets train, test et validation. Quelle est la taille des images ? Assurez-vous de mettre la même taille à toutes les images.
 - b. Retravailler la base de données pour qu'on puisse appliquer notre CNN dessus : normalisez les images entre 0 et 1.

2. Maintenant, codez un CNN comme vu dans le module précédent.
Vous pouvez prendre l'architecture avec laquelle vous avez eu la meilleure précision sur le dataset ou si elle est trop complexe (temps de calcul trop long) prenez une architecture très simple. 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.
Essayez d' "augmenter" votre jeu de données en utilisant ImageDataGenerator.
3. Lisez la ressource R2.5 jusqu'au chapitre "Transfer Learning" et suivez l'exemple donné dans le chapitre « Utilisation du VGG-16 pré-entraîné ».
Quel est le top 3 des classes les plus probables pour les images contenues dans le dossier data/images_test?
4. Testez le même module pour les modèles : MobileNetV2 et InceptionV3. Quels sont les résultats de ces modèles sur les images contenues dans le dossier?
5. On va appliquer le concept de "Transfer Learning" sur notre base de données. Vous pouvez choisir le modèle pré-entraîné que vous voulez parmi VGG16, MobileNetV2 ou InceptionV3.
Quelle est la précision du modèle choisi sur votre base de données test?
 - a. Trouver l'architecture d'un de ces réseaux et traduisez la en keras. N'essayez pas d'entraîner ce modèle, le GPU n'est pas assez puissant et ça va prendre des jours/semaines 😊.
 - b. 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 2.5 dans le chapitre « Transfer Learning ».
Appliquez les 3 types de stratégie.
Quelles sont vos nouvelles performances ?
Affichez pour chaque cas les courbes d'erreur et de précisions.

Ressources:

R2.5

<https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5097666-tp-implementez-votre-premier-reseau-de-neurones-avec-keras>

R2.6

<https://vijayabhaskar96.medium.com/tutorial-image-classification-with-keras-flow-from-directory-and-generators-95f75ebe5720>