Travail de lecture et de rédaction scientifique sur le Federate Learning

Bal Sébastien

15 août 2021

Remerciements

TABLE DES MATIÈRES

CHAPITRE

ONE

MACHINE LEARNING

Le Machine Learning appelé en Français apprentissage automatique "[...] est un champ d'étude de l'intelligence artificielle qui se fonde sur des approches mathématiques et statistiques pour donner aux ordinateurs la capacité d'apprendre à partir de données[...]".Ceci a pour objectif de traiter l'information afin de lui donner de la valeur ajoutée.

De nos jours, le Machine Learning est présent partout sur la toile, cela va du moteur de recherche comme Google, aux assistants vocaux comme Siri et Alexa, les fil d'actualités des réseaux sociaux comme Facebook et Twitter. Le point commun entre toutes ces platformes et le stockage massif des données de leur utilisateur appelé Big Data, comme le site le dictionnaire Larousse, "c'est un domaine technologique dédié à l'analyse de très grands volumes de données informatiques". Le Big Data est une technologie apparue dans les années 1900, elle a permis l'essor de l'apprentissage automatique, le Machine Learning. En effet, cet imposant volume de données collectées sur les utilisateurs a permis, dans les exemples cités ci dessus, de mieux cibler le comportement des utilisateurs et ainsi améliorer les expériences.

Pour fonctionner, le Machine Learning(ML) a besoin de données à ingérer et d'avoir un modèle appris afin de fournir des données à valeur ajoutée comme des algorithmes de prédiction de bourse et la maintenance prédictive. Il est donc nécessaire d'identifier les approches techniques pour créer ces modèles de ML.

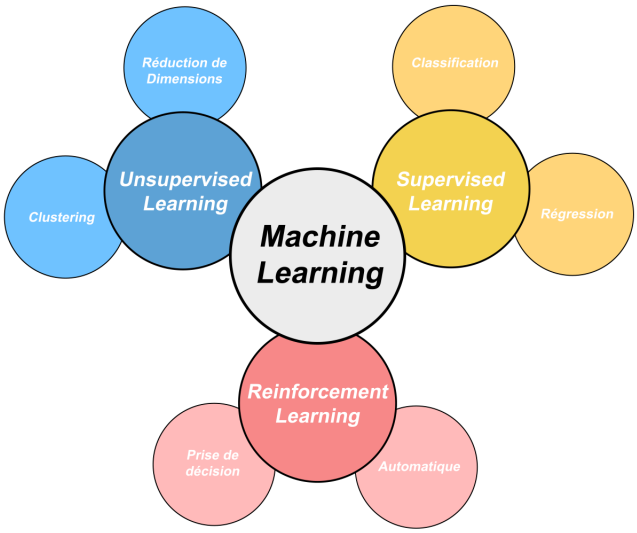

FIGURE 1.1: Familles d'algorithmes les plus utilisés

La première famille, l'appentrissage supervisé (en jaune dans la figure 1.1) consiste à donner des données en entrées, le résultat attendu itéré sur un grand jeu de données afin de trouver le modèle. Pour que le modèle deviennent performant, on fournit un grand volume de données dans le but qu'il se rapproche du modèle attendu. Ce type de modèle nécessite donc une bonne connaissance du processus métier vu qu'il est nécessaire de fournir des jeux de données en entrée pour obtenir le résultat désiré. Ce type de modèle est donc souvent utilisé pour simplifier ou optimiser des solutions existantes.

En prenant l'algorithme de l'arbre de classification et de régression, on peut constater que les données suivent un chemin qui a été défini au préalable par un développeur. Dans la fig 1.2, on peut remarquer le cheminement de l'algorithme sur un jeu de données.

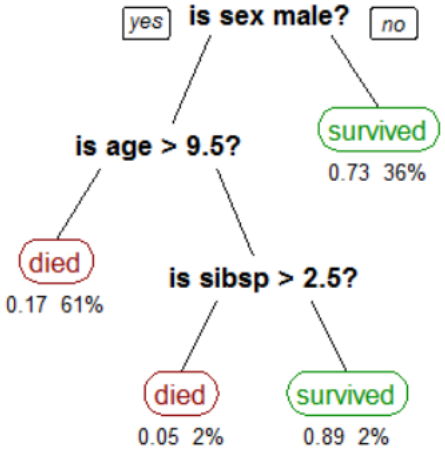

FIGURE 1.2: Abre de classiffication et de régression

La deuxième famille,l'apprentissage non-surpervisé (en bleu dans la figure 1.1) consiste à apprendre par identification des ressemblances et des différences entre les données fournies. L'algorithme rassemble les données en groupe ce qui permet lors de l'intégration d'une nouvelle information de la classifier dans un des groupes existants très rapidement. Ce type d'exemple a souvent pour objectif de créer des arbres de décisions sur base de "clustering". L'un des plus connus est le partionnement k-means, c'est un algorithme qui met en place un centre de gravité, dont les coordonnées vont servir pour localiser cette zone. Pour mieux comprendre l'algorithme, la notation $c^{(i)}$ représente la partition de point i et u_j le centre de la partition j. L'algorithme k-means répète l'étape suivante jusqu'à sa convergence fig 1.3.

$$\boxed{c^{(i)} = \arg\min_j ||x^{(i)} - \mu_j||^2} \quad \text{et} \quad \boxed{\mu_j = \frac{\sum_{i=1}^{m} 1_{\{c^{(i)}=j\}} x^{(i)}}{\sum_{i=1}^{m} 1_{\{c^{(i)}=j\}}}}$$

FIGURE 1.3: Algorithme k-means

Dans la fig 1.4, l'algorithme trie les données pour arriver à une convergence des données.

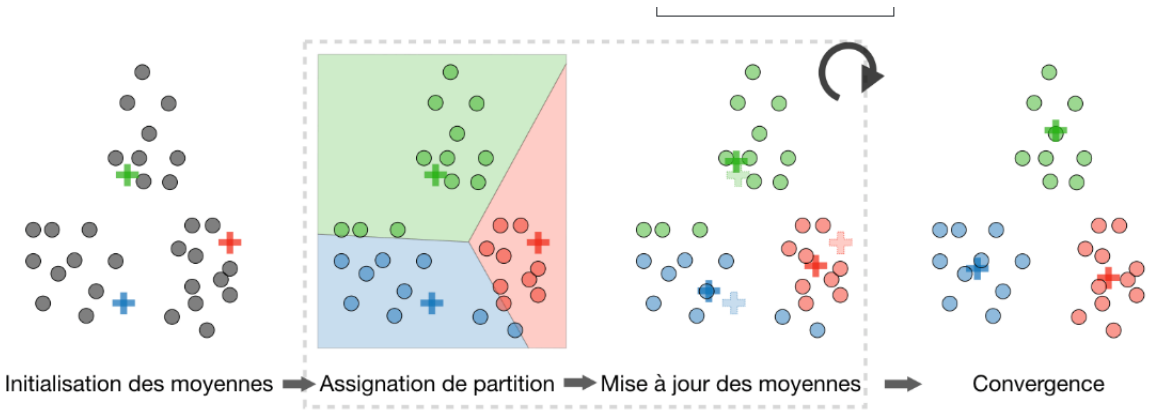

FIGURE 1.4: Shema k-means

Pour finir, il existe une catégorie qui gère sa propre expérience. En effet, l'apprentissage par renforcement (en rouge dans la figure 1.1) consiste à générer ses propres expériences. On se rapproche de l'automobile autonome, la machine change ses états suivant les actions qu'elle entreprend de faire. Un système de récompense positive et négative est mis en place pour constituer une nouvelle expérience et rendre la machine attentive pour maximiser ses chances de réussite. Ce type d'approche est la plus générique, elle convient aussi bien pour optimiser des solutions existantes que pour classifier des données. En règle générale, cette approche est souvent utilisée pour des recherches exploratoires. L'inconvénient principal de cettre approche scientifique est son besoin de ressources. En effet, les combinatoires peuvent très vites être importantes et pas conséquence, l'exploration aura besoin de temps ou de ressources physiques pour les tester.

Pour conclure, le ML est une technologie qui vise à trouver des modèles, comprendre des comportements afin de prédire les besoins d'une application suivant un besoin.

On peu constater que ce type de besoin se focalise sur une application spécifique cependant, le ML connait des limites au niveau des complexités combinatoire [x :], or certaines applications nécessitent des applications plus complexes avec plus d'entrées, tels que le traitement des images. Pour palier aux limites du ML, le Deep Learning a vu son essort [x : lien vers essort DL]

CHAPITRE

TWO

DEEP LEARNING

Le Deep Learning(DL) est représentatif d'un système neuronal comme notre système cérébral, il est conçus de plusieurs neuronnes qui intéragissent entre eux. Pour rendre performant tous ces neuronnes, il est nécessaire d'avoir une grande quantité de données, un Data Lake. Celui ci fournit au système plusieurs informations afin de lui constituer une "mémoire". Cette mémoire lui permet de reconnaitre des éléments bien particulier suivant l'entrainement qu'il aura suivit. Cet entrainement est supervisé par des développeurs qui vérifient que le DL ne sort pas de son modèles définis. Si il s'en éloigne, ils corrigent son algorithme mathématique pour le rendre plus performant et le remettre sur le droit chemin. Pour que le modèle mathématique deviennent performant, il faudra l'entrainer à reconnaitre une donnée en particulier. C'est le sujet de notre prochaine section.

2.1 Modèle d'entraînement

Pour entrainer le DL, il lui faut un algorithme mathématique et un grand flux de données afin d'affiner ses recherches et prendre de l'expérience. Dans notre situation, on décider de prendre la reconnaissance d'une image, en particulier celle d'un chat. On fournit à l'algorithme un flux d'images de plusieurs espèces d'animaux, on définit les paramètres qui permettent d'affirmer que l'image que l'on soumet soit bien un chat. Ainsi avec ces critères, l'algorithme devient plus précis car on le guide un peu sur l'objectif qu'il doit atteindre.

Voici un schéma pour l'exemple du fonctionnement du DL,[fig2.1], les boules vertes représente le bon chemin que le système va prendre pour arriver à vérifier le modèle qui était demandé. Les boules bleus sont celles qui ont des caractéristiques avec le modèle mais ne correspondra pas exactement au

modèle qui était demandé. Les boules rouges quand à elles, représentent les erreurs que le système a exclu pour pouvoir apprendre le modèle exacte. Les erreurs sont par la suite renvoyées en amont du système pour que le système ajuste son modèle mathématique

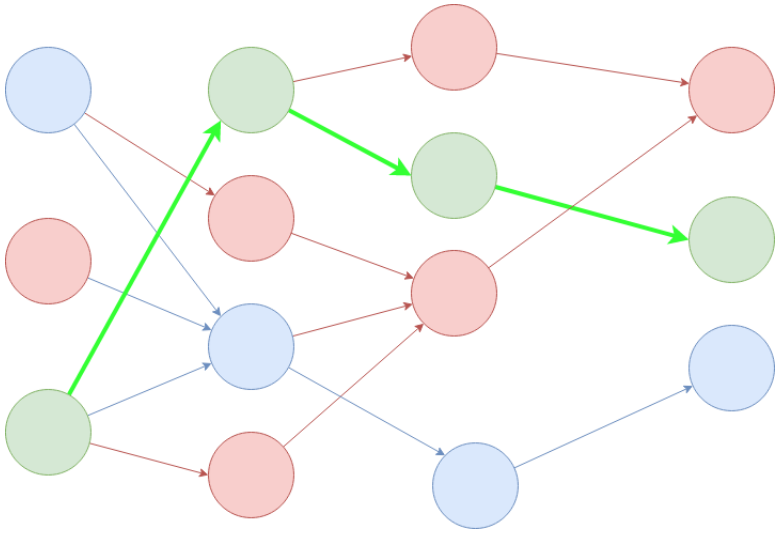

FIGURE 2.1: Autoapprentissage Deep Learning

Les avantages de cette méthode sont qu'elle permet de sortir une bonne qualité sur des résultats obtenus. Elle permet aussi d'avoir une meilleure exécution sur des tâches de routine tout en gardant de très bon résultat car traiter des données non structurées n'est pas chose aisée. Pourtant même avec des documents, des photos, des mails, elle arrive a sotcké des informations sous ces formats désordonnés. Cela montre bien la puissance du Deep Learning.

Bien entendu, il y a aussi des désavantage,on peut constater que ce type de modèle peut devenir très vite énergivore. De plus, le DL se focalise principalement sur une application. Or, il peut être intéressant d'étudier plusieurs applications similaires afin d'identifier des modèles plus poussés. C'est dans ce contexte que nous allons introduire le Federated Learning.

CHAPITRE

THREE

FEDERATED LEARNING

Avec la digitalisation des données et l'augmentation du nombre d'individus sur notre planète, on fait face à de nouveaux défis pour l'utilisation de toutes ces données présentes sur Internet. C'est pour cette raison que notre époque fait face à deux défis importants en terme d'avancer technologique :

1. Le respect de la vie privée. L'une des plus importantes est celle de la privatisation de ces données sur le web. Avec la protection des données (RGPD) promulgueé en 2018, les données privées font entièrement partie de l'utilisateur, ces données ne peuvent pas être utilisées sans l'accord de son propriétaire.
2. L'union fait la force, or, le traitement en silo des données de chaque entreprise freine énormément l'évolution des apprentissages de machine learning. En effet, en partageant les données entre différents secteurs, les algorithmes pourraient s'enrichir dans d'autres contextes afin de prendre de meilleures décisions en fonction du besoin de chaque application.

Lors de mes recherches, j'ai lu la citation de Benjamin Merci, Responsable Digital Analytics :
≪Sans connaissance de son audience, et sans hypothèses préalables solides, la Big data ne sert à rien≫.
Benjamin Merci veut mettre en avant au travers de sa citation l'importance de la collecte des données. La masse d'information nécéssaire pour obtenir des apprentissages de qualité demande beaucoup de rigueur et d'expérience dans le domaine ciblé.

Le Federated Learning peut se composer en deux familles bien distinctes :

1. La première est celle qui partage ses données sur un serveur centralisé.
2. La seconde est celle qui partage son modèle avec d'autres participants.

3.1 Apprentissage centralisé

Pour commencer, la première méthode se base sur des modèles précédents comme le ML et le DL. Cette approche a besoin d'un large éventail de données pour pouvoir améliorer son échantillonnage et ressortir des données suivant un besoin spécifique au niveau du modèle.

Son fonctionnement se constitue dans notre cas de mise à disposition de plusieurs appareils (des robots présents dans l'article : "Data-Driven Federated Learning for Spatio-Temporal Predictions in Multi-Robot Systems") connectés sur un réseau à un serveur centrale.

Dans cet exemple, les robots doivent partager leurs données collectées afin de les fusionner. Cette fusion des données permet d'apprendre avec une meilleure perception de leur environnement global. Ce type d'approche ne fonctionne que si les données ne sont pas sensibles. Au cas contraire, il est important de trouver une solution afin d'anonymiser ou pseudonomiser les données afin de les protéger.

De nos jours, il existe des approches de pseudonomisation très abouties comme les méthodes de chiffrement. Au travers de l'artile [X] "Extractop et gestion des connaissances, il est possible de constater l'utilité de la pseudonymisation en montrant la conservation d'informations sensibles à des fins scientifiques ou pour affiner des statistiques suivant un besoin. De plus, l'article met en avant que la méthode la plus courante pour la pseudonymisation est l'utilisation d'un chiffrement. Cette méthode utilise un identifiant et un quasi-identifiant et les remplace par d'autres chiffres ce qui rend l'identité cachée tout en rendant possible de ré-identification des informations en ayant la bonne clé de déchiffrement. Le schéma 3.1 montre le fonctionnement du chiffrement d'une pseudonymisation.

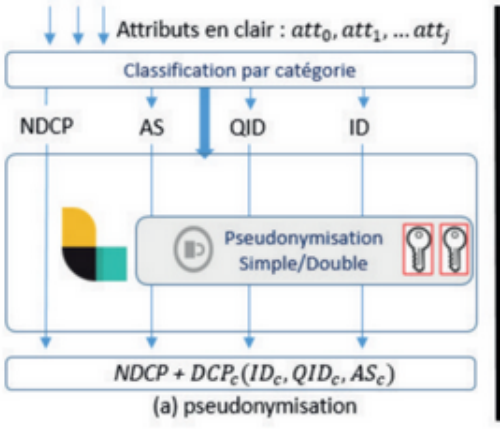

FIGURE 3.1: Pseudonymisation

Ainsi, la solution de l'apprentissage centralisé permet d'utiliser les technologies de plus en plus matures que sont le ML et le DL. Cependant, il faut être extrêmement vigilant à la protection des données d'apprentissage. Pour palier à ce problème de pratage des données sources, nous allons présenter le concept de partage des modèles d'apprentissages.

3.2 Modèle centralisé

Dans cette section, nous allons détailler comment le FL peut améliorer la convergence des apprentissages sans forcément avoir besoin d'accéder aux données sources.

C'est pour cettre raison que le modèle centralisé permet de partager ces modèles tout en faisant attention de ne pas divulguer des informations sensibles avec une protection des données. Cette approche de partage de modèle de prise de décision, comme nous le montre l'article "Federated Learning, Synthesis Lectures on Artificial Intelligence and Machine Learning", des entreprises on décider de partager leur modèles à des fins de détection d'objets. Ils forment ainsi un ensemble de modèles puissants dans le but de répondre rapidement aux demandes des clients. Le travail des modèles est illustré à la fig 3.2.

1. Chaque participant va rechercher sur le serveur le modèle de détection d'objet.
2. Chaque participant utilise le modèle sur leur donnée en interne.
3. Chaque participant envoie ses paramètres sur le serveur via un protocole sécurisé.
4. Le serveur utilise les paramètres du modèle de chaque participant et met à jour son propre modèle de détection d'objet.

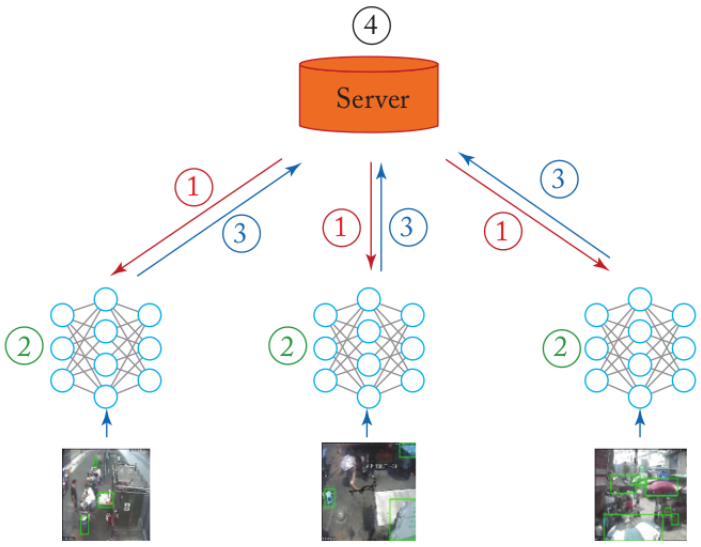

FIGURE 3.2: Modele partager

Une autre approche peut être nécessaire pour mieux comprendre son fonctionnement. Le FL se déroule en plusieurs étapes, la formation fédérée est celle qui est la plus courante comme nous le montre la fig 3.3. Pour commencer, un appareil mobile se procure un modèle afin d'acquérir les données et les traiter en local. En second lieu, le modèle est soumis à de multiples mises à jour régulières en local afin d'être amélioré, celles-ci contiennent des données locales qui appartiennent à différents appareils séparés. Ensuite, les appareils mobiles téléchargent des informations d'un champ de vecteurs présent dans un cloud. En troisième lieu, les modèles locaux effectuent une mise à jour moyenne au sein du cloud et est envoyé à un appareil défini comme le modèle mondial renouvelé. Pour finir, ces étapes précédentes se répètent afin d'arriver à un modèle avec une certaine performance ou qu'une date limite soit atteinte.

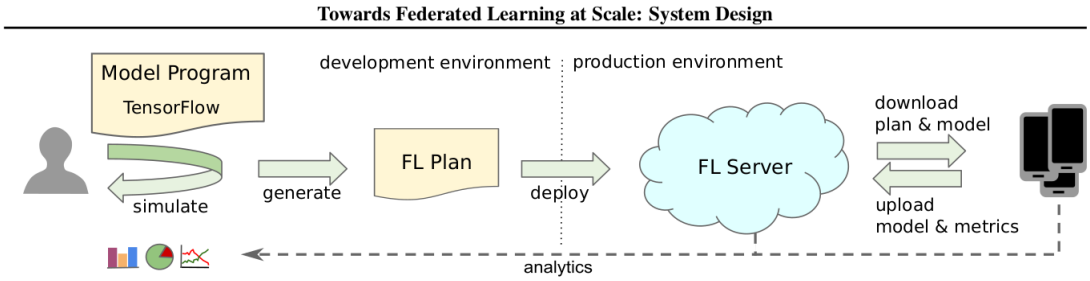

FIGURE 3.3: Schema FL

3.3 Exploration des modèles

Le FL est caractérisé par trois grands groupes qui surviennet régulièrement, ceux ci sont le FL Horizontal, le FL vertical et le Federated Transfer Learning. Les données stockées sont placées dans différents noeuds, celles ci sont sous forme de matrice de caractéristiques. Les données sont composées de multiples instances, la partie horizontale est représentée comme un client, celle verticale se caractérise par les caractéristiques du client. Pour finir, on peut séparer le FL suivant le mode de partition des données.

3.3.1 Horizontal FL

Pour celui du FL Horizontal, des chevauchements peuvent apparaitrent entre les caractéristiques des données qui sont réparties sur différents noeuds,

malgré le fait que les données ne sont pas semblables sur l'espace d'échantillonnage. Les données du point de vue de l'échantillonage sont différentes dans les scénarios des appareils connecté à Internet et les appareils intelligents. Mais ils sont particulièrment similaire dans un espace de fonctionnalité. Les mises à jours, comme nous l'avons vu précédement dans le point du FL, celle si est faite de façon horizontal. En effet, la dimension d'entité est la même pour chaque données. Dans les applications médicales, une forte hausse de travail est nécessaire pour collecté un grand nombre de données. Car il est difficile voir inconcevable pour chaque hôpital de créer une banque de données à partager. Via un réseau fédéral pour les hôpitaux construit par le FL permet d'améliorer le modèle(voir Fig 3.4).

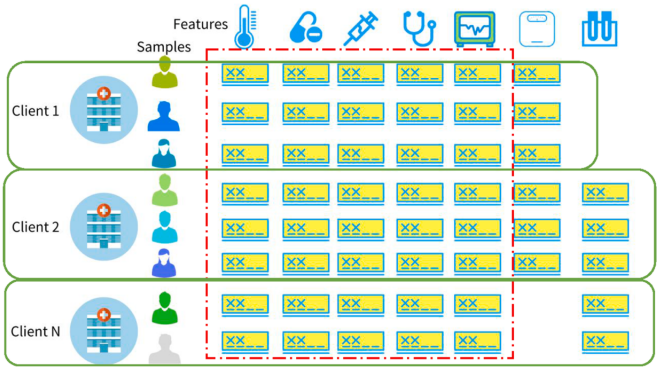

FIGURE 3.4: Horizontal FL

3.3.2 Vertical FL

Lorsque les données sont partitionnées, le FL vertical est plus approprié, chaque parties contient des données homogènes. Ce qui implique que les chevauchement se font en partie sur l'ID de l'échantillon alors qu'elle est différente dans l'espace fonctionnel. Prenons le cas d'un établissement médical, ils souhaitent identifier des maladies telles que le diabète. Il peut être analysé suivant certaines dimensions comme l'âge, le poids et les antécédants médicaux le type de diabète qu'un patient peut avoir. Avec le FL, certaines applications possèdent des données comme le nombre de pas ou la composition des plats qu'un personne a. Ces données peuvent être utilisées pour faciliter cette reconnaissance. La figure 3.5 illustre très bien le FL vertical.

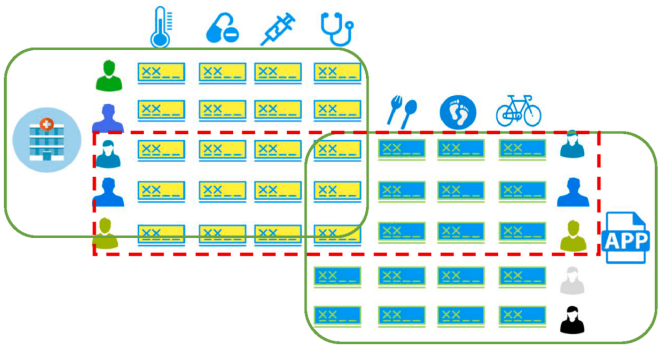

FIGURE 3.5: Vertical FL

3.3.3 Federated Transfer Learning (FTL)

En comparaison des deux scénarios vu précédement, dans la plupart des cituations, sur les espaces d'échantillonage et d'espace d'entité les données n'y sont pas partagées. Ce qui implique un manque d'étiquette et une mauvaise qualité des données. Le principe du FTL est transféré les données d'un domaine vers un autre afin de réaliser de meilleurs résultats d'apprentissage. Ainsi, le FTL permet d'avoir une application étendue pour ce qui est de parties communes avec des intersections étroites. Pour un système de réseaux de neuronnes qui possèdent une technologie de cryptage homomorphique peut empêcher des fuites de confidentialités mais aussi garder une très bonne précision.

Dans le cadre d'une application avec le modèle FedHealth rassemble un packet de données qui appartiennent à des organisations différentes via FL et permet d'avoir un service personnalisé pour des soins de santé. Sur la Fig 3.6, des informations sur le diagnostic et le traitement de certaines maladies peuvent être partagées avec un autre hôpital dans le but de faciliter des diagnostics. Les problèmes qui surviennent à l'heure actuelle dans le milieu de l'industrialisation sont les îlots de données et la protection de la vie privée mais avec le FTL, celui ci permet d'avoir un service de sécurité fiable pour la protection des données et la confidentialité des utilisateurs tout en cassant les barreaux des îlots de données.

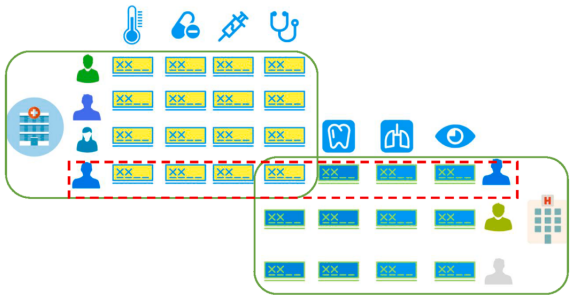

FIGURE 3.6: FTL

En quelques mots, c'est un apprentissage automatique distribué qui permet d'entraîner un modèle mathématique avec un large groupe de données décentralisées qui se trouvent sur des appareils distants.

3.4 Agrégation des modèles

L'esssort de l'industrie vient de l'utilisation du Federated Learning(FL) et c'est ainsi qu'elle a reussis à relever ces défis. L'utilisation d'îlots de données est la clé pour ces industries car le FL utilise un processus d'apprentissage automatique qui utilisent ces gros îlots. Cellui-ci utilise les concepts du ML et du DL pour pouvoir travailler avec ces grands échantillonnages de données. Des algorithmes de ML et de DL effectuent des calculs basé sur leur données de leur application. L'objectif du FL est de pouvoir agréger des informations de diversses applications, ou de différents acteurs afin de générer un modèle plus aboutie. Cette centralisation des données se fait généralement à l'aide d'un serveur centralisé comme le présente de schéma 1.3. Les données sont ainsi distribuées mais il reste la question de la privatisation des données et c'est là qu'intervient la pseudonymisation des données.

CHAPITRE

FOUR

EXEMPLES D'UTILISATIONS

Dans notre société, le FL peut être un atout majeur pour plusieurs secteurs comme le multimédia, la finance et les soins de santé. Plusieurs applications qui peuvent faciliter les employers dans leur travail de tous les jours ou encore aider à la reconnaissance de certaines maladies et bien entendu aider nos têtes blondes pour améliorer et faciliter leur apprentissage pour qu'ils deviennent les hommes et les femmes de demain. Beaucoup de possibilités et pour se faire, un aperçu des ses applications dans lesquels il est possible d'utiliser le FL.

4.1 Le Multimédia

Le secteur du multimédia fait partie intégrante de notre quotidien et nous ne pretons plus attention à toutes ses données misent sur le Web. L'article "Big and Personal : data and models behind Netflix recommendations" montre que le secteur du multimédia dispose de plusieurs données sur nos habitudes et nos préférences en terme de choix de visionnage. L'un des plus connu est Netflix, en effet cette société utilise nos données pour nous suggérer des choix parmis un large éventail de séries et de films. Ceux-ci se basent sur nos historiques, nos choix du moment, une autre de leur fonctionnalité est de partager les séries/films que nos amis aiment. Ainsi l'utilisateur pourra être diriger vers un visionnage qui pourra lui plaire. C'est l'un des exemples qui est présenté ici mais on peut imaginer plus loin cette utilisation. C'est pour cette raison que l'industrie a un enjeux capital dans tous ces îlots de données.

4.2 La Finance

Le secteur de la finance est très important avec des réglementations gouvernementales pour la protection des investisseurs contre la fraude et la mauvaise gestion, conserver la confidentialité et la sécurités des données des utilisateurs. Dans le but de réduire les coûts et la charge de travail, des sociétés banquaires et financières exploitent les technologies modernes comme l'IA, les services de cloud et la technologie sur les téléphone mobile pour assurer un service financier de qualité tout en respectant les réglementations gouvernemantales. Un cas bien particulier est le financement intelligent à la consomation, son objectif vise à prendre parti des techniques de ML afin de proposer des services financiers propre à chacun. Les données utilisées dans un crédit à la consommation sont fait d'informations sur les consommateurs comme le pouvoir d'achat, la préférence d'achat et aussi les caractéristiques du produit. Ces données peuvent être utilisée par diversses entreprises ou sociétés. Ainsi dans la figure 4.1, les informations de qualifications et le pouvoir d'achat peuvent être déduit de son épargne bancaire et de sa préférence d'achat sur des produits ou services. On fait face dans notre exemple à deux problèmes. Le premier est la protection de la vie privé des consommateurs et le sécurité des données barrière entre les banques, les réseaux sociaux, les sites d'e-commerce. Le deuxième, ce sont le stockages des données par trois sections hétérogènes, ce qui empêche le ML classique de fonctionner sur ces données hétérogènes. C'est pour cette raison que le FL résout ces problèmes. Le FL peut scinder les données en trois parties sans exposer les données. On peut aussi utiliser l'apprentissage par transfert pour examiner les données hétérogène.

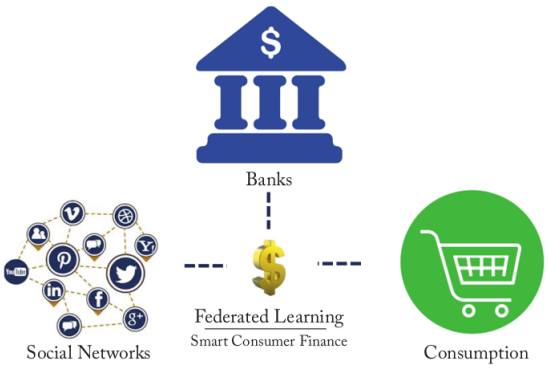

FIGURE 4.1: FL dans une petite entreprise financière

4.3 La santé

Dans le domaine médical l'intervention de l'IA peut aider à réduire les couts et les erreurs humaines dans les hôpitaux. En effet, un programme développé pour la diologie et la radiologie permet d'aider à poser un diagnostique sur des maladies cardiaques et identifier des cellules du cancer dans les premiers stades. Avec ces applications encourageantes de l'IA, beaucoup de fournisseurs de soins de santé tirent parti de l'IA pour augmenter l'éfficacité et améliorer les soins pour les patients. Cependant, l'utilisation de cette technologie n'en est qu'à ses débuts car il est déjà arrivé des erreurs suites à un mauvais diagnostique. Le plus gros problème dans ce domaine est la quantité d'informations suffisantes pour diagnostiquer précisément des symptômes sur un patient. Par exemple pour diagnostiquer un rhume, il faut plusieurs caractéristiques provenant de différentes sources, ainsi que les symptômes de la maladie.

C'est pour cette raison, que les institutions médicales doivent partagé leur données en respectant les règles de la protection de la vie privée. Par après, on peut former un grand ensemble de données qui fonctionne bien mieux que des données provenant d'un seul établissement. Si on fusionne le FL et l'apprentissage par transfert, cela promet une bonne solution pour réaliser cet objectif. L'apprentissage fédéré permet de former un modèle partager sans exposition et échange de données propre à un patient. Ensuite, les techniques d'apprentissage par transfert aide à élargir l'échantillonnage et améliore les performances de partages des données. Pour terminer, l'apprentissage par transfert joue un rôle considérable pour le développement des systèmes médicaux. Si plusieurs institutions médicales établissent une bonne base de données grâce à l'apprentissage fédéré, l'avenir de l'IA dans le domaine de la santé peut avoir un net avantage sur les maladies.

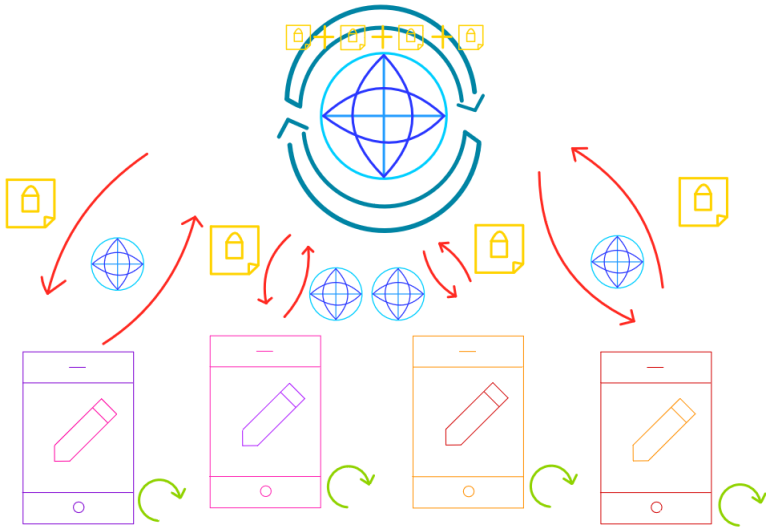

FIGURE 4.2: FL dans une petite entreprise financière

BIBLIOGRAPHIE

[1] Leslie Lamport, *A Document Preparation System*. Addison Wesley, Massachusetts, 2nd Edition, 1994.

[2] Li Li, Fan de Yuxi, Mike Tse, Kuo-Yi Lin, *A review of applications in federated leaning*. Comuters and Industrial Engineering, Volume 149, November 2020, 106854

[3] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, Han Yu *Federated Learning, Synthesis Lectures on Artificial Intelligence and Machine Learning*. Ronald J. Brachlan, Francesca Rossi and Peter Stone, Morgan and Claypool publishers, Series Editors December 2019

[4] Xavier Amatriain *Big and Personal : data and models behind Netflix recommendations*. Augustus 2013

[5] Jérôme Azé. *Extractop, et gestion des connaissances* `https://books.google.fr/books?hl=fr&lr=&id=dOEaEAAAQBAJ&oi=fnd&pg=PA419&dq=pseudonymisation+des+donn%C3%A9es&ots=EBRSMlwkfU&sig=Ss5Ik0PWRg_lR8K7-MVDwXPf2jA&redir_esc=y#v=onepage&q=pseudonymisation%20des%20donn%C3%A9es&f=false`

[6] Hausmane Issarane. *Apprentisage Supervise* `URL:https://analyticsinsights.io/5-apprentissage-supervise/` Consulté le 12/08/2021

[7] Retengr. *Deep Learning : définition, applications, avantages et inconvénients.* `https://www.retengr.com/2021/01/22/deep-learning-definitions-applications-avantages-inconvenients/` Consulté le 10/08/2021

[8] Bastien L. *Machine Learning et Big Data.* `https://www.lebigdata.fr/machine-learning-et-big-data` Consulté le 14/06/2021

[9] Afshine Amidi et Shervine Amidi. *Pense-bête d'apprentissage non-supervise.* `https://stanford.edu/~shervine/l/fr/teaching/`

`cs-229/pense-bete-apprentissage-non-supervise` Consulté le 14/08/2021

ANNEXE

A

ANNEXE

TOWARDS FEDERATED LEARNING AT SCALE: SYSTEM DESIGN

Keith Bonawitz [1] **Hubert Eichner** [1] **Wolfgang Grieskamp** [1] **Dzmitry Huba** [1] **Alex Ingerman** [1] **Vladimir Ivanov** [1] **Chloé Kiddon** [1] **Jakub Konečný** [1] **Stefano Mazzocchi** [1] **H. Brendan McMahan** [1] **Timon Van Overveldt** [1] **David Petrou** [1] **Daniel Ramage** [1] **Jason Roselander** [1]

ABSTRACT

Federated Learning is a distributed machine learning approach which enables model training on a large corpus of decentralized data. We have built a scalable production system for Federated Learning in the domain of mobile devices, based on TensorFlow. In this paper, we describe the resulting high-level design, sketch some of the challenges and their solutions, and touch upon the open problems and future directions.

1 INTRODUCTION

Federated Learning (FL) (McMahan et al., 2017) is a distributed machine learning approach which enables training on a large corpus of decentralized data residing on devices like mobile phones. FL is one instance of the more general approach of "bringing the code to the data, instead of the data to the code" and addresses the fundamental problems of privacy, ownership, and locality of data. The general description of FL has been given by McMahan & Ramage (2017), and its theory has been explored in Konečný et al. (2016a); McMahan et al. (2017; 2018).

A basic design decision for a Federated Learning infrastructure is whether to focus on asynchronous or synchronous training algorithms. While much successful work on deep learning has used asynchronous training, e.g., Dean et al. (2012), recently there has been a consistent trend towards synchronous large batch training, even in the data center (Goyal et al., 2017; Smith et al., 2018). The Federated Averaging algorithm of McMahan et al. (2017) takes a similar approach. Further, several approaches to enhancing privacy guarantees for FL, including differential privacy (McMahan et al., 2018) and Secure Aggregation (Bonawitz et al., 2017), essentially require some notion of synchronization on a fixed set of devices, so that the server side of the learning algorithm only consumes a simple aggregate of the updates from many users. For all these reasons, we chose to focus on support for synchronous rounds, while mitigating potential synchronization overhead via several techniques we describe subsequently. Our system is thus amenable to running large-batch SGD-style algorithms as well as Federated Averaging, the primary algorithm we run in production; pseudo-code is given in Appendix B for completeness.

In this paper, we report on a system design for such algorithms in the domain of mobile phones (Android). This work is still in an early stage, and we do not have all problems solved, nor are we able to give a comprehensive discussion of all required components. Rather, we attempt to sketch the major components of the system, describe the challenges, and identify the open issues, in the hope that this will be useful to spark further systems research.

Our system enables one to train a deep neural network, using TensorFlow (Abadi et al., 2016), on data stored on the phone which will never leave the device. The weights are combined in the cloud with Federated Averaging, constructing a global model which is pushed back to phones for inference. An implementation of Secure Aggregation (Bonawitz et al., 2017) ensures that on a global level individual updates from phones are uninspectable. The system has been applied in large scale applications, for instance in the realm of a phone keyboard.

Our work addresses numerous practical issues: device availability that correlates with the local data distribution in complex ways (e.g., time zone dependency); unreliable device connectivity and interrupted execution; orchestration of lock-step execution across devices with varying availability; and limited device storage and compute resources. These issues are addressed at the communication protocol, device, and server levels. We have reached a state of maturity sufficient to deploy the system in production and solve applied learning problems over tens of millions of real-world devices; we anticipate uses where the number of devices reaches billions.

[1]Google Inc., Mountain View, CA, USA. Correspondence to: Wolfgang Grieskamp <wgg@google.com>, Vladimir Ivanov <vlivan@google.com>, Brendan McMahan <mcmahan@google.com>.

Proceedings of the 2^{nd} SysML Conference, Palo Alto, CA, USA, 2019.