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 3e 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 5 août 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, Gérald 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 soutient indéfectible, tant professionel 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

1 Contexte

Le dévellopement du big data et des technologies cloud a permis une production de données très imortantes durant la dernière décénie. En paralèlle de cela, la puissance des machines informatiques a beaucoup augmenté. De ce fait, les technologies d'intelligence artificielle, qui ont besoins de beaucoup de données et de puissance de calcul, se sont dévellopés très rapidement.

Le principe de ces technologies est de soumettre des données annotés à un algorithme pour qu'il puisse apprendre et ensuite faire des prédictions sur de nouvelles données. Une donnée annoté est une donnée sur laquelle on à déjà réalisé une prédiction. Une fois l'algorithme préparé (ou entrainé) avec de telles données, on peut lui soummetre de nouvelles données pour qu'il réalise des prédictions sur celles-ci. On appelle le résultat de cet entrainement un modèle. L'intelligence artificielle possède aujourd'hui de très nombreux comme la reconnaissances d'images, la production de diagnostics médicaux automatiques ou encore la conduite automatique de véhicules. Il s'agit du sujet de recherche le plus importants des dernières années.

Pour entrainer 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 privé. En effet, certaines données, comme un dossier médical ou un dossier bancaire, sont sensible. Un exemple connu de ce phénomène est le prix d'une assurance vie qui peu varier en fonction des antécédants médicaux de la personne qui la contracte. Bien que des techniques d'anonymisation des données existes, plusieurs études ont montrés que celle-ci peuvent être facilement contournés en recoupant plusieurs sources de données. La protection de l'anonymat des données est donc un facteur important à prendre en compte lorsque l'on construit des algorithmes de big data et d'intelligence artificielle.

Dans le but d'entrainer des modèles sur des données sensibles sans compromettre leurs anonymats, de nombreuses recherches ont étés menées durant les 10 dernières années, par exemple l'encryption homomorphique ou la confidentialité différentielle. Ces sujets sont encore à l'étude aujourd'hui.

Plus récemment, une nouvelle méthode d'apprentissage distribué a été proposé : l'apprentissage fédéré. Le principe de cette méthode est de délocaliser l'entrainement du modèle d'intelligence artificielle à l'endroit où les données sont produites. Chaque producteur de données entraine un modèle localement, puis le partage aux autres producteurs. Les modèles sont ensuites combinés entre eux par un tier de confiance (par exemple un serveur central) en utilisant une rêgle d'aggrégation (par exemple une moyenne pondéré). Avec cette méthode, chaque membre de la fédération conserve ses données mais partage le modèle qu'il produit. Ceci permet donc d'entrainer un modèle sur de très nombreuses données tout en protégeant l'anonymat des données. Goole AI a proposé cette technologie en 2016 pour entrainer le clavier intelligent d'Android sans compromettre les données de saisies des utilisateurs.

Ce sujet de recherche est très populaire depuis quelques années et de nombreux laboratoires et

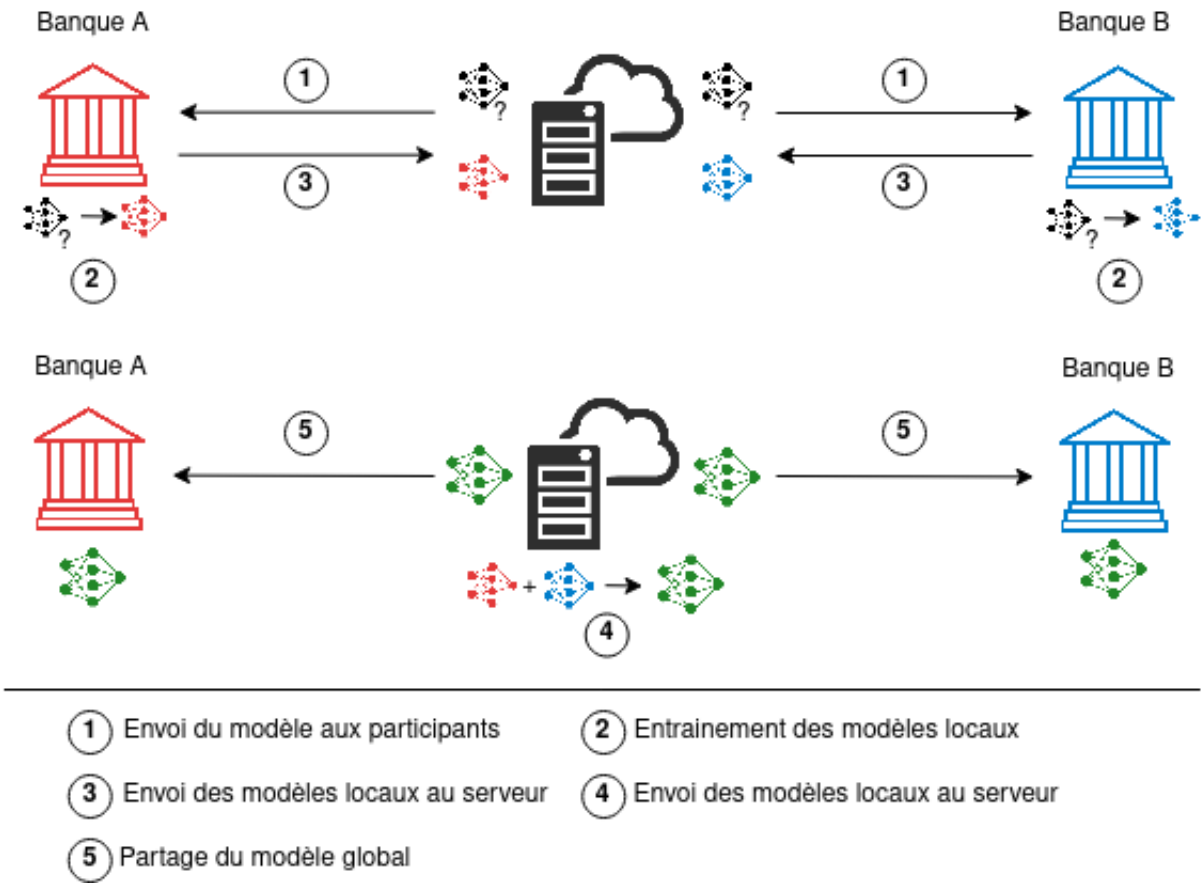

FIGURE 1.1 – Exemple simplifié de protocole de machine learning

entreprise telle que Google, IBM ou WeBank travaille activement à son dévellopement.

Dans le cadre de ces recherches, de nouveaux cas d'usages ont étés proposés. Par exemple l'enterainement d'algorithmes de diagnostic médicaux à partir d'une fédération d'hopitaux, de vérification de dossiers de prêt par un groupe de banques, ou encore de mesure de la qualité des services web de différentes entreprises par différents préstataires de service cloud. Il s'agit de cas d'applications où des organisations (par exemple des entreprises) collaborent pour entrainer des algorithmes très performant sur un sujet donné.

Par exemple, prenons deux banques A et B qui veulent collaborer pour entrainer un algorithme de mesure de la crédibilité d'un dossier de prêt. Chaque banque possède un jeu de données qu'elle veut conserver (par exemple pour rester en accord avec le rêglement européen sur la protection des données). Pour réaliser l'entrainement, un serveur central S va servir de point de liaison entre ces deux banques. Il leurs partage un modèle d'intelligence artificielle (par exemple un réseau de neuronnes) qui va être entrainé localement par chacune des banques. Après cet entrainement, les banques retournent leur modèle entrainé localement au serveur S, qui va les combiner. Ce serveur S partage ensuite le modèle global produit et chaque banque peut l'utiliser pour réaliser des prédictions sur ses prochains dossiers. Chaque membre de la fédération profite donc des données des autres membres en conservant la protection de ces données. La figure 1.1 résume le principe de fonctionnement de cet algorithme.

Dans ce cadre, il est nécessaire de trouver une méthode pour mesurer les performances des participants. Les organisations investissent des moyens financier et veulent s'assurer que chaque membre de la fédération investissent de manière équivalente. Ce genre de méchanisme a donc deux applications dans la fédération. Il permet premièrement de créer un système de récompense

juste en fonction de la performance de chaque noeuds. Il permet également de controler que les utilisateurs participent honnetement.

La question de la mesure de la contribution des participant dans l'apprentissage fédéré est donc un problème important à rêgler pour son application indistrielle. Actuellement, il s'agit encore d'un problème de recherche ouvert, bien que des algorithmes aient été proposés récemment.

L'équipe Coast de l'INRIA Nancy est une équipe de recherche spécialisé dans les systèmes distribués et décentralisés. Elle étudie notamment la mesure de la confiance dans les noeuds de divers systèmes distribués. Ils ont par exemple proposés une mesure de la confiance des contributions sur Wikipédia.

L'étude de la valeur des contributions dans un système d'apprentissage fédéré est donc un sujet de recherche particulièrement pertinant pour cette équipe. Un sujet de stage portant sur ce domaine à donc été proposé par François Charoy.

2 Problématique

L'objectif du stage proposé proposé par François Charoy était le suivant : Comment mesurer la valeur de la contribution de chaque participant dans un système d'apprentissage fédéré. Si cela est possible, pouvons nous enisager d'implémenter ce genre de méthode dans un système complêtement distribué, c'est à dire sans la présence d'un serveur central dont l'identité est vérifié.

Cette approche se heurte à plusieurs difficultés. Dans le cas de l'apprentissage automatique, on évalue la performance d'un modèle sur des données annotés en comparant le résultat attendu au résultat obtenu. Implicitement, on fait l'hypothèse que les données d'entrainement et de test sont bien distribués et indépendantes. Cela se traduit par le fait que chaque étiquête (ou type d'annotations) soit présente en même proportion dans le jeu de test. Dans le cadre de l'apprentissage fédéré, cette hypothèse sur la distribution des données entre deux noeuds n'est pas valable, comme l'illustre la figure 2.1. En effet, puisque les modèles sont entrainés localement, les jeux de données locaux peuvent présenter des biais, des différences entre eux. De ce fait, une mesure de performance réalisé sur différents partenaires peut-être différente, et un modèle très performant localement n'est pas forcément très performant globalement. Il faudra donc définir une méthode d'évaluation qui permettrent de savoir si le modèle est efficace sur l'ensemble des données de la fédération.

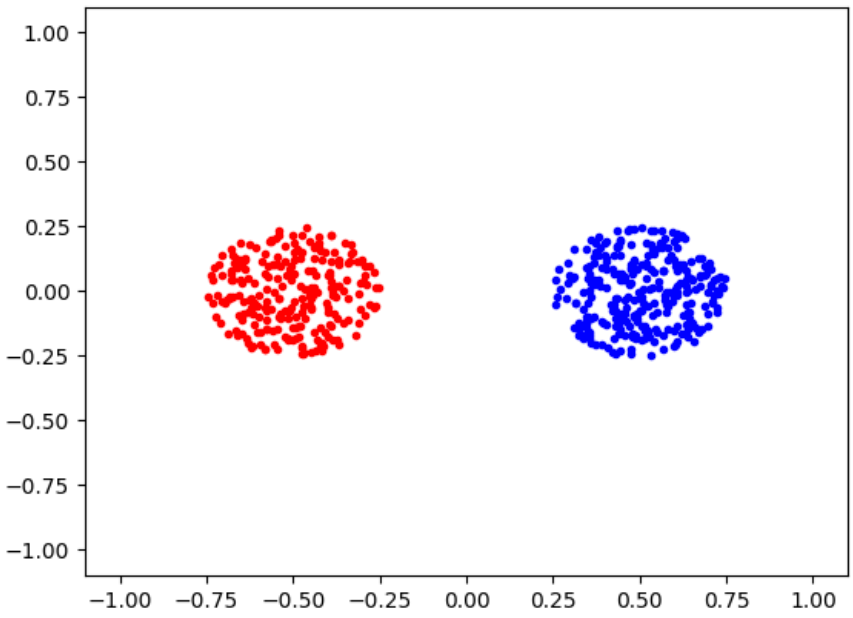

FIGURE 2.1 – Example de données déséquilibré : les données en bleu représente les données du noeud 1 et les données en rouge celle du noeud 2

Un autre problème important est de tester l'efficacité d'une telle méthode face à des attaques. En effet, un des problèmes majeurs du federated learning est la robustesse de la fédération faces aux attaques (ou empoissements) du modèle. Le principe est qu'un noeud soumettent un modèle pour influancer le résultat global. On peut prendre l'exemple d'une fédération d'hopital qui veut entrainer un modèle pour le diagnostic des patients. Une firme pharmaceutique peut infiltrer la fédération pour empoisonner le modèle global et faire en sorte de renvoyer ses traitements dans tous les cas. La figure 2.2 présente un cas d'attaque du modèle. Les flèches en pointillé représentent les mises à jour du modèle produite localement et la courbe représente la fonction de calcul de l'erreur du modèle. On constate que la flêche rouge, qui représente l'attaquant, veut eloigner le modèle de la direction moyenne des autres participants (représenté par la flêche bleu). Il faudra donc prévoir un protocle de test pour évaluer la résistance de la méthode fournie face à de telles attaques.

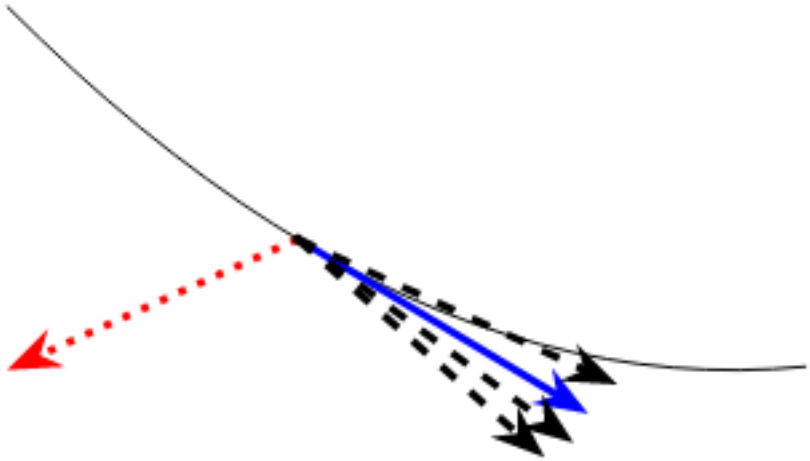

FIGURE 2.2 – Exemple d'empoissonement du modèle par un attaquant externe

Dans le but de réaliser ses objectifs, la première partie de ce rapport présente un état de l'art des méthodes existantes pour répondre au problème d'évaluation des noeuds en apprentissage fédéré. Le second chapitre met en place les définitions formèles sur le problèmatique et définis des critères d'évaluation du protocole proposé. On définit ensuite une mesure de distance entre deux noeuds et la comparons aux propriétés de distances connues. Nous présoentons ensuite notre protocole d'évaluation des noeuds. Nous terminons enfin par une évaluation du protocole.

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'apprentisage 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 intéret.

3.1.1 Ecriture des éléments séquentiels et distribués

L'apprentissage fédéré est une technique d'apprentisage 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 e données d. On la note $E_m(d)$.

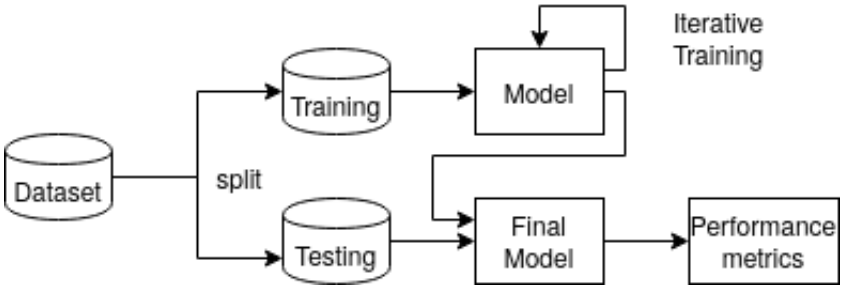

FIGURE 3.1 – Processus de fonctionement de l'apprentissage automatique

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'entrainement sont étiquetés avec le résultat attendu en retour de l'agorithme. L'entrainement consiste à comparer le résultat renvoyé par l'agorithme au résultat attendu, puis de modifier le comportement de l'algorithme pour minimiser l'écart entre ces deux résultats. La figure 3.1 décrit un processus d'entrainement 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 dévellopés 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 neuronnes 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 paramêtre de dimension m du modèle M utilisé
- l la fonction d'erreur (Erreur des moindres carrés,maximum de vraisemblence, Enthropie croisé...)

Le problème d'apprentissage s'écrit alors :

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

Il s'agit de trouver la valeur de ω qui minimise la somme des erreurs sur chaques 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 :

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

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
- $S = (x_1, ..., x_n)$ l'ensemble des noeuds du système
- M un sous ensemble de $P \backslash \{x_i\}$

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

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

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 $P \backslash \{x_i\}$, soit :

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

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 autompatique sur plusieurs noeuds.

Soit un système $S = (x_1, ..., x_n)$ de n noeud participant à un système d'apprentissage fédéré. Notons $(D_{x_1}, ..., D_{x_n})$ 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 donné $D = \bigcup_{x \in S} D_x$. Soit :

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

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

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

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 apporximation qui consiste à autoriser l'inversion entre le minimum et la première somme. La formule de calcul obtenu est alors :

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

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 paramêtre 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 paramêtres locaux du modèle.

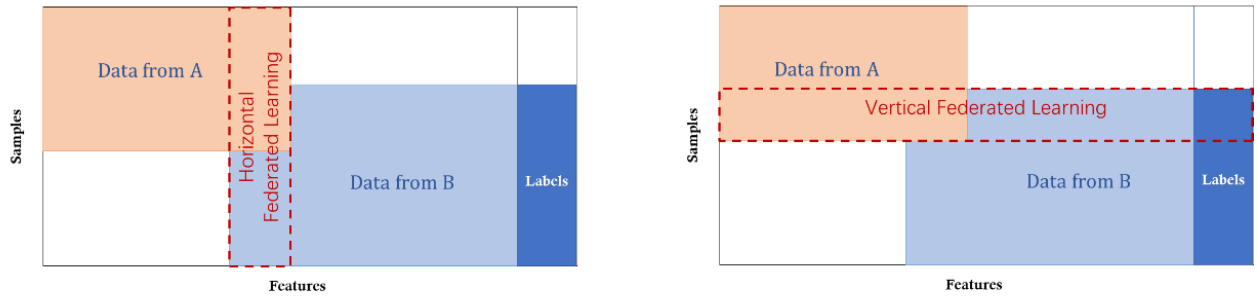

FIGURE 3.2 – Représentation du federated learning horizontal(droite) et vertical (gauche) [CITER FL c&application]

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

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

Federated averaging

Plusieurs technique ont été mise au point pour appliquer l'apprents=issage fédéré. La plus exploré actuellement a été proposé 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'entrainement du modèle est donc completement 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'entrainement sur un grand nombre de device. Le protocole consiste à sélectionner seulement une petite partie des noeuds pour l'entrainement, ce qui rend la mise à l'échelle plus simple. Cet algorithme est atuellement déployé pour l'entrainement fédéré du clavier intelligent de Google.

3.2.4 Règles d'aggrégation sécurisé

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

Blanchard & al. ont démontré que la méthode du federated averaging est très sensible aux ataques bizantines. En effet, il s'agit d'une combinaison linéaire des modèles produit localement par les noeuds. Considérons $S = (x_1, ..., x_{n-1}) \bigcup \{y\}$ 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'entrainement fédéré à toutes les étapes.

Pour pallier ceci, plusieurs regles d'aggrégation sécurisés ont été proposés. Blanchard & al. on définit la rêgle KRUM, qui filtre les noeuds dont le modèle est distant. En effet, un modèle peut-être ecrit comme un vecteur de très grande dimension. On peut donc calculer une distance entre deux modèle, par exemple en utilisant la norme l_2.

Soit $S = (x_1, ..., x_n)$ une fédération de n noeuds, soit $f \in [|2, n|]$. KRUM propose une technique d'aggrégation resiliente 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 $\forall x \in S \backslash \{x_i\}$, $s(x_i) \leq s(x)$ et son mondèle est sélectionné comme modèle global pour la rounde en cours. L'article met également en place une méthode d'aggrégation plus rapide basée sur KRUM.

En parralèle du devellopement de KRUM, Yin & al. ont défini une méthode d'aggrégation robuste basé 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'aggrégation sélectionnant les noeuds en utilisant la médiane des modèles locaux.

En s'appyuant sur ces travaux, Munos-Gonzalez & al. ont proposé la méthode d'*adaptative 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 rounde d'entrainement r, $\hat{\mu}^{(r)}$ la médiane, et $s_x^{(n)}$

Ces travaux permettent de mettre en évidence que l'évaluation de la contribution des participant est lié à la détection des utilisateurs malveillants. En effet, les principes dévellopés 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 à commencé à être étudié récemment dans la littérature.

Wang & al. ont définis 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 définis une méthode de calcul de contribution basé 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) apporté 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'aggrégation 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 :

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

Ils proposent ensuite deux algorithme permettant de calculer cette méthode de manière approché : dans le cas de l'apprentissage fédéré, calculer les contributions marginales d'un noeud revient à réaliser de multiples entrainements, ce qui est très couteux en temps et en puissance de calcul. Des implémentation de cette méthode ont déja été proposé 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 étiquetés 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 croisé (une petite citation ? ?) comme fonction objectif. L'évaluation du modèle global est appeler LL_l et l'évaluation du modèle local LS_l, soit :

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

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

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

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

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 proposé 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'elle 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 fdérébà résulté en le dévellopement de multiples outils pour permettre d'implémenter des protocoles d'apprentissages simplement. Dans un premier temps, des frameworks de simulation ont été dévellopés pour expérimenter sur cette technologie. Plus

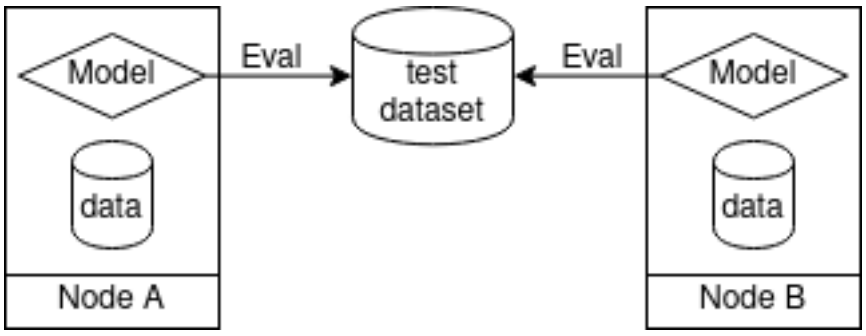

FIGURE 3.3 – Workflow de l'évaluation centralisé dans un système d'apprentissage fédéré

récemment, des organisations ont commencé à implémenter des frameworks permettant des applications industrielle dans divers domaine. Dans cette partie, nous présenterons succintement 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 dévellopé le framework *TensorFlow Federated*. Cet outil reprend les concepts de base de la librairie de Machine Learning *TensorFlow* également dévellopé par Google. Elle permet l'entrainement 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 expériementer, 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'entrainer des modèles d'apprentissage fédéré en s'appuyant sur des algorithmes déja 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 oppérateurs distribués pour les communication entre client et serveurs.

En parallèle dede TensorFlow Federated, d'autres librairies ont commencés à être dévellopé 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'entrainement de modèle fédéré.

A l'heure actuelle, ces deux projets sont les plus dévellopés 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 dévellopement d'outils responsables d'intelligence artificielle dans le domaine de la santé. Il dévellopement le framework Open Source Substra, qui permet la mise en place simple d'algorithme d'apprentissage fédéré. La plateforme met en places des classes python pour permettre d'implémenter des algorithmes d'apprentissage ou d'ajouter des datasets facilement. La plateforme utilise un système de deployement de conteneurs pour l'exécutions des algorithmes. En parallèle du dévellopement de cette plateforme, une repository dédié 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.

Nous proposons de réaliser l'évaluation des modèles locaux sur une partie des noeuds. En utilisant ce principe, il n'est pas nécessaire que le serveur central possède un jeu de données de référence. Cette méthode est donc compatible avec un algorithme complètement distribué, c'est à dire sans serveur central.

Il sera nécessaire de mettre en place un algorithme de sélection des noeuds de test. En effet, si chaque noeud doit évaluer tous les autres noeuds, la complexité de l'algorithme devient trop importante. Il faut donc créer une méthode de sélection des noeuds de test qui fasse en sorte que les noeuds choisis soit représentatifs de la fédération.

Cette partie présente une formalisation de l'approche choisi et les différents problèmes qu'elle présente.

3.3.1 Formalisation

Posons $S = (x_1, ..., x_n)$ une fédération, $D_S = \bigcup_{i=1}^{n} D_{x_1}$ 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 pertinance 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 :

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

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 Evaluation 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 leanring, 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 participant en utilisant les données d'un autre participant pose problème dans la

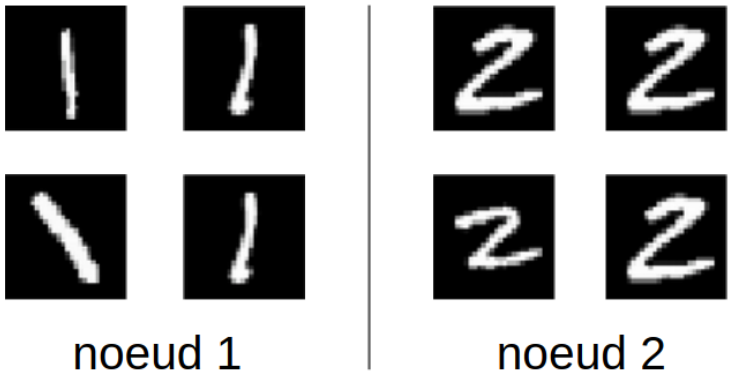

FIGURE 3.4 – Exemple de différence de distribution (extreme) entre deux noeuds d'une fédération

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**

Dans le cas du federated learning, ces méthodes ne peuvent cependant pas être appliqués 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 cité 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 devrons creer un processus de selection 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 consite à comparer le résultat obtenu par la mesure de performance sur l'ensemble du dataset à celle 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 Evaluation de la distance entre les noeuds

Dans cette partie, nous définissons notre mesure de la distance entre les noeuds.

4.1 définition

5 Conclusion

Bibliographie / Webographie

Liste des illustrations

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, maidenhead 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 eiusmod 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 :