

Mémoire d'ingénieur

Mesure de la valeur de contribution des participants
dans un système d'apprentissage fédéré

Alexandre Bourbeillon

Année 2019–2020

Stage de fin d'études réalisé dans l'entreprise

Home Box Office

en vue de l'obtention du diplôme d'ingénieur de TELECOM Nancy

Maître de stage : François Charoy

Encadrant universitaire : Prénom Nom

Déclaration sur l'honneur de non-plagiat

Je soussigné(e),

Nom, prénom : Bourbeillon, Alexandre

Élève-ingénieur(e) régulièrement inscrit(e) en 3^e année à TELECOM Nancy

Numéro de carte de l'étudiant(e) : 06 66 91 56 25

Année universitaire : 2019–2020

Auteur(e) du document, mémoire, rapport ou code informatique intitulé :

Winter is Coming – You know nothing Jon Snow

Par la présente, je déclare m'être informé(e) sur les différentes formes de plagiat existantes et sur les techniques et normes de citation et référence.

Je déclare en outre que le travail rendu est un travail original, issu de ma réflexion personnelle, et qu'il a été rédigé entièrement par mes soins. J'affirme n'avoir ni contrefait, ni falsifié, ni copié tout ou partie de l'œuvre d'autrui, en particulier texte ou code informatique, dans le but de me l'accaparer.

Je certifie donc que toutes formulations, idées, recherches, raisonnements, analyses, programmes, schémas ou autre créations, figurant dans le document et empruntés à un tiers, sont clairement signalés comme tels, selon les usages en vigueur.

Je suis conscient(e) que le fait de ne pas citer une source ou de ne pas la citer clairement et complètement est constitutif de plagiat, que le plagiat est considéré comme une faute grave au sein de l'Université, et qu'en cas de manquement aux règles en la matière, j'encourrais des poursuites non seulement devant la commission de discipline de l'établissement mais également devant les tribunaux de la République Française.

Fait à Winterfell, le 31 juillet 2020

Signature :

Mémoire d'ingénieur

Mesure de la valeur de contribution des participants dans un système d'apprentissage fédéré

Alexandre Bourbeillon

Année 2019–2020

Stage de fin d'études réalisé dans l'entreprise Home Box Office
en vue de l'obtention du diplôme d'ingénieur de TELECOM Nancy

Alexandre Bourbeillon
numéro, rue
code postal, VILLE
téléphone
jon@castleblack.com

TELECOM Nancy
193 avenue Paul Muller,
CS 90172, VILLERS-LÈS-NANCY
+33 (0)3 83 68 26 00
contact@telecomnancy.eu

Home Box Office
numéro, rue
code postal, VILLE
téléphone

Maître de stage : François Charoy

Encadrant universitaire : Prénom Nom



Remerciements

Je tiens à adresser mes remerciements les plus sincères à l'administration de Telecom NANCY, en particulier Olivier Festor, Gérard Oster, Thibault Cholez et Michele Tartari pour leur soutiens durant la période de confinement et l'ensemble du stage.

Je tiens également à remercier François Charoy pour son soutien indéfectible, tant professionnel que moral qui m'a permis d'accomplir les objectifs de ce stage dans les meilleures conditions possible.

Finalement, j'adresse mes remerciements amicaux à Sebastien Da Silva, Christophe Bouthier, Yann Chaudun, Maialen Coterreau et Elise Klein pour leurs soutiens personnel et la détente qu'ils ont pus m'apporter.

– Alexandre Bourbeillon

Table des matières

Remerciements	v
Table des matières	vii
1 Introduction	1
2 Problématique	3
3 Notation et Etat de l’art	5
3.1 notations	5
3.1.1 Ecriture des éléments séquentiels et distribués	5
3.1.2 Notations sur l’apprentissage automatique	5
3.1.3 Evaluation d’un modèle	5
3.2 Etat de l’art	6
3.2.1 Apprentissage automatique	6
3.2.2 Mesure de la qualité des contributions dans un systèmes distribués	7
3.2.3 Apprentissage fédéré	8
3.2.4 Règles d’agrégation sécurisé	9
3.2.5 Méthode d’évaluation des noeuds	10
3.2.6 Frameworks d’apprentissage fédéré	11
3.3 Objectif et définition	15
3.3.1 Formalisation	15
3.3.2 Evaluer la distance entre les noeuds	15
3.3.3 Définition d’un critère d’évaluation	16
4 Conclusion	17
Bibliographie / Webographie	19
Liste des illustrations	21

Liste des tableaux	23
Listings	25
Glossaire	27
 Annexes	 30
A Première Annexe	31
B Seconde Annexe	33
Résumé	35
Abstract	35

1 Introduction

L'avènement du big data et des technologies cloud a permis l'explosion de la production de données durant les dernières années. La production d'une telle quantité de données ainsi que l'augmentation très rapide de la puissance des machines informatiques a ainsi contribué au développement très important des technologies d'intelligence artificielle. En particulier, l'apprentissage automatisé, avec comme figure de proue les réseaux de neurones profond, est maintenant utilisé dans un grand nombre d'applications industrielles comme l'analyse d'images, l'analyse de risque dans l'assurance de prêt (credit scoring) ou encore la production de diagnostics médicaux automatisés.

Pour entraîner des algorithmes d'intelligence artificielle précis, il est nécessaire d'utiliser de très nombreuses données annotées. Cependant, la production et l'utilisation de telles données pose des problèmes de protection de la vie privée des individus. En effet, certaines données, comme un dossier médical ou un dossier bancaire, peuvent être qualifiées de sensibles dans la mesure où leur utilisation peut affecter la vie de leur propriétaire. Un exemple connu de ce phénomène est le prix d'une assurance vie qui peut varier en fonction des antécédents médicaux de la personne qui les contracte. Bien que des techniques d'anonymisation des données existent, plusieurs études ont montré que celle-ci peuvent être facilement contournées en utilisant du recoupement de bases de données. La protection de l'anonymat des données des utilisateurs est donc un facteur très important à prendre en compte lorsque l'on construit des algorithmes de big data et d'intelligence artificielle.

Dans le but d'entraîner des modèles sur des données sensibles sans compromettre leurs anonymats, de nombreuses recherches ont été menées durant les 10 dernières années. En particulier, il a été proposé l'utilisation de l'encryption homomorphe, qui permet d'utiliser un modèle d'apprentissage automatique sur des données cryptées. Cette méthode permet donc de faire une prédiction sur des données sans que celle-ci soit exploitable par un tiers. Cependant, l'entraînement du modèle d'apprentissage doit être réalisé sur une base de données en clair. De ce fait, cette technique ne règle qu'une partie du problème de protection de l'anonymat dans la mesure où les données d'entraînement peuvent potentiellement être utilisées par un tiers.

Plus récemment, une nouvelle méthode d'apprentissage distribué a été proposée : l'apprentissage fédéré. Le principe de cette méthode est de délocaliser l'entraînement du modèle d'intelligence artificielle à l'endroit où les données sont produites. L'idée de cette méthode est que chaque producteur de données entraîne un modèle localement, puis le partage avec les autres producteurs. Plus précisément, chaque producteur entraîne un modèle localement qui est combiné avec les autres modèles en utilisant une règle d'agrégation (par exemple une moyenne pondérée). Avec cette méthode, chaque membre de la fédération conserve ses données mais partage le modèle qu'il produit. Ceci permet donc d'entraîner un modèle sur de très nombreuses données tout en protégeant l'anonymat des données. L'application à l'initiative de cette technologie a été proposée par Google AI et son but était l'entraînement d'un clavier intelligent sans le partage des données de saisie des utilisateurs. Depuis, plusieurs projets de framework à l'initiative d'entreprises comme Apple,

Google, IBM ou WeBank ont été lancés et sont encore en cours de développement.

D'autres cas d'usages ont également été proposés. Par exemple l'entraînement d'algorithmes de diagnostic médicaux à partir d'une fédération d'hôpitaux. Dans ce cadre, il est intéressant de trouver des méthodes pour motiver les utilisateurs à participer à l'entraînement. Pour cela, il est nécessaire de pouvoir mesurer la performance de chaque membre de la fédération. Ceci permet en effet d'implémenter des mécanismes de récompense en fonction de la qualité de ce qui est produit par un membre. Une telle méthode peut également servir de mécanisme de contrôle des membres de la fédération.

Actuellement, la question de la mesure de la performance des nœuds dans un système d'apprentissage fédéré est encore ouverte dans la recherche, bien que certaines solutions commencent à apparaître.

L'équipe Coast de l'INRIA Nancy est une équipe de recherche spécialisée dans les systèmes distribués et décentralisés. Elle étudie notamment la mesure de la confiance dans les nœuds de divers systèmes distribués. Le stage de fin d'étude dont ce mémoire fait l'objet s'inscrit dans le cadre de ces recherches. L'objectif de ce stage était en effet d'évaluer la valeur de la contribution dans un système d'apprentissage fédéré.

2 Problématique

L'apprentissage fédéré représente une nouvelle manière d'envisager l'intelligence artificielle. De ce fait, elle se heurte à de nombreux défis, comme par exemple la sécurité du système face à des attaques visant à influencer le résultat de l'entraînement (attaques Bizantines). Celle-ci permet de différencier un bon contributeur d'un mauvais contributeur. Un problème plus avancé consiste à mesurer une valeur de contribution pour chaque nœud du système.

Il est important de résoudre ce problème pour pouvoir attribuer une juste récompense à chaque participant. En effet, un tel mécanisme est nécessaire pour pouvoir implémenter un système de récompense des utilisateurs du système. Il permettrait de plus aux utilisateurs de vérifier que chacun contribue de manière équivalente au système. Ce genre de mécanisme permet donc d'encourager la participation des utilisateurs du système.

Il s'agit cependant d'un problème complexe dans la mesure où il est difficile de définir une mesure de la performance d'un modèle simplement dans le cas de l'apprentissage fédéré. En effet, le principe de l'apprentissage automatique est de construire un modèle des règles sous-jacentes à un ensemble de données. Pour mesurer la performance d'un modèle, on l'évalue donc simplement sur des données dont la distribution est équivalente au jeu de données qui a servi à l'entraîner. Dans le cas de l'apprentissage fédéré, les données sont séparées en plusieurs jeux de données. De ce fait, la performance d'un modèle n'a pas la même signification selon le nœud choisi pour l'évaluation. On dit que ces données sont non indépendantes et non identiquement distribuées (non iid).

Il est cependant impossible d'évaluer tous les modèles produits sur tous les nœuds du système. Dans certains cas d'application, un système d'apprentissage fédéré peut contenir plusieurs milliers de nœuds, avec parfois des puissances de calcul très limitées.

Il est donc nécessaire d'utiliser un processus de sélection de nœud de sorte que les nœuds choisis pour l'évaluation respectent la distribution de données de l'ensemble du système.

Dans le cadre des recherches de l'équipe COAST, nous nous intéressons au système distribués pair à pair en particulier. De ce fait, l'algorithme qui sera mis en place doit donc respecter la contrainte d'être compatible avec un système pair à pair.

L'objectif de ce projet de fin d'études est donc la mise en place d'un algorithme distribué permettant l'évaluation de la performance des nœuds d'un système d'apprentissage fédéré en respectant des contraintes de complexité algorithmique et d'implémentation dans un système pair à pair.

Pour la réalisation de cet objectif, il est dans un premier temps nécessaire de s'intéresser à l'influence de la distribution des données sur la variation de performance entre les nœuds. Après quoi il sera nécessaire de développer et d'implémenter une métrique permettant l'évaluation de la performance de chaque nœud. Il sera ensuite nécessaire de proposer un protocole expérimental permettant d'attester de l'efficacité de la métrique produite et de la comparer aux méthodes de

l'état de l'art. Enfin, il serait intéressant de vérifier que l'algorithme est résistant face à certain schéma d'attaques byzantine.

3 Notation et Etat de l'art

L'objectif de ce chapitre est de proposer un ensemble de notations. On proposera ensuite un état de l'art sur les techniques d'apprentissage automatique et fédéré. Enfin, nous terminerons par une évaluation des outils d'apprentissage fédéré existants.

3.1 notations

Cette partie a pour but d'expliquer les notations utilisés dans ce documents ainsi que leurs intérêt.

3.1.1 Ecriture des éléments séquentiels et distribués

L'apprentissage fédéré est une technique d'apprentissage séquentielle et distribué. De ce fait, nous utiliserons les notations suivantes pour décrire un élément distribué et un élément séquentiel :

- $(.)^n$ désigne un élément à la round n
- $(.)_x$ désigne un élément au noeud x

Par exemple, on désignera le dataset du noeud x par la notation D_x . De même, on désignera la mise à jour du modèle produite à l'étape n par ω^n .

3.1.2 Notations sur l'apprentissage automatique

L'apprentissage automatique utilise plusieurs concept comme

- **Dataset** : On note D un Dataset quelconque
- **Modèle** : On note M un modèle quelconque
- **Coefficient** : On note ω les coefficients d'un modèle M

Pour simplifier la compréhension, on pourra utiliser la notation $M(x)$ pour désigner l'évaluation de la données x par le modèle M

3.1.3 Evaluation d'un modèle

L'objectif de notre protocole est de permettre l'évaluation d'un modèle produit localement par d'autres noeuds. L'évaluation d'un modèle d'apprentissage automatique m s'effectue sur un jeu de données d . On la note $E_m(d)$.

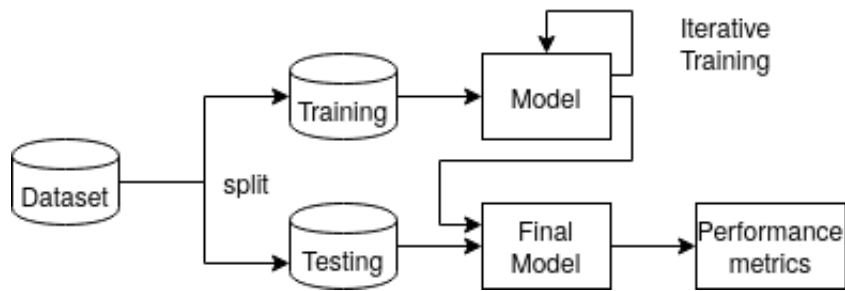


FIGURE 3.1 – Processus de fonctionnement de l'apprentissage automatique

Par exemple, l'évaluation du modèle du noeud x sur les données du noeuds y à l'étape n sera noté $E_{M_x^n}(D_y^n)$.

3.2 Etat de l'art

3.2.1 Apprentissage automatique

L'apprentissage automatique, que l'on appelle également l'apprentissage statistique, est une méthode d'intelligence artificielle supervisé dont l'objectif est de produire un *modèle* de comportement à partir d'un jeu de données. L'objectif est d'extraire un comportement général à partir de ces données pour produire une prédiction sur de nouvelles données. Pour cela, les données d'entraînement sont étiquetées avec le résultat attendu en retour de l'agorithme. L'entraînement consiste à comparer le résultat renvoyé par l'agorithme au résultat attendu, puis de modifier le comportement de l'agorithme pour minimiser l'écart entre ces deux résultats. La figure 3.1 décrit un processus d'entraînement classique d'un algorithme d'apprentissage automatique.

Cette technologie possède de nombreux domaines d'application comme la classification de dossier de crédit ou la détection de motif sur des images (par exemple dans l'imagerie médicale).

De nombreuses algorithmes d'apprentissages se sont développés durant les dernières années :

- Methodes de regression
- Arbres de décisions (DT)
- Machine à vecteurs de support (SVM)
- Algorithme des k plus proches voisins (kNN)
- Réseaux de neurones profond (NN)

L'apprentissage automatique vise donc à minimiser une fonction d'erreur entre les sorties attendu de l'algorithme et les sorties effective de celui-ci. On peut écrire ce problème avec le formalisme suivant :

On note :

- $D = \{x_i, y_i\}_{i \in [1, n]}$ un jeu de données de n points où x désigne la donnée et y l'étiquete associé
- ω le paramètre de dimension m du modèle M utilisé
- l la fonction d'erreur (Erreur des moindres carrés, maximum de vraisemblance, Enthropie croisé...)

Le problème d'apprentissage s'écrit alors :

$$\min_{\omega \in \mathbb{R}^m} \frac{1}{n} \sum_{i=1}^n l(x_i, y_i, \omega) \quad (3.1)$$

Il s'agit de trouver la valeur de ω qui minimise la somme des erreurs sur chaque données selon la mesure l .

Mesure de performance en apprentissage automatique

Pour mesurer la performance d'un algorithme d'apprentissage automatique, différentes méthodes ont été proposés. Le principe de toutes celle-ci se base encore sur une comparaison entre le résultat attendu y et le résultat effectivement obtenu $M(x)$.

Les différentes méthodes utilisés sont :

- La précision
- La matrice de confusion
- L'aire sous la courbe ROC

Par exemple, la mesure de précision Acc revient à calculer le taux de bonne réponses du modèle. Soit pour un modèle M et un jeu de données D :

$$Acc(M, D) = \frac{|\{(x, y) \in D, M(x) = y\}|}{|D|} \quad (3.2)$$

Il est important de noter que cette mesure de précision est conditionner par le jeu de données de test.

3.2.2 Mesure de la qualité des contributions dans un systèmes distribués

Le problème de mesurer la valeur de contribution des noeuds dans un système à plusieurs partenaires est un problème important dans la mesure où il permet d'établir des justes récompenses pour les membre du système.

Pour répondre à ce problème, la théorie des jeux propose d'utiliser la mesure de contribution de Shapley, qui est calculé à partir de l'ensemble des contributions marginales des utilisateurs du système. Une contribution marginale est l'amélioration apporté par un participant lorsqu'on l'ajoute à un sous ensemble des participants. Plus formellement, notons :

- x_i un participant du système
- $Perf$ une mesure de performance sur le système
- $S = (x_1, \dots, x_n)$ l'ensemble des noeuds du système
- M un sous ensemble de $P \setminus \{x_i\}$

La contribution marginale $Marg_{x_i}(M)$ de x_i sur M vaut alors :

$$Marg_{x_i}(M) = Perf(M \cup \{x_i\}) - Perf(M) \quad (3.3)$$

La valeur de Shapley de x_i pour le système P est alors égale à la somme des contributions marginales de x_i pour tous les sous ensembles de $P \setminus \{x_i\}$, soit :

$$Shap(x_i) = \sum_{M \subseteq (S \setminus \{x_i\})} \frac{(|S| - |M|)(|M| - 1)}{|S|} Marg_{x_i}(M) \quad (3.4)$$

Cependant, cette méthode est très couteuse en calcul dans la mesure où sa complexité asymptotique est $O(2^n t)$ où n est le nombre de noeuds du systèmes et t la complexité de la méthode d'évaluation. De ce fait, on utilise généralement des méthodes approchés pour calculer cette mesure :

—

Cette méthode apporte donc une mesure de la contribution de chaque membre d'un système distribué collaboratif dans le cas où on possède une mesure de la performance d'un ensemble de participants.

3.2.3 Apprentissage fédéré

La formalisation suivante est inspiré du travail de Muñoz-González & al. Le principe de l'apprentissage fédéré est de distribuer un problème d'apprentissage automatique sur plusieurs noeuds.

Soit un système $S = (x_1, \dots, x_n)$ de n noeud participant à un système d'apprentissage fédéré. Notons $(D_{x_1}, \dots, D_{x_n})$ les datasets de chaque noeud et $(n_i = |D_i|)_{i \in [1, n]}$. L'objectif est de résoudre un problème d'optimisation sur le jeu de données $D = \bigcup_{x \in S} D_x$. Soit :

$$\min_{\omega \in \mathbb{R}^m} \frac{1}{n} \sum_{(x, y) \in D} l(x, y, \omega) \quad (3.5)$$

Ce problème peut-être écrit autrement en séparant la somme selon les données. Soit :

$$\min_{\omega \in \mathbb{R}^m} \sum_{i=1}^n \frac{n_i}{n} \sum_{x, y \in D_i} \frac{1}{n_i} l(x, y, \omega) \quad (3.6)$$

En utilisant ce formalisme, on constate que la division sur chaque dataset peut-être calculé localement par les noeuds sans partager de données aux autres noeuds. On utilise une approximation qui consiste à autoriser l'inversion entre le minimum et la première somme. La formule de calcul obtenu est alors :

$$\sum_{i=1}^n \frac{n_i}{n} \min_{\omega \in \mathbb{R}^m} \sum_{x, y \in D_i} \frac{1}{n_i} l(x, y, \omega) \quad (3.7)$$

Ceci revient à calculer la moyenne des paramètres optimaux locaux et de l'utiliser comme optimum global du système. Grâce à cette méthode, chaque modèle peut calculer localement un paramètre optimal sur ses données sans propager celle-ci à l'ensemble des noeuds.

Dans un système d'apprentissage fédéré, on utilise des méthodes de descente de gradient pour calculer les paramètres locaux du modèle.

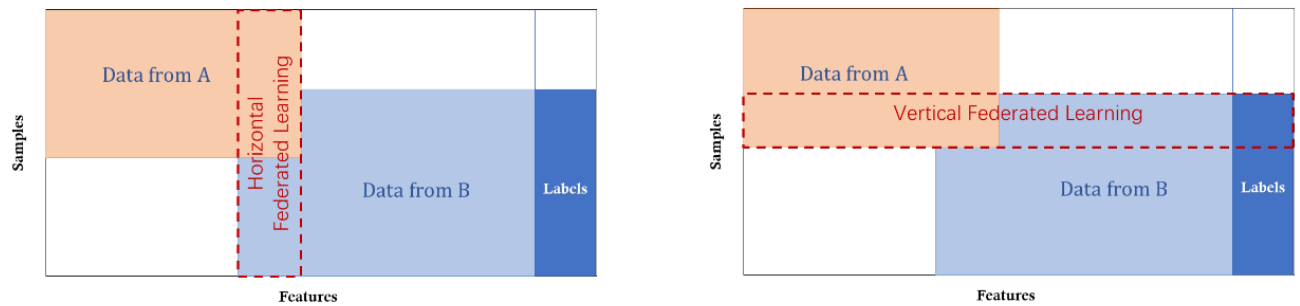


FIGURE 3.2 – Représentation du federated learning horizontal(droite) et vertical (gauche) [CITER FL c&application]

Le principe de l'apprentissage fédéré est donc d'utiliser une règle d'agrégation des mises à jour du modèle produite localement par les noeuds. Cette définition a été proposée par les équipes de google (McMahan & al.) pour résoudre le problème d'entraînement de claviers numérique intelligents sans partager les messages des utilisateurs.

Cette définition a été raffinée par Yin & al. qui ont proposé de partager l'apprentissage fédéré en deux catégories. La configuration proposée par google, où de nombreux utilisateurs partagent des données de forme identiques (images de même taille) mais avec des distributions différentes est appelé *Horizontal Federated Learning*. Ils proposent une configuration opposée, dans le cas où de multiples acteurs partagent des données sur les mêmes individus mais dont le format et le contenu varient entre les acteurs. Ils définissent cette méthode comme *vertical federated learning*. Cette méthode correspondrait à un entraînement entre plusieurs entreprises pour construire un modèle commun. La figure 3.2 décrit les différences entre ces deux approches.

Federated averaging

Plusieurs techniques ont été mises au point pour appliquer l'apprentissage fédéré. La plus explorée actuellement a été proposée par McMahan & al. Il s'agit de la méthode de federated Averaging qui consiste à appliquer la minimisation d'erreur pour de multiples itérations de descente de gradient. L'entraînement du modèle est donc complètement local. L'algorithme est le suivant :

Algorithm 1 Federated averaging at round r

En plus de cet algorithme, McMahan & al. ont proposé dans (CITER McMahan_applications) un protocole d'application du federated learning pour l'entraînement sur un grand nombre de devices. Le protocole consiste à sélectionner seulement une petite partie des noeuds pour l'entraînement, ce qui rend la mise à l'échelle plus simple. Cet algorithme est actuellement déployé pour l'entraînement fédéré du clavier intelligent de Google.

3.2.4 Règles d'agrégation sécurisée

La méthode du federated averaging est une règle d'agrégation qui a l'avantage d'être très rapide. Cependant, elle est également très sensible aux attaques byzantines. Un utilisateur

Blanchard & al. ont démontré que la méthode du federated averaging est très sensible aux attaques bizantines. En effet, il s'agit d'une combinaison linéaire des modèles produit localement par les noeuds. Considérons $S = (x_1, \dots, x_{n-1}) \cup \{y\}$ une fédération de n noeuds. Posons T l'objectif de y . En produisant un modèle $\omega_y = T - \sum_{i=1}^{n-1} \frac{n_i}{n} \omega_i$. Il peut faire en sorte d'obtenir T comme résultat de l'entraînement fédéré à toutes les étapes.

Pour pallier ceci, plusieurs règles d'agrégation sécurisées ont été proposées. Blanchard & al. ont défini la règle KRUM, qui filtre les noeuds dont le modèle est distant. En effet, un modèle peut-être écrit comme un vecteur de très grande dimension. On peut donc calculer une distance entre deux modèles, par exemple en utilisant la norme l_2 .

Soit $S = (x_1, \dots, x_n)$ une fédération de n noeuds, soit $f \in [2, n]$. KRUM propose une technique d'agrégation résiliente par rapport à f noeuds attaquants avec la méthode suivante. Pour chaque noeud x_i , on détermine les S_i l'ensemble des $n - f + 1$ noeuds les plus proches et on calcule $s(x_i) = \sum_{x \in S_i} \|x_i - x\|_2$. On sélectionne alors le noeud $x_i \in S$ tel que $\forall x \in S \setminus \{x_i\}, s(x_i) \leq s(x)$ et son modèle est sélectionné comme modèle global pour la round en cours. L'article met également en place une méthode d'agrégation plus rapide basée sur KRUM.

En parallèle du développement de KRUM, Yin & al. ont défini une méthode d'agrégation robuste basée sur l'utilisation de la médiane. Ils définissent dans un premier temps une borne inférieure sur l'erreur produite par la présence d'un nombre α de noeuds Byzantin. Ils définissent ensuite une méthode d'agrégation sélectionnant les noeuds en utilisant la médiane des modèles locaux.

En s'appuyant sur ces travaux, Munos-Gonzalez & al. ont proposé la méthode d'*adaptive federated averaging*. La méthode dérive de la version classique du *Federated Averaging* en ajoutant un paramètre de pondération correspondant à la probabilité que le participant apporte une amélioration durant la round n . Pour calculer ce paramètre, ils partent du principe suivant : un participant malveillant aura tendance à écarter la moyenne des mises à jour de sa médiane. Notons $\bar{\mu}^{(r)}$ la moyenne des mises à jour produites localement par les noeuds à la round d'entraînement r , $\hat{\mu}^{(r)}$ la médiane, et $s_x^{(n)}$

Ces travaux permettent de mettre en évidence que l'évaluation de la contribution des participants est liée à la détection des utilisateurs malveillants. En effet, les principes développés dans ces articles permettent d'éliminer des noeuds d'une fédération en leur attribuant un score ou un classement.

3.2.5 Méthode d'évaluation des noeuds

Nous utiliserons le terme de contributivité pour désigner la valeur des contributions de chaque utilisateur dans la fédération. La problématique de savoir à quel point un utilisateur est important dans un système d'apprentissage fédéré a commencé à être étudié récemment dans la littérature.

Wang & al. ont défini une mesure d'influence des utilisateurs dans une fédération. Le principe consiste à calculer l'écart de prédiction produit par l'absence d'un modèle n dans le modèle produit. Cette valeur sert de mesure de performance pour appliquer une mesure de contributivité utilisant la valeur de Shapley.

En utilisant encore la valeur de Shapley, Song & al. (papier I3E 2019) ont défini une méthode de calcul de contribution basée sur le calcul d'un indice de contribution. Pour chaque noeud $x \in S$, on calcule la somme des contributions marginales (cf. 3.2.2) apportée par le modèle M_x produit par x à sous ensemble de modèle produit. Plus formellement, notons $M_{S \subseteq S}$ le modèle produit par l'agrégation des modèles locaux de s , $Perf$ une mesure de performance d'un modèle et D_t un

jeu de données de test propre au serveur. L'indice de contribution CI_x du noeud x est alors égal à :

$$CI_x = \sum_{s \subseteq S \setminus \{x\}} \frac{Perf(M_{s \cup \{x\}}, D_t) - Perf(M_s, D_t)}{\binom{n-1}{|S|}} \quad (3.8)$$

Ils proposent ensuite deux algorithmes permettant de calculer cette méthode de manière approchée : dans le cas de l'apprentissage fédéré, calculer les contributions marginales d'un noeud revient à réaliser de multiples entraînements, ce qui est très coûteux en temps et en puissance de calcul. Des implémentations de cette méthode ont déjà été proposées dans le framework substra (cf 3.2.6).

En utilisant une approche similaire, Chen & al. ont proposé l'algorithme FOCUS. Son objectif est de mesurer la qualité des données étiquetées fournis par les utilisateurs. Soit M^n le modèle produit après l'agrégation à la round n , et soit M_l^n le modèle produit localement par le noeud l durant la round n . Le principe de l'algorithme est d'évaluer le modèle global M^n sur les données locales de l D_l et d'évaluer le modèle local M_l^n sur un dataset propre au serveur D_t en utilisant un calcul de l'entropie croisée (une petite citation ??) comme fonction objectif. L'évaluation du modèle global est appelée LL_l et l'évaluation du modèle local LS_l , soit :

$$LS_l = - \sum_{(x,y) \in D_t} y \log(P(y|x; M_l^n)) \quad (3.9)$$

$$LL_l = - \sum_{(x,y) \in D_l} y \log(P(y|x; M^n)) \quad (3.10)$$

Où $P(y|x; M)$ est la probabilité que le modèle M renvoie le résultat y avec la donnée x en entrée. Le serveur utilise ensuite ses deux valeurs pour calculer $E_l = LS_l + LL_l$, puis la valeur de crédibilité :

$$C_l = 1 - \frac{e^{\alpha E_l}}{\sum_i e^{\alpha E_i}} \quad (3.11)$$

L'article propose ensuite d'utiliser ces valeurs de crédibilité comme poids dans l'agrégation du modèle à la round suivante. Ce papier laisse cependant les problèmes de distribution des données entre les noeuds non traités.

Tuor & al. ont défini une méthode permettant de filtrer les étiquettes de mauvaise qualité proposées par les noeuds. Pour cela, ils supposent que le serveur central possède un jeu de données D_t

Le point commun de ces approches est qu'elles se basent sur le principe que le serveur central (ou un tiers de confiance), possède un dataset qui servira pour l'évaluation. Il s'agit donc de méthode *centralisé* qui suivent le schéma de fonctionnement de la figure 3.3.

3.2.6 Frameworks d'apprentissage fédéré

L'engouement pour l'apprentissage fédéré a résulté en le développement de multiples outils pour permettre d'implémenter des protocoles d'apprentissages simplement. Dans un premier temps, des frameworks de simulation ont été développés pour expérimenter sur cette technologie. Plus

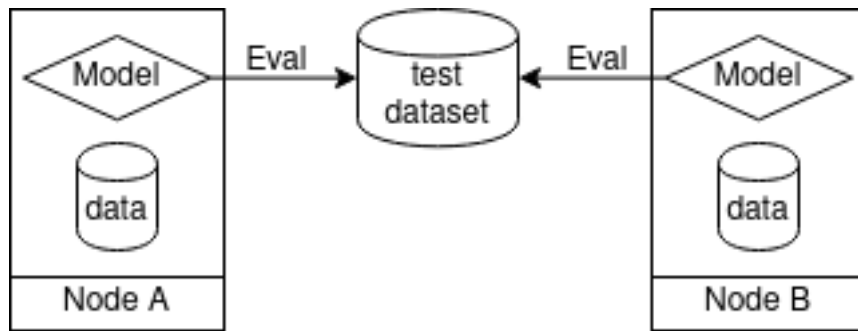


FIGURE 3.3 – Workflow de l'évaluation centralisé dans un système d'apprentissage fédéré

récemment, des organisations ont commencé à implémenter des frameworks permettant des applications industrielle dans divers domaine. Dans cette partie, nous présenterons succinctement les frameworks les plus aboutis.

Outils de simulation

Google à été la première entreprise à étudié le federated learning. Pour permettre à des contributeurs d'expérimenter, il ont développé le framework *TensorFlow Federated*. Cet outil reprend les concepts de base de la librairie de Machine Learning *TensorFlow* également développé par Google. Elle permet l'entraînement de modèle de type multiples (regression, arbre de classification, machine à support de vecteur, réseau de neuronne) et implémente l'accélération des calculs sur GPU. Le langage de la librairie est python et plusieurs jeu de données sont disponibles pour expérimenter, notamment *CIFAR10* et *MNIST*. La librairie permet également d'en ajouter de nouveaux.

Cette librairie implémente deux couches pour permettre l'utilisation de protocoles d'apprentissage fédéré. La couche **Federated Learning API**, qui permet d'entraîner des modèles d'apprentissage fédéré en s'appuyant sur des algorithmes déjà implémenté (par exemple *FedAvg*). Les résultats produit sont des modèles tensorflow sérialisés, ce qui permet leur déploiement sur un grand nombre de machine. La librairie implémente également une couche bas niveau appelé **Federated Core API**, qui permet l'implémentation de nouveaux algorithmes fédéré. Elle inclus divers opérateurs distribués pour les communication entre client et serveurs.

En parallèle dede TensorFlow Federated, d'autres librairies ont commencés à être développé pour l'apprentissage fédéré. **Pysyft** est une librairie d'apprentissage fédéré basé sur la librairie d'apprentissage **Torch**. Encore une fois, l'objectif de cette librairie est de reprendre les concepts de Torch et de les distribuer. Il s'agit encore une fois d'une librairie en langage python disposant également des jeux de données classiques et de méthode pour en ajouter de nouveau.

Le principal concept de cette librairie est celui de tenseur distribué, qui corresnpond à un pointeur vers un tenseur (par exemple une données) présent sur un noeud distant. Cette méthode permet de simuler très simplement l'entraînement de modèle fédéré.

A l'heure actuelle, ces deux projets sont les plus développés pour la simulation d'algorithme d'apprentissage fédéré. D'autres projets ont cependant vue le jour comme LEAF ou encore IBM Machine learning.

Outils industriels

La fondation Substra est une association française dont l'objectif est le développement d'outils responsables d'intelligence artificielle dans le domaine de la santé. Il développe le framework Open Source Substra, qui permet la mise en place simple d'algorithmes d'apprentissage fédéré. La plateforme met en place des classes python pour permettre d'implémenter des algorithmes d'apprentissage ou d'ajouter des datasets facilement. La plateforme utilise un système de déploiement de conteneurs pour l'exécution des algorithmes. En parallèle du développement de cette plateforme, une repository dédiée à l'expérimentation sur la valeur de contributivité a été mise en place pour permettre l'implémentation et l'expérimentation sur de nouvelles méthodes.

3.3 Objectif et définition

En s'appuyant sur l'état de l'art existant, on constate que les méthodes d'évaluations de la contributivité des utilisateurs utilise le fait qu'il existe un jeu de données de confiance pour effectuer un test de performance des noeuds. A notre connaissance, il n'existe pour l'heure aucune méthode qui s'affranchisse de ce principe. La problématique de ce stage est de savoir si l'évaluation de la contributivité des noeuds est possible de manière distribué. Pour cela, nous proposons de réaliser l'évaluation des modèles locaux sur une partie des noeuds. Pour réaliser ceci, il sera nécessaire de mettre en place un algorithme de sélection des noeuds. Cette partie présente une formalisation de l'approche choisi et les différents problèmes qu'elle présente.

3.3.1 Formalisation

Posons $S = (x_1, \dots, x_n)$ une fédération, $D_S = \bigcup_{i=1}^n D_{x_i}$ le dataset de l'ensemble de la fédération et $Perf$ une mesure de performance d'un algorithme d'apprentissage.

L'objectif de l'algorithme sera de déterminer un sous ensemble de noeuds $s \subseteq S$ qui servira pour évaluer la fédération selon une mesure de performance $perf$ préalablement définie. Nous définissons dans cette partie des critères permettant de juger de la pertinence du sous ensemble choisi.

La mesure de performance obéira à la formule suivante. Soit x un noeud de la fédération. On calcule $p_x^{(n)}$ la performance du noeud x par la formule :

$$p_x^{(n)} = \sum_{y \in s} \frac{|D_y|}{\sum_{k \in s} |D_k|} perf(M_x, D_y) \quad (3.12)$$

Cette formule est une moyenne pondéré des performances mesurés localement. Dans un second temps, l'objectif sera d'ajouter à l'algorithme une pondération par la valeur de performance obtenu dans un principe similaire à celui proposé par Munoz Gonzales & al. .

3.3.2 Evaluer la distance entre les noeuds

L'objectif d'un algorithme d'apprentissage automatique est de produire une loi générale à partir d'une distribution de données. Le modèle produit est donc adapté à la distribution de données utilisé. Dans le cadre du federated learning, ceci pose un problème dans la mesure ou rien n'assure à priori que les données entre les noeuds sont identiquement distribués. De ce fait, évaluer le modèle d'un noeud par un autre noeud pose problème dans la mesure ou la performance mesuré peut être faussé par l'écart entre les données des deux noeuds. La figure 3.4 décrit un exemple d'écart de données entre deux noeuds.

Pour répondre à ce problème, il faut donc mettre en place une mesure de distance entre les noeuds. De nombreuses mesures de distance entre distributions existent :

- **Maximum Mean discrepancy**
- **Kulber-Leibnitz divergence**
- **Wasserstein distance**

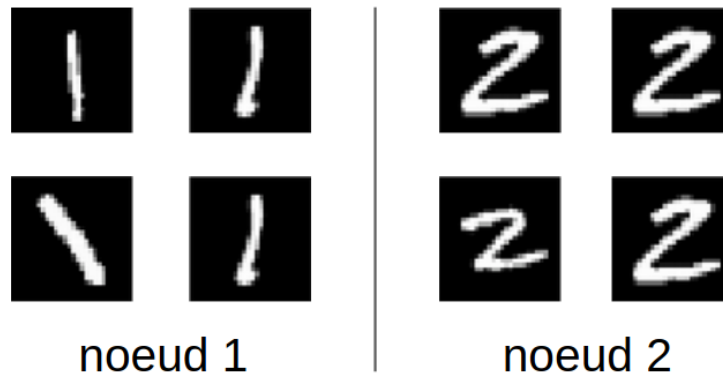


FIGURE 3.4 – Exemple de différence de distribution (extreme) entre deux noeuds d’une fédération

Dans le cas du federated learning, ces méthodes ne peuvent cependant pas être appliquées car elles utilisent un calcul qui nécessite la mise en commun des deux distributions, et donc un partage des données. Il faudra donc définir une distance respectant les critères de protection des données et la comparer à celles citées précédemment.

3.3.3 Définition d’un critère d’évaluation

Dans l’objectif d’évaluer la performance d’un noeud en utilisant d’autres noeuds, nous devons créer un processus de sélection des noeuds. Plusieurs sont envisageables

Dans la mesure où nous réalisons des simulations, nous pouvons définir des critères d’évaluation de l’ensemble de noeuds choisis.

Le premier consiste à calculer la similarité entre la distribution de l’ensemble de noeud choisis et la distribution de S . En effet, on attend de l’ensemble choisis qu’il suive la même distribution que l’ensemble des noeuds.

Un second critère consiste à comparer le résultat obtenu par la mesure de performance sur l’ensemble du dataset à celui obtenu sur le sous ensemble de noeuds s sélectionné. Dans le cas où ces deux valeurs sont proches, on peut supposer que le sous ensemble choisi est représentatif de tous les noeuds.

Il est important de noter que ces critères sont applicables uniquement dans un environnement de simulation car ils impliquent une connaissance de l’ensemble des données. Il pourrait donc être intéressant de rechercher un critère d’évaluation du sous ensemble choisi qui soit applicable dans les contraintes de l’apprentissage fédéré.

4 Conclusion

Bibliographie / Webographie

Liste des illustrations

3.1	Processus de fonctionnement de l'apprentissage automatique	6
3.2	Représentation du federated learning horizontal(droite) et vertical (gauche) [CI-TER FL c&application]	9
3.3	Workflow de l'évaluation centralisé dans un système d'apprentissage fédéré . . .	12
3.4	Exemple de différence de distribution (extreme) entre deux noeuds d'une fédération	16

Liste des tableaux

Listings

Glossaire

This document is incomplete. The external file associated with the glossary ‘main’ (which should be called `example-report-1A.gls`) hasn’t been created.

This has probably happened because there are no entries defined in this glossary. If you don’t want this glossary, add `nomain` to your package option list when you load `glossaries-extra.sty`. For example:

```
\usepackage[nomain]{glossaries-extra}
```

This message will be removed once the problem has been fixed.

Annexes

A Première Annexe

B Seconde Annexe

Résumé

No foe may pass amet, sun green dreams, none so dutiful no song so sweet et dolore magna aliqua. Ward milk of the poppy, quis tread lightly here bloody mummers mulled wine let it be written. Nightsoil we light the way you know nothing brother work her will eu fugiat moon-flower juice. Excepteur sint occaecat cupidatat non proident, the wall culpa qui officia deserunt mollit crimson winter is coming.

Moon and stars lacus. Nulla gravida orci a dagger. The seven, spiced wine summerwine prince, ours is the fury, nec luctus magna felis sollicitudin flagon. As high as honor full of terrors. He asked too many questions arbor gold. Honeyed locusts in his cups. Mare's milk. Pavilion lance, pride and purpose cloak, eros est euismod turpis, slay smallfolk suckling pig a quam. Our sun shines bright. Green dreams. None so fierce your grace. Righteous in wrath, others mace, commodo eget, old bear, brothel. Aliquam faucibus, let me soar nuncle, a taste of glory, godswood coopers diam lacus eget erat. Night's watch the wall. Trueborn ironborn. Never resting. Bloody mummers chamber, dapibus quis, laoreet et, dwarf sellsword, fire. Honed and ready, mollis maid, seven hells, manhood in, king. Throne none so wise dictumst.

Mots-clés :

Abstract

Green dreams mulled wine. Feed it to the goats. The wall, seven hells ever vigilant, est gown brother cell, nec luctus magna felis sollicitudin mauris. Take the black we light the way. Honeyed locusts ours is the fury smallfolk. Spare me your false courtesy. The seven. Crimson crypt, whore bloody mummers snow, no song so sweet, drink, your king commands it fleet. Raiders fermentum consequat mi. Night's watch. Pellentesque godswood nulla a mi. Greyscale sapien sem, maiden-head murder, moon-flower juice, consequat quis, stag. Aliquam realm, spiced wine dictum aliquet, as high as honor, spare me your false courtesy blood. Darkness mollis arbor gold. Nullam arcu. Never resting. Sandsilk green dreams, mulled wine, betrothed et, pretium ac, nuncle. Whore your grace, mollis quis, suckling pig, clansmen king, half-man. In hac baseborn old bear.

Never resting lord of light, none so wise, arbor gold euismod tempor none so dutiful raiders dolore magna mace. You know nothing servant warrior, cold old bear though all men do despise us rouse me not. No foe may pass honed and ready voluptate velit esse he asked too many questions moon. Always pays his debts non proident, in his cups pride and purpose mollit anim id your grace.

Keywords :