

# **Deep Learning et réseaux de neurones : techniques avancées en traitement d'image**

## **Durée**

3 jours.

## **Participants**

Analystes des données, ingénieurs ML.

## **Prérequis**

Bonne connaissance et pratique du Machine Learning avec Python.

## **Description**

Formation avancée sur les réseaux de neurones

## **Objectifs pédagogiques**

À l'issue de la formation, le participant sera en mesure de :

- Maîtriser les bases théoriques et pratiques d'architecture et de convergence de réseaux de neurones
- Connaître les différentes architectures fondamentales existantes et maîtriser leurs implémentations fondamentales
- Maîtriser les méthodologies de mise en place de réseaux de neurones, les points forts et les limites de ces outils
- Connaître les techniques d'interprétation de réseaux de neurones
- Maîtriser l'utilisation de données non-supervisé

## **Démonstrations**

Multiples démonstrations à l'aide de notebooks utilisant Keras.

## **Programme**

### **Concepts fondamentaux d'un réseau de neurones**

- Rappel de bases mathématiques.
- Le réseau de neurones : architecture, fonctions d'activation et de pondération des activations précédentes...
- L'apprentissage d'un réseau de neurones : fonctions de coût, back-propagation, stochastic gradient descent...
- Modélisation d'un réseau de neurones : modélisation des données d'entrée et de sortie selon le type de problème.

- Appréhender une fonction par un réseau de neurones. Appréhender une distribution par un réseau de neurones.
- Data Augmentation : comment équilibrer un dataset ?
- Généralisation des résultats d'un réseau de neurones.
- Initialisations et régularisations d'un réseau de neurones : L1/L2 Regularization, Batch Normalization.
- Optimisations et algorithmes de convergence.

### **Outils usuels Machine Learning et Deep Learning**

- Outils de gestion de donnée : Apache Spark, Apache Hadoop.
- Outils Machine Learning usuel : Numpy, Scipy, Sci-kit.
- Frameworks DL : PyTorch, Keras

### **Convolutional Neural Networks (CNN)**

- Présentation des CNNs : principes fondamentaux et applications.
- Fonctionnement fondamental d'un CNN : couche convolutionnelle, utilisation d'un kernel, padding et stride...
- Classification et segmentation d'image
- Architectures CNN ayant porté l'état de l'art en classification d'images : LeNet, VGG Networks, Network in Network...
- Utilisation d'un modèle d'attention.
- Application à un cas de figure de classification usuel (texte ou image).
- CNNs pour la génération : super-résolution, segmentation pixel à pixel.
- Principales stratégies d'augmentation des Feature Maps pour la génération d'une image.
- Interprétations de CNN avec saliency maps, gradconv ou occlusion sensitivity

### **Non-supervisé : Autoencodeurs, VAE et GAN**

- Intérêt des données non-supervisées, techniques semi-supervisées
- Présentation des modèles générationnels Variational AutoEncoder (VAE) et Generative Adversarial Networks (GAN).
- Auto-encoder : réduction de dimensionnalité et génération limitée.
- Variational AutoEncoder : modèle générationnel et approximation de la distribution d'une donnée.
- Définition et utilisation de l'espace latent. Reparameterization trick.
- Fondamentaux du Generative Adversarial Networks.
- Convergence d'un GAN et difficultés rencontrées.
- Convergence améliorée : Wasserstein GAN, BeGAN. Earth Moving Distance.
- Applications de génération d'images ou de photographies, génération de texte, super résolution.

### **Recurrent Neural Networks (RNN)**

- Présentation des RNNs : principes fondamentaux et applications.

- Fonctionnement fondamental du RNN : hidden activation, back propagation through time, unfolded version.
- Évolutions vers les GRU (Gated Recurrent Units) et LSTM (Long Short Term Memory).
- Problèmes de convergence et vanishing gradient.
- Types d'architectures classiques : prédiction d'une série temporelle, classification...
- Architecture de type RNN Encoder Decoder. Utilisation d'un modèle d'attention.
- Applications NLP : word/character encoding, traduction.
- Applications vidéo : prédiction de la prochaine image générée d'une séquence vidéo.