

Un réseau de neurones Jupyterien...

Avec l'avènement des plateformes de type MLaaS [Machine Learning as a Service], la complexité des algorithmes se déporte petit à petit du côté du fournisseur de la solution. Il n'apparaît donc plus aussi complexe d'envisager la mise en place d'un algorithme d'intelligence artificielle (machine ou deep learning) au sein de ses activités.

Cependant, lorsque l'objectif d'un tel algorithme s'éloigne des modèles standards (accessibles notamment via des APIs en mode black box), il peut s'avérer nécessaire de développer une solution ex nihilo. En effet, la plupart des modèles pré-entraînés performants sont plutôt créés pour de la reconnaissance d'images ou de sons, ou de l'analyse du langage naturel.

Tout le monde ne possède pas les ressources financières et humaines d'une grande entreprise (GAFA ou banques et assurances), mais avec une équipe compétente, issue d'un mix entre ingénieurs et docteurs, et avec un budget raisonnable, il est possible de se donner les moyens de ses ambitions.

La Retail Marketing Platform d'Ysance (RMP) a développé ces derniers mois un réseau de neurones profond permettant de prédire l'évolution de parcours clients omnicanaux (probabilité d'achat, influence d'une campagne marketing sur les parcours). La suite de ce document est un retour d'expérience sur la mise en place de ce réseau, de la phase de recherche jusqu'à sa mise en production.

Après avoir défini les besoins fonctionnels et réalisé une analyse exploratoire poussée des données (deux étapes indispensables et cruciales à la réussite de tout projet), nous nous sommes concentrés sur la création d'un prototype.

Nous avons réalisé nos développements dans un notebook Jupyter (en Python) attaché à une machine virtuelle d'une plateforme de Cloud Computing. En phase exploratoire, nous n'avons pas forcément accès aux ressources techniques nécessaires (développeurs, techops) et avons donc préparé nous-mêmes notre machine, en prenant en compte la possibilité de pouvoir plus tard la déployer sur d'autres instances.

Une fois l'installation complète des logiciels et framework nécessaires (supervisor, TensorFlow, CUDA, cuDNN) dans les bonnes versions, i.e. des versions compatibles entre elles, nous avons dû paramétriser des règles de firewall (ouverture des ports pour Jupyter, TensorBoard et supervisor) ainsi que des règles d'accès : nous utilisons des données clients et il faut donc s'assurer du respect de la confidentialité ainsi que de la sécurité de ces dernières. Au démarrage de ce projet, il n'existe pas réellement d'images docker stables d'un tel environnement. Nous pouvons citer les GitHub ufoym/deepl et floydhub qui proposent désormais des images complètes pour le deep learning.

Nous avons également mis en place un système de surveillance de l'utilisation processeur et mémoire des cartes graphiques, et des alertes afin de contrôler le coût de nos essais (la gestion du budget est essentielle dans notre type de structure).

Contrairement à du machine learning, pour lequel la construction ad hoc de nouvelles variables explicatives est indispensable et demande un temps conséquent, le deep learning permet de s'affranchir de cette étape. En effet, les réseaux neuronaux intègrent cela dans leurs différentes couches.

Ayant choisi d'utiliser un réseau de neurones récurrents de type Long Short-Term Memory (LSTM) qui prend en entrée une séquence temporelle de longueur (donc de pas de temps) fixe, il nous a fallu tout de même pré-



parer la donnée en amont pour la transformer en séquences adaptées au réseau. Après avoir entraîné, vérifié et évalué notre prototype, nous avons lancé le processus d'industrialisation en passant tout d'abord par un environnement d'intégration, un clone de l'environnement de production.

Pour insérer ce modèle, ou plutôt ce produit, dans un système déjà complexe, nous avons dû nous adapter aux contraintes de production. Par exemple, en production nous ne pouvons pas nous permettre d'exécuter un notebook Jupyter sur une machine montée « à la main ». Dès lors, nous avons opté pour le Cloud ML engine de Google qui permet d'exécuter des programmes dans un environnement entièrement géré et qui enlève donc une partie de la complexité du déploiement, malgré certaines difficultés liées à la relative jeunesse du service et à des incompatibilités entre différentes versions de TensorFlow (ce framework évoluant très rapidement).

De plus cette brique fonctionnelle étant amenée à en remplacer une autre, nous ne pouvions pas nous permettre, même si elle fonctionne de façon optimale en intégration, de la déployer directement en production à la place de la brique existante. Nous avons choisi de réaliser un double-run pendant quelques temps afin de nous assurer du bon fonctionnement du modèle et de l'évaluation du coût de cette technologie en conditions réelles par rapport à l'existant.



Nous ne rentrons pas ici dans les détails de l'algorithme et l'optimisation de l'architecture du réseau et de ses hyperparamètres, ces éléments ayant nécessité un temps de développement relativement long en partie dû au caractère hautement déséquilibré de nos données.

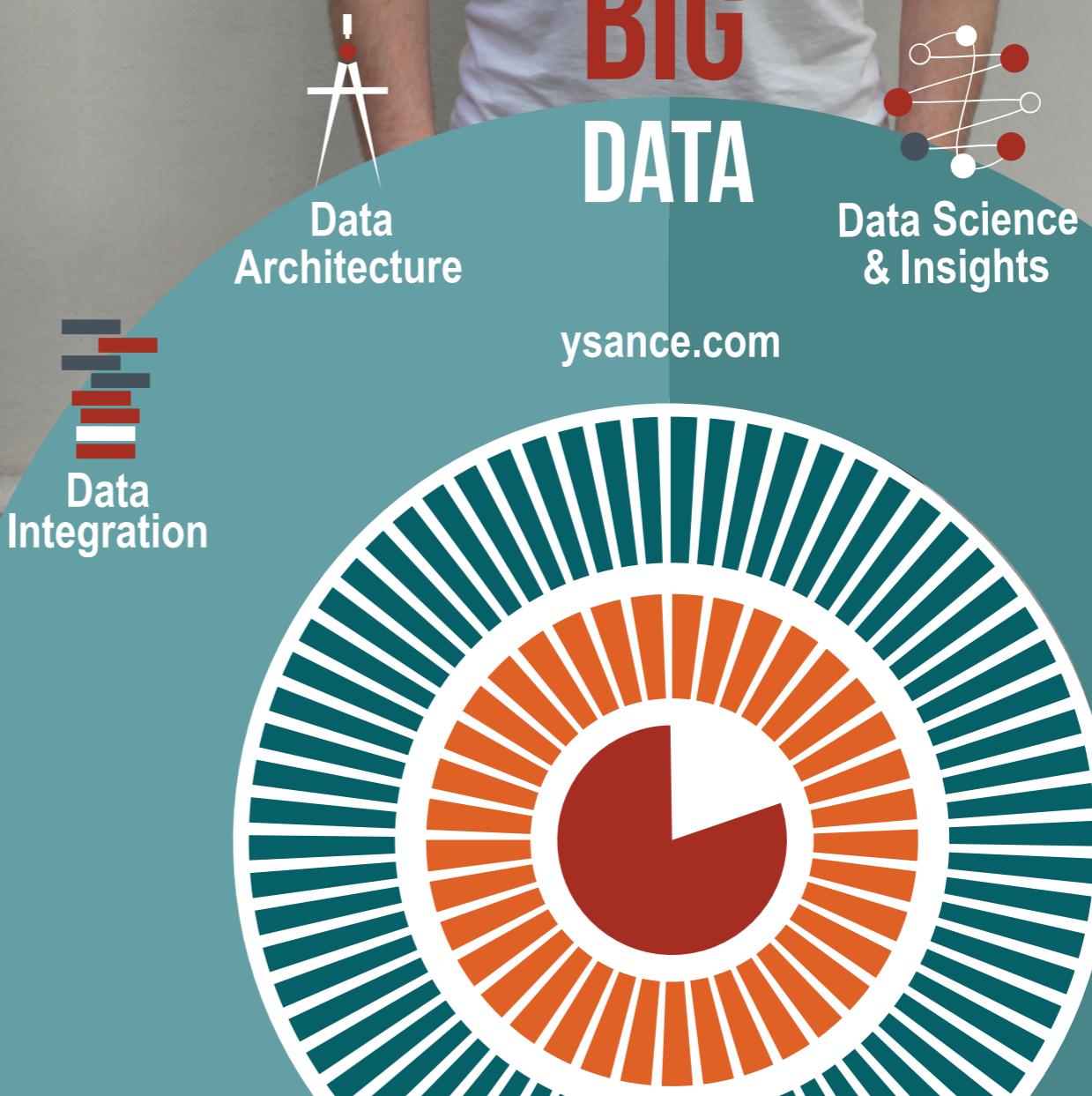
MLaaS est un outil puissant qui se perfectionne de jour en jour (pré-traitement automatique des données, BigQuery ML : du ML directement dans BigQuery et avec SQL) mais qui n'est pas forcément (encore?) adapté à toutes les structures ou aux multiples besoins fonctionnels que l'on peut rencontrer à la fois en tant qu'éditeur de logiciel ou en tant que consultant data scientist. Toutefois, la mise à disposition d'outils comme Cloud ML qui permettent de faire tourner ses propres modèles de façon relativement simple dans le cloud a permis une certaine démocratisation de ce type d'algorithmes. Attention néanmoins à ne pas entièrement se reposer sur un cloud provider ou un service en particulier au risque de se retrouver pieds et poings liés.

A propos de Thomas Gorbinet...

Thomas Gorbinet, spécialiste en analyse de données et apprentissage automatique est aujourd'hui Lead Data Scientist chez Ysance. Il est en charge du Data Lab, la cellule R&D de la RMP.

I'm a Data Badass.

Thomas, Lead Data Scientist chez Ysance



Dataviz & Analytics