Projet M1

Année scolaire 2023/2024

Institut Supérieur de l'Électronique et du Numérique

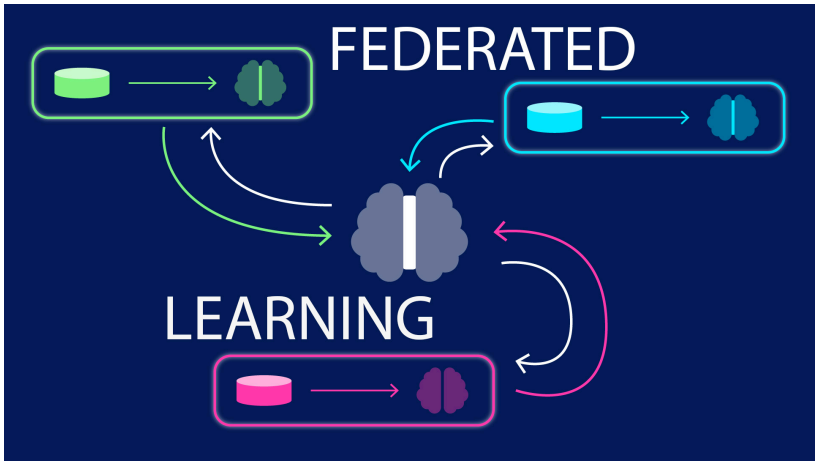
Tél. : +33 (0)2.98.03.84.00

Fax : +33 (0)2.98.03.84.10

20, rue Cuirassé Bretagne

CS 42807 - 29228 BREST Cedex 2 - FRANCE

Extraction sécurisée de connaissances à partir de documents via l'intelligence artificielle



Proposé par Karine Ayoub
Thématique : Informatique et intelligence artificielle

Royer Thomas - Intelligence artificielle
Soydemir Antoine - Génie logiciel

Résumé

Ce rapport expose les résultats obtenus lors de la réalisation du projet de quatrième année mené par Thomas ROYER et Antoine SOYDEMIR.

L'apprentissage fédéré ou "Federated Learning" est un paradigme d'apprentissage. Il apporte un concept nouveau qui présente de nombreux avantages d'un point de vue de la sécurité et la confidentialité des données.

Ce rapport détaillera exhaustivement le processus d'apprentissage du Federated Learning, de sa mise en place à son association avec l'OCR (Optical Character Recognition), une technologie permettant de faire de la reconnaissance de caractères. Le rapport apporte donc une dimension technique avec les outils et technologies utilisées et déployées mais également une dimension théorique pour comprendre et démystifier les principaux rouages et concepts dans le domaine de l'intelligence artificielle.

Pour finir, le rapport présentera des résultats pour quantifier les performances de ce type d'apprentissage.

Remerciements

Nous tenons à exprimer notre profonde gratitude envers Monsieur Ayoub Karine, Associate Professor spécialisé en Computer Vision, pour avoir initié ce projet et pour nous avoir enseigné les fondements de l'intelligence artificielle. Sa guidance et ses conseils nous ont grandement aidé tout au long du développement du projet.

De même, nous souhaitons adresser nos sincères remerciements à Madame Hajar Rehioui, Docteur-Ingénieur en Recherche et Développement en Intelligence Artificielle. Nous sommes reconnaissants pour la richesse de ses connaissances qu'elle a partagées avec générosité, contribuant ainsi à l'enrichissement de notre compréhension du sujet.

SOMMAIRE

Table des figures

I - Introduction

L'intelligence artificielle (IA) se présente aujourd'hui comme l'un des domaines les plus dynamiques et révolutionnaires de la technologie moderne. Cette technologie façonne le monde de demain à grande vitesse.

Fusion entre cognition humaine et science informatique, l'intelligence artificielle a révolutionné la productivité et l'innovation dans de nombreux secteurs : de la finance à la santé en passant par la logistique. Pour comprendre son impact actuel et à venir, il est important de se plonger dans son histoire, marquée par des prouesses et des avancées technologiques.

Pour comprendre l'histoire de l'intelligence artificielle, il faut remonter à plusieurs décennies. Durant les années 1950, les chercheurs Alan Turing et John McCarthy ont exploré la possibilité de créer des machines capables d'apprendre et de penser comme des êtres humains. Ces premières recherches ont donné naissance à des concepts fondamentaux : les algorithmes d'apprentissage automatique et les réseaux neuronaux.

Le projet repose sur l'étude d'une technique d'apprentissage récente et révolutionnaire parmi les techniques existantes : le Federated Learning. Le rapport examinera les choix technologiques effectués et proposera des explications avancées sur le fonctionnement de cette méthode d'apprentissage. Il abordera les problématiques de performances pour mesurer la qualité de cet apprentissage.

Le rapport présente un cas concret d'application de l'apprentissage réalisé à travers l'Optical Character Recognition (OCR) dans le but de reconnaître les chiffres entre zéro et neuf écrits de manière manuscrite. Il présentera les différentes étapes parcourues pour mener ce projet.

II - Cahier des charges

Le principe initial auquel le projet doit aboutir est la montée en compétences sur des technologies d'avenir telles que le Federated Learning et l'Optical Character Recognition (OCR). L'objectif est de réaliser une production qui est capable de simuler l'apprentissage fédéré pour permettre aux clients de donner en entrée des chiffres manuscrits qui seront identifiés, reconnus et convertis numériquement par la solution en conservant la confidentialité des données entre les données du client et la prédiction effectuée. Les données d'entrée seront des fichiers au format d'image, il sera donc nécessaire d'effectuer des transformations pour que l'image soit identifiée par l'intelligence artificielle. Une phase de prétraitement de l'image est à prévoir.

Le projet est réalisé dans le cadre scolaire mais également en partenariat avec le pôle recherche et développement de l'entreprise OpenBee spécialisée en GED (Gestion Electronique des Documents).

Pour mener à bien la mise en place de cette solution, il a été important de fixer des choix technologiques déterminants. Notamment, le choix de l'Integrated Development Environment (IDE) ou du choix de différentes bibliothèques. Le cahier des charges n'a imposé aucune condition à ce sujet. Cependant, les différents choix doivent être justifiés et un comparatif entre les différentes possibilités a été effectué.

Le projet propose un site internet permettant de consulter le rendu de la solution. L'utilisateur sera en mesure de parcourir le site pour obtenir des informations générales au sujet du projet et des collaborateurs ayant participé à sa réalisation. Une démonstration de la solution sera disponible ainsi que des explications plus exhaustives au sujet des différentes technologies utilisées. Des animations mettront en lumière les concepts clés à retenir. Les choix technologiques à utiliser concernant la mise en place du site web sont laissés libre.

L'ensemble du projet se fait en autonomie et avec l'accompagnement des enseignants et intervenants chargés du projet. Une méthode pour suivre l'avancée du projet et maintenir des échéances au sein de ce dernier pour s'assurer de son bon déroulement sera nécessaire.

III - Gestion du projet

Analyse du diagramme de Gantt

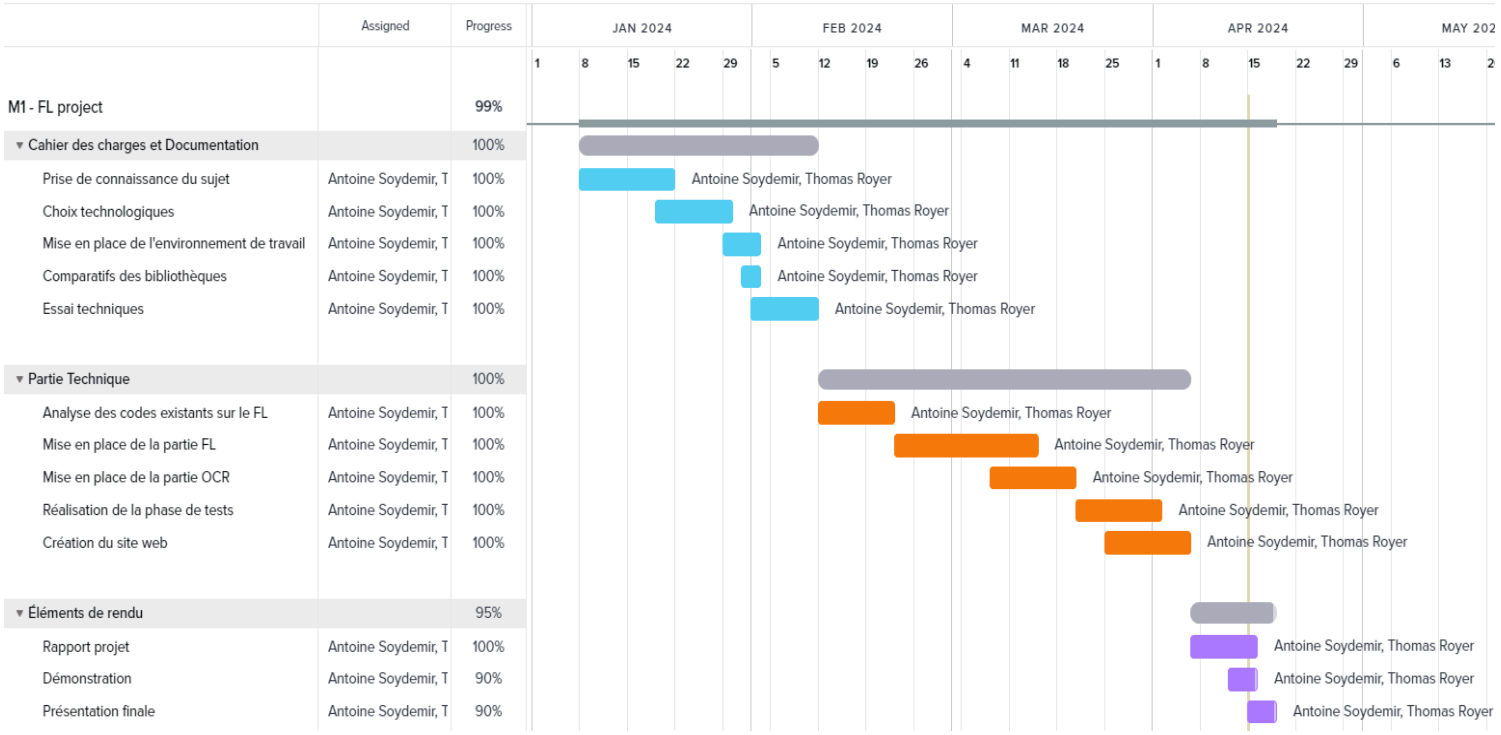


Figure 1 - Diagramme de Gantt

Pour la partie gestion de projet, un diagramme de Gantt a été réalisé pour permettre d'avoir un suivi dans l'avancement des tâches. Ce diagramme est adapté car il permet d'avoir une visualisation claire des tâches. Il offre une vue d'ensemble des différentes tâches complétées, en cours et à venir. Il permet de définir une durée dans les tâches ce qui permet en interne de s'organiser de la meilleure des façons. Il permet aussi de revoir rapidement la durée attribuée aux tâches. Notamment dans les cas où une activité a été sous-évaluée et qu'il est nécessaire d'y attribuer davantage de temps. À l'inverse, dans les cas où l'on perd un temps important sur une fonction, il permet de nous en rendre compte et d'adapter en conséquence.

La première partie du projet concernait la prise de connaissance du cahier des charges et du sujet en lui-même. Les premières semaines du mois de janvier ont donc été consacrées à une découverte des différents outils et technologies que nous allions utiliser durant le projet. Nous avons également organisé des réunions et un espace de travail dédié avec l'enseignant référent du projet et l'intervenante

extérieure de l'entreprise OpenBee. Après avoir mis en place l'environnement de travail, nous avons commencé par étudier les différents choix technologiques. Nous avons pris le soin de faire valider nos choix par les encadrants du projet.

Très rapidement, nous nous sommes rendus compte que les technologies utilisées pour réaliser le projet sont des technologies récentes qui ne disposent que de très peu de ressources sur internet. Après plusieurs réunions avec les encadrants, nous sommes parvenus à trouver diverses sources sur Google Scholar avec des papiers scientifiques réalisés par des chercheurs. La dimension technique du projet étant élevée, nous avons pris le soin de réaliser des comparatifs entre les différents moyens techniques à notre disposition. Nous les avons présentés aux encadrants qui ont validé ces choix et nous ont permis d'avancer plus sereinement sur la suite des tâches. Durant cette première phase du projet, nous avons également tenté de rapidement commencer à coder une solution pour l'apprentissage fédéré. Cela a été une des erreurs que nous avons effectuées car le travail de recherche préalable n'était pas assez important et ne nous avait pas permis de comprendre tout le fonctionnement de l'apprentissage fédéré.

Une fois le cahier des charges et les choix technologiques effectués, nous avons pu débuter sereinement l'approche technique du projet. Nous avons commencé par mettre en place une analyse des différents codes existants basés sur les choix technologiques précédemment réalisés. En s'inspirant des différentes solutions existantes, nous avons construit notre propre code en assemblant le résultat des recherches et des compétences acquises lors des premières semaines de projet (mise en place de la base de données, transfer learning, script pour évaluer la qualité des résultats…). Nous avons sollicité l'aide des encadrants du projet durant cette période car nous avons rencontré des blocages sur plusieurs points techniques et particulièrement au sujet des ressources nécessaires pour obtenir des résultats dans un temps imparti. Les phases de tests en intelligence artificielle peuvent être chronophages et cela nous a fait prendre du retard sur la suite des tâches à compléter.

Après discussion, nous avons revu nos objectifs finaux pour respecter la date butoir fixée par le projet. Ceci a donc eu des répercussions sur la base de données sur laquelle nous devions tester notre solution. Nous nous sommes basés sur un seul type de données à savoir les chiffres au format manuscrit. Initialement, nous aurions aimé pouvoir traiter les caractères alphabétiques ainsi que les tableaux à l'aide de l'OCR.
Ensuite, nous avons réalisé une phase de tests intensifs sur différents cas d'usages dans le but de démontrer l'intérêt de l'apprentissage fédéré dans le cadre de la reconnaissance d'image. Ces tests ont été réalisés durant les dernières semaines du mois de mars, à cette date notre infrastructure était proprement définie et fonctionnelle pour réaliser l'apprentissage. Hormis des ralentissements vis-à-vis des ressources octroyées par Google Collab dont il sera sujet par la suite dans le

rapport, nous avons pu effectuer efficacement une grande batterie de tests permettant de respecter les attendus.

Dans les dernières semaines du projet, nous avons effectué la création d'un site web vitrine simple qui reprend les grandes technologies utilisées dans ce projet ainsi qu'un support pour la démonstration. Le site permet, à une personne n'ayant pas de connaissance au sujet de l'apprentissage fédéré, de se faire une idée rapide des grands axes technologiques utilisés et des secteurs d'application de cette dernière. Il propose également une démonstration visant à démystifier l'aspect "complexe" d'une technologie de cette envergure à travers des animations.

En somme, le diagramme de Gantt a été un élément central dans le projet pour nous permettre de suivre la progression dans les tâches et de s'adapter face aux différents imprévus auxquels nous avons été confrontés. Il a guidé notre progression du début à la fin et nous a permis de surmonter les obstacles techniques et à ajuster nos objectifs en fonction des contraintes rencontrées. Ce dernier nous à aussi permis de coordonner nos efforts et assurer une communication fluide au sein du projet.

Nous avons également utilisé Microsoft Teams comme outil pour partager les données et avoir une communication plus efficace avec les encadrants. Solution que nous avons adoptée les premières semaines mais que nous avons délaissée par la suite car l'un des encadrants ne parvenait pas à faire fonctionner l'outil. Nous nous sommes basés sur des échanges classiques par mail qui ont très bien fonctionné.

Retour sur la gestion du projet

Le premier élément clé pour garantir un bon déroulement et une avancée limpide est la communication. Ayant préalablement effectué des projets en commun, il a été plus facile de rapidement trouver des automatismes permettant de garantir une fluidité dans les échanges.

Nous avons mis en place une méthode en faisant des réunions chaque matin pour définir les objectifs et les tâches à réaliser dans la journée. Au cours d'une journée type, nous partagions la charge de travail équitablement de manière à avancer chacun de notre côté. Le sujet étant vaste, en fin d'après-midi, nous faisions un point sur les différents sujets traités dans la journée pour que notre binôme soit au courant de l'ensemble des points qui ont été travaillés. Les réunions de fin de journée de chaque fin de semaine étaient consacrées à la mise à jour de notre diagramme de Gantt. C'est également à ce moment-là que nous prenions des décisions concernant le temps accordé à chaque tâche du diagramme.

D'un point de vue technique et pour se partager les documents, nous avons utilisé la plateforme de communication Discord en mettant en place un serveur qui nous a permis de communiquer efficacement avec différents canaux de discussions selon le sujet. Ce serveur nous a également permis de stocker l'ensemble des documents sur lesquels nous avons effectué des recherches ainsi que les vidéos, documentations techniques des bibliothèques et également les rapports des différents tests effectués. Cet outil, peu commun dans le monde professionnel face à son principal concurrent "Slack" nous a séduit de par ses nombreuses fonctionnalités (épingler des messages, mentionner un utilisateur, organiser les canaux de discussions…) et surtout pour sa gratuité.

Nous avons pris la décision d'utiliser des outils qui sont accessibles hors environnement local pour pouvoir avancer le projet depuis chez nous également. Dans cette même démarche, les fichiers de développement ont été mis sur le drive permettant d'être visible par le groupe. Le choix de mettre en place un Github ne semblait pas pertinent du fait que l'ensemble de notre pile technologique se trouvait chez Google et que l'ensemble du développement des scripts a été réalisé sur Google Collab.

Nous avons également été confrontés à des difficultés durant le projet. Les ressources au sujet de l'apprentissage fédéré qui est une technologie nouvelle sont assez faibles. De ce fait, trouver les bonnes ressources pouvait rapidement devenir laborieux et déconcertant. À plusieurs reprises, nous avons été bloqués pendant plusieurs heures. Les phases de tests sont également très longues et peuvent s'avérer complexes à déboguer.

Le projet n’a pas engendré de coût supplémentaire. En faisant le choix d’utiliser une plateforme comme Google, les ressources matérielles nécessaires pour exécuter notre code étaient déjà disponibles. Il n’y a donc pas eu de gestion financière sur notre projet.

Pour le site internet, nous avons cette fois-ci privilégié un environnement qui nous était davantage familier. Pour l’outil de développement intégré, nous avons choisi Visual Studio Code couplé à Github comme plateforme de développement collaborative. Cette partie a été réalisée durant les dernières semaines du projet, nous avons segmenté le travail en répartissant les différentes pages à créer et leurs contenus respectifs. Pour cela, nous avons préalablement discuté et choisi quelle structure adopter dans l’agencement des pages. Ensuite, nous avons chacun développé nos parties respectives et les avons mises en commun en faisant des remarques constructives pour améliorer les tournures de phrases et la manière dont le sujet devait être vulgarisé pour une personne ne connaissant pas le sujet du projet.

En conclusion, malgré les difficultés rencontrées, notamment dans la recherche de ressources pour des technologies émergentes comme l’apprentissage fédéré, notre persévérance nous a permis d’obtenir des résultats satisfaisants et de monter en compétences sur une technologie d’avenir. Par ailleurs, la gestion des ressources, en optant pour des plateformes telles que Google et des outils comme Visual Studio Code couplé à Github, a contribué à une bonne efficacité dans l’ensemble du projet.

IV - Développement technique

Intelligence Artificielle

Depuis quelques années, que ce soit dans les médias ou les entreprises, tout le monde parle d'intelligence artificielle. Mais qu'est-ce que l'intelligence artificielle ? L'intelligence artificielle est un procédé logique et automatisé reposant généralement sur un algorithme et en mesure de réaliser des tâches bien définies. L'IA est répartie en un ensemble de familles. Notre projet utilise des concepts de Deep learning (apprentissage profond) mais avant d'en expliquer les détails, il faut tout d'abord comprendre qu'est-ce que le machine learning (apprentissage automatique).

Le Machine Learning consiste à laisser l'ordinateur apprendre quel calcul effectuer, plutôt que de lui donner ce calcul (c'est-à-dire le programmer de façon explicite). C'est en tout cas la définition du Machine Learning selon son inventeur Arthur Samuel, un mathématicien américain qui a développé un programme pouvant apprendre tout seul comment jouer aux Dames en 1959.

Le machine learning est un domaine de l'intelligence artificielle qui consiste à programmer une machine pour que celle-ci apprenne à réaliser des tâches en étudiant des exemples de ces dernières. D'un point de vue mathématique, ces exemples sont représentés par des données que la machine utilise pour développer un modèle. Par exemple une fonction du type F(x) =Ax + B.

Le but en machine learning est de trouver les paramètres A et B qui donnent le meilleur modèle possible. C'est-à-dire le modèle qui s'ajuste le mieux à nos données. Pour cela on programme dans la machine un algorithme d'optimisation qui va venir tester les différentes valeurs A et B jusqu'à obtenir la combinaison qui minimise la distance entre le modèle et les points.

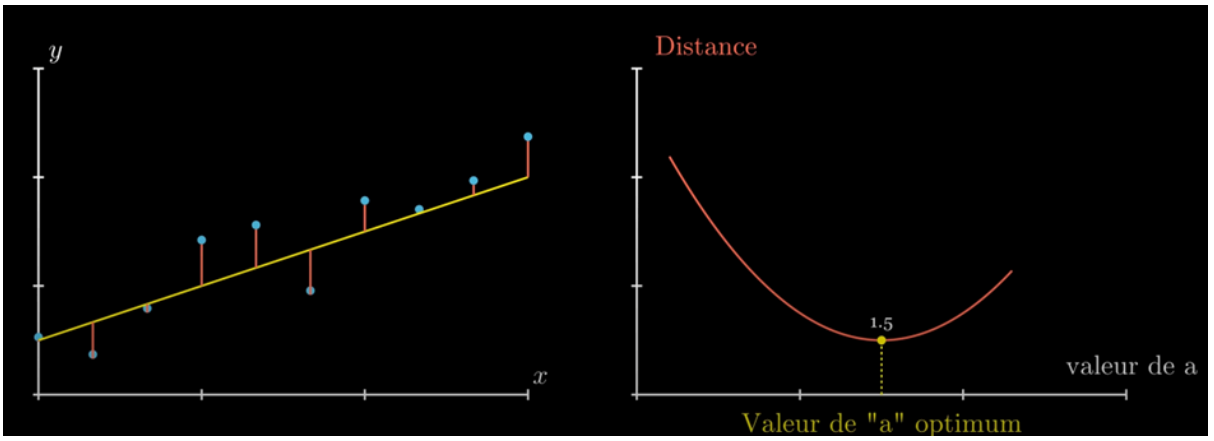


Figure 2 - Fonctionnement du machine learning

La nouvelle définition du machine learning de nos jours serait : développer un modèle en se servant d'un algorithme d'optimisation pour minimiser les erreurs entre le modèle et les données. Il y a une multitude de modèles qui viennent avec leurs propres algorithmes d'optimisation tels que le modèle linéaire avec l'algorithme de descente de gradients, les arbres de décision avec l'algorithme de CART ou encore les supports vecteurs machines avec son algorithme de marge maximum (les algorithmes cités ici sont des exemples parmi tant d'autres).

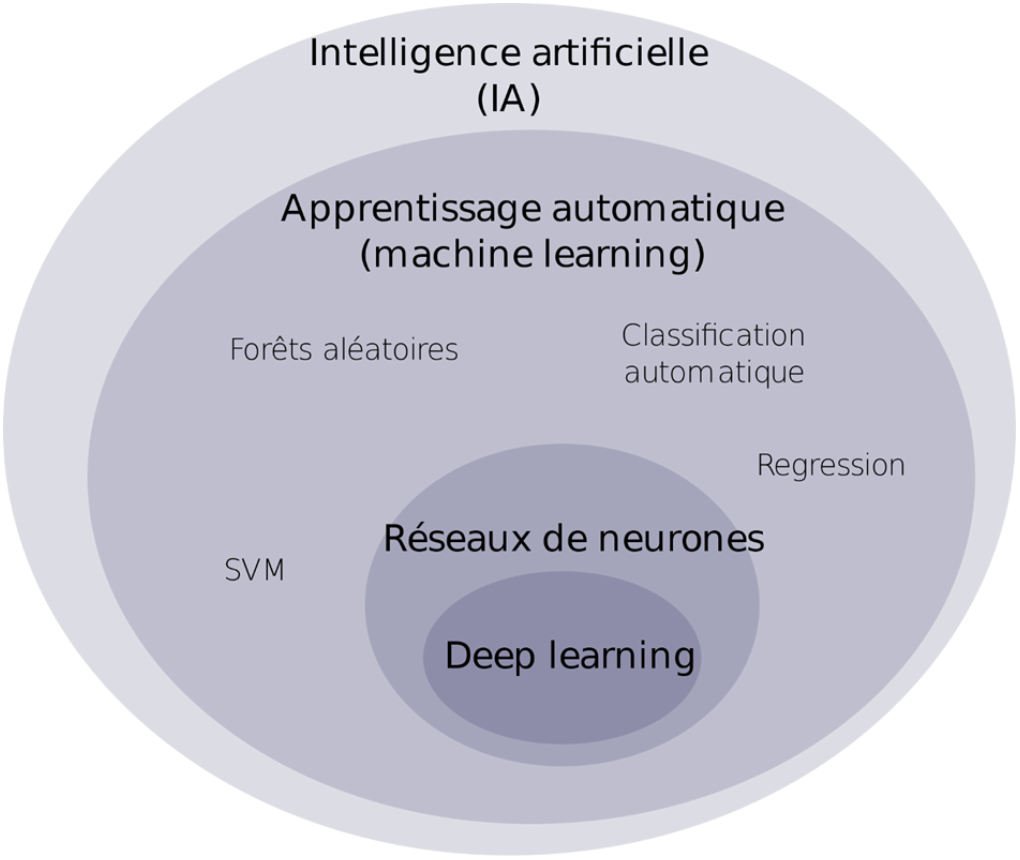


Figure 3 - Diagramme de Venn de l'IA

Deep learning

Maintenant qu'en est-il du Deep learning ? Le Deep learning est un domaine du machine learning dans lequel, au lieu de développer un des modèles comme on vient de citer, on développe des réseaux de neurones artificiels. Il faut savoir que le principe est exactement le même, c'est-à-dire qu'on fournit à la machine des données et elle utilise un algorithme d'optimisation pour ajuster le modèle à ces données. Mais cette fois-ci notre modèle n'est pas une simple fonction du type F(x) = Ax+B mais plutôt un réseau de fonctions connectées les unes aux autres que l'on nomme un réseau de neurones.

Les premiers neurones artificiels ont été inventés en 1943 par 2 mathématiciens et neuroscientifiques du nom de Warren McCulloch et Walter Pitts. Cette première version des neurones artificiels, comme expliqué dans leur article scientifique « A

logical calculus of the idies immanent in nervous activity », est basée sur le système des neurones humains. Pour rappel les neurones sont des cellules excitables connectées les unes aux autres et ayant pour rôle de transmettre des informations dans notre système nerveux.

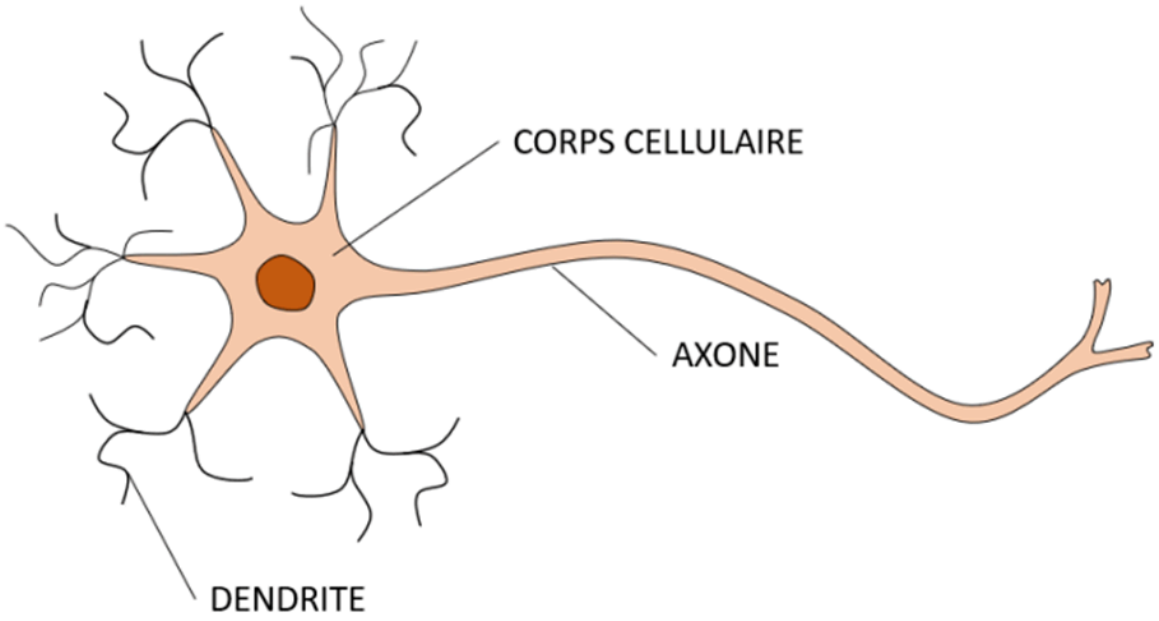


Figure 4 - Schéma d'un neurone biologique

Chaque neurone est composé de plusieurs dendrites, d'un corps cellulaire et d'un axone. Il faut savoir que les dendrites sont comme les portes d'entrée d'un neurone. C'est à cet endroit, au niveau des synapses, que l'information est captée par les neurones qui le précèdent. Une fois que cette information (signal) dépasse un certain seuil, le corps cellulaire s'active et produit un signal électrique envoyé dans l'axone jusqu'à la terminaison afin de le transmettre aux neurones suivants.

Ce que les 2 scientifiques ont essayé de faire est de modéliser mathématiquement ce fonctionnement en considérant que le neurone peut être représenté par une fonction de transfert qui prend en entrée des signaux x et qui retourne une sortie y.

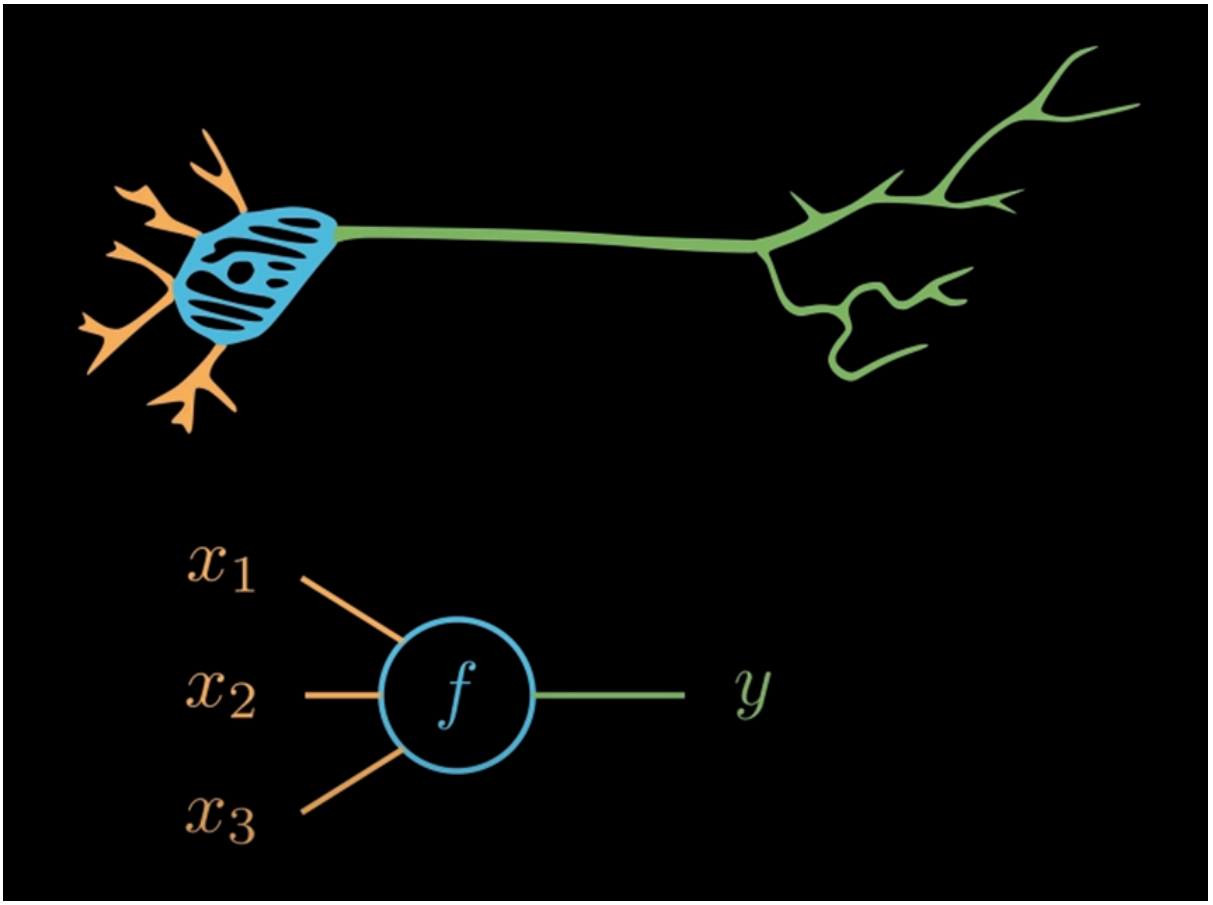


Figure 5 - Schéma montrant le rapport entre un neurone biologique et artificiel

Le problème de cette première approche d'un neurone artificiel, est qu'il ne pouvait prendre en entrée que des nombres binaires et qu'il n'existait pas à l'époque d'algorithmes d'apprentissage.

Il a fallu attendre 1957 pour que le scientifique Frank Rosenblatt améliore ce modèle en créant le premier algorithme d'apprentissage. Ce nouveau neurone artificiel est plus communément connu sous le nom de perceptron et est toujours utilisé de nos jours en deep learning. Pour créer le perceptron, Frank Rosenblatt s'est inspiré de la théorie de Hebb qui dit que lorsque 2 neurones biologiques sont excités conjointement, alors ils renforcent leur lien synaptique. C'est-à-dire qu'ils renforcent les connexions qui les unissent (en neurosciences, ce phénomène s'appelle la plasticité synaptique). À partir de cette idée Frank Rosenblatt a développé un algorithme d'apprentissage qui consiste à entraîner un neurone artificiel sur des données pour que celle-ci renforce ces paramètres à chaque fois qu'une entité X est activée en même temps que la sortie Y présente dans ces données.

Il a fallu ensuite attendre 1986 pour que Geoffrey Hinton invente le Perceptron Multicouche. Le problème du perceptron simple est qu'il résolvait des problèmes linéaires, alors qu'avec le perceptron multicouche, on peut résoudre des problèmes

polynomiaux. Le perceptron multicouche est un ensemble de perceptrons associé de bout en bout ce qui crée un véritable réseau de neurones.

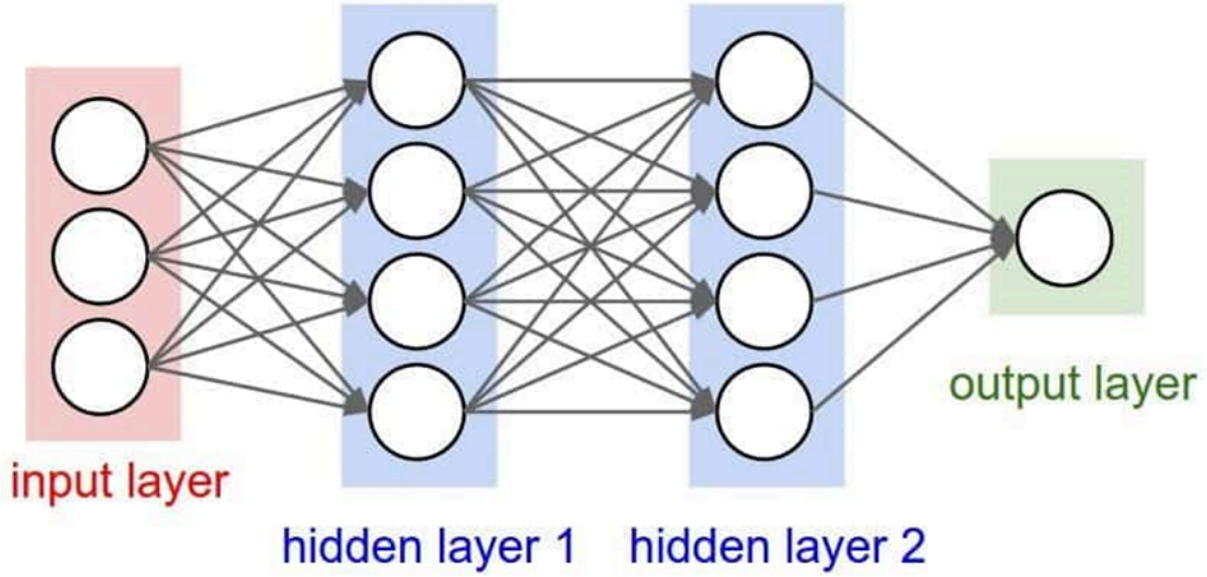


Figure 6 - Réseau de neurone artificiel

Le problème du perceptron multicouche a été de créer un algorithme d'apprentissage beaucoup plus complexe que le simple perceptron. La grande révolution fut la création de ce qu'on appelle la back-propagation qui consiste à déterminer comment la sortie du réseau varie en fonction des paramètres présents dans chaque couche. Autrement dit, comment le gradient de chaque couche évolue en fonction de sa couche précédente. Grâce au gradient, on peut mettre à jour les paramètres de chaque couche de telle sorte à ce que l'on minimise l'erreur entre la sortie du modèle et la réponse attendue. Et pour faire cela, on utilise un algorithme très proche de celui de Frank Rosenblatt appelé la descente de gradient.

On peut résumer le fonctionnement dans un réseau de neurones artificiels en 4 grands principes :

La forward propagation qui est un processus par lequel les données sont transmises à travers le réseau de neurones générant des prédictions à partir des entrées.

La fonction coût évalue l'écart entre les prédictions du modèle et les valeurs réelles, guidant ainsi l'optimisation pour minimiser l’erreur.

La back-propagation est un mécanisme où l'erreur de prédiction est propagée de manière rétroactive dans le réseau de neurones ajustant ainsi les poids de connexion.

Et enfin la descente de gradients qui est une méthode itérative qui utilise la pente de la fonction coût pour ajuster progressivement les paramètres du modèle, cherchant ainsi les valeurs qui minimisent l'erreur de prédiction.

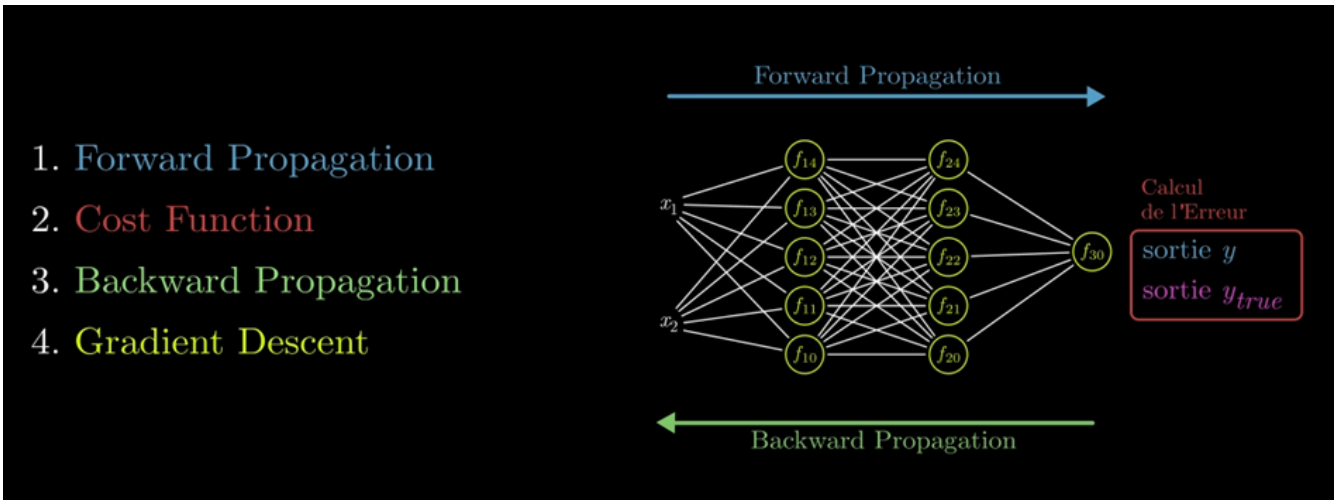


Figure 7 - Étape d'apprentissage d'un réseau de neurone artificiel

Au fur et à mesure du temps, le modèle du perceptron multicouche a évolué pour donner des modèles performants à des tâches précises. On peut nommer le CNN (Convolution Neural Network) qui est devenu célèbre après sa grande performance sur la base de données imagenet et plus généralement sur la détection d'images, ou encore les RNN (Récurrent Neural Network) qui permettent de résoudre des problèmes de séries temporelles comme la lecture de texte ou de l'audio.

OCR

L'un des axes principaux de projet a été de réaliser un OCR (Optical character recognition). L'OCR est un procédé qui permet de détecter des caractères et plus généralement un texte qu'il soit manuscrit ou dactylographié afin de le transformer en version numérique. L'OCR est très intéressant pour les entreprises, car il permet de répondre à deux problématiques. D'un côté, la digitalisation des données permet de ne pas les perdre et de mieux pouvoir les classer et d'un autre côté, pouvoir construire des Big data sur ces données numérisées afin de créer des modèles de prédiction ou d'une façon plus générale manipuler ces données avec de l'intelligence artificielle.

En intelligence artificielle pour faire de la reconnaissance de texte, il faut d'abord maîtriser l'analyse d'images. La plupart des technologies actuelles pour la détection d'images se basent sur le CNN (Convolution Neural Network). C'est un type de réseau de neurones particulièrement bien adapté pour le traitement d'image. Il est inspiré du fonctionnement du cortex visuel des animaux afin de reconnaître des motifs et de les hiérarchiser.

Un CNN est principalement composé d'une suite de couches de convolution et de couches de pooling. La couche de convolution est comme le cœur du CNN. Elle est un ensemble de filtres qui sont appliqués à l'image afin d'extraire des caractéristiques telles que les bords, les textures ou encore les motifs. Chaque filtre parcourt l'image en entrée et calcule la somme pondérée des pixels sous le filtre produisant ainsi une carte d'activation.

Après chaque opération de convolution, une fonction d'activation non-linéaire est appliquée pour introduire de la non-linéarité dans le réseau. Il faut comprendre que les fonctions d'activation sont appliquées à la sortie d'un neurone ou d'une couche de neurones, elles introduisent de la non-linéarité dans ce modèle qui permet de capturer les relations complexes entre les motifs dans les données. Il y a une multitude de fonctions d'activation, mais d'une façon générale pour CNN la fonction d'activation ReLU (Rectified Linear Unit), qui permet de transformer toutes les valeurs négatives en 0 et laisse les positives inchangées, est la plus optimisée.

S'ensuivent les couches de poolings utilisées pour réduire la dimension spatiale des cartes d'activation (générées par les couches de convolution). Elles permettent de réduire le nombre de paramètres dans le réseau et d'accélérer le processus d'apprentissage. Il faut comprendre que toutes ces étapes servent à décortiquer l'image et à repérer des patterns, des caractéristiques de l'image. Ce qui va faire la prédiction d'images est en réalité l'étape suivante.

Cette dernière étape que nous appellerons “couches entièrement connectées” est un ensemble de couches de neurones traditionnelles qui ont pour responsabilité la classification finale de la régression sur les caractéristiques extraites. Pour notre sujet d'application, ces couches ont pour but de déterminer la lettre, le mot ou le chiffre présent dans une image.

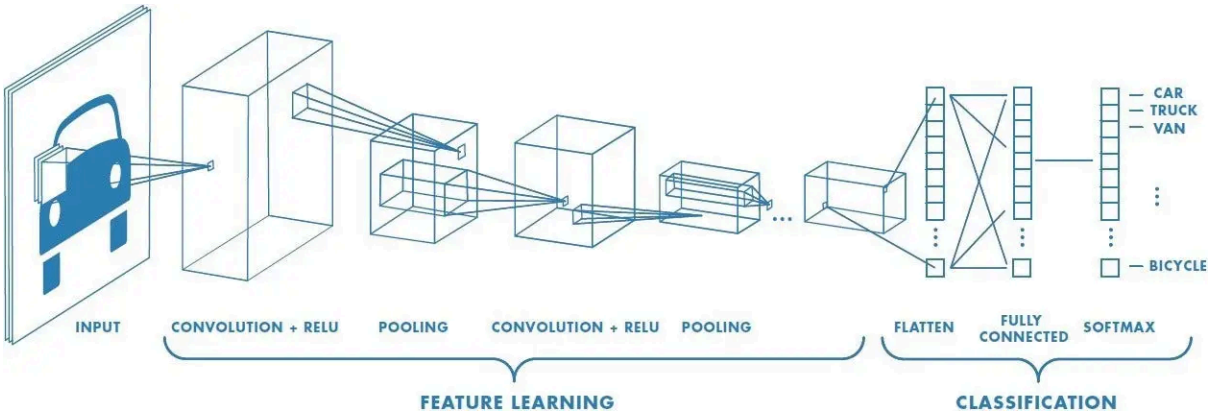


Figure 8 - Schéma d’un CNN

Transfer Learning

Le sujet du projet voudrait que l'on entraîne CNN sur une Big data afin de pouvoir reconnaître des caractères. Le problème est que nous ne disposons pas de matériel assez puissant pour pouvoir entraîner ces modèles avec des bases de données assez conséquentes pour former un modèle robuste. C'est pour cela que nous avons opté pour une technique d'apprentissage appelée le transfer learning.

Pour faire simple, le transfer learning consiste à prendre un modèle déjà existant qui est performant pour réaliser une certaine tâche et l'entraîner sur une nouvelle base de données plus petite afin de ne pas partir de zéro. La non modification des poids du modèle lors de la phase d'analyse constitue la performance de cette méthode. Par exemple pour le CNN, cette phase d'analyse a lieu lors du feature learning comme présenté sur le schéma du CNN où le réseau de neurones fait une suite de convolutions et de pooling pour détecter les formes sur une image.

Étant donné que le modèle choisi est performant pour faire de la détection, les nouvelles données vont uniquement entraîner les couches de classification afin que le modèle puisse reconnaître de nouveaux labels. Par exemple, si un modèle peut reconnaître une voiture, on peut appliquer un transfer learning sur ce modèle en l’entraînant sur de nouvelles données avec des images de camions afin qu'il puisse reconnaître cette nouvelle classe.

Une fois le transfert learning réalisé, il est conseillé de réaliser un fine tuning afin de modifier légèrement tous les poids du modèle pour s'affiner avec la nouvelle donnée. Le fine-tuning permet généralement d'améliorer les performances de quelques pourcentages, ce qui n'est pas négligeable dans le domaine du data science.

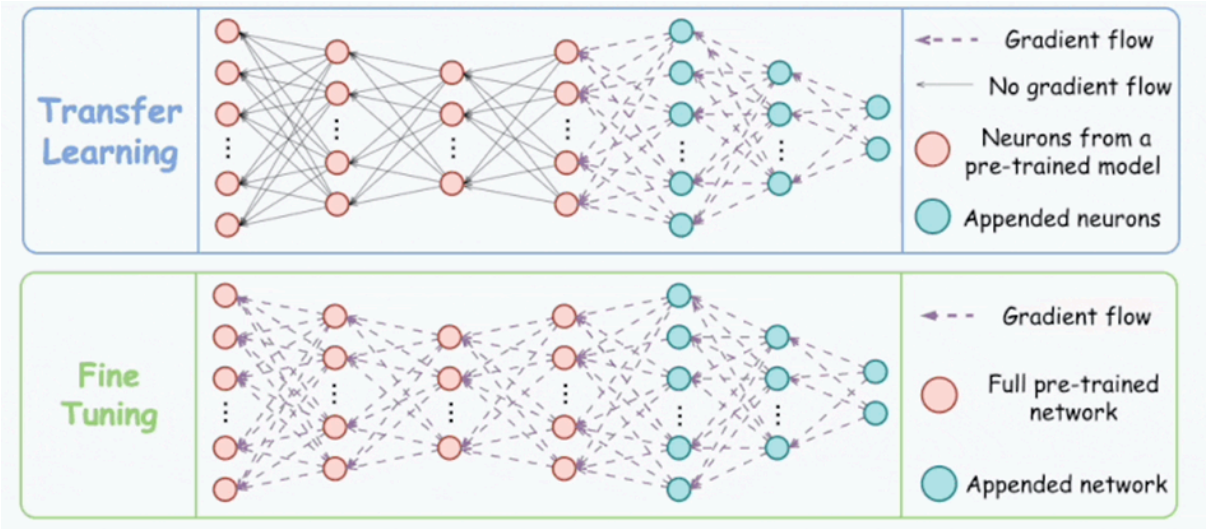


Figure 9 - Schéma explicatif du transfer learning et du fine tuning

On peut donc affirmer que le transfer learning est une technique d'apprentissage automatique dans laquelle un modèle pré-entraîné est utilisé comme point de départ pour une nouvelle tâche, généralement avec un ensemble de données différents. Il s'agit d'affiner les paramètres du modèle pré-entraîné sur les nouvelles données, en tirant parti des connaissances acquises lors de la tâche initiale pour améliorer les performances sur la nouvelle tâche. Cette approche est particulièrement utile lorsque les données étiquetées sont limitées ou lorsque l'entraînement d'un modèle à partir de zéro serait coûteux en termes de calcul, ce qui permet un développement plus rapide et plus efficace de modèles pour diverses tâches.

Federated Learning

Nous allons maintenant nous intéresser au federated learning, le sujet principal de notre projet. Tout d'abord, il faut comprendre que le federated learning est une philosophie d'apprentissage pour l'intelligence artificielle un peu comme l'orienté objet pour la programmation. En temps normal, lorsque l'on souhaite faire une intelligence artificielle, on collecte dans un premier temps de la donnée. Cette donnée est stockée sur un serveur central, dans le cas d'une entreprise. Afin de faire de la prédiction, des data-scientists vont développer un modèle qui va ensuite s'entraîner sur cette donnée. Après plusieurs rectifications du modèle et après une phase de validation qui retourne une précision (accuracy) satisfaisante, le travail est donc terminé. Le problème de cette méthode d'apprentissage est que l'entreprise qui développe le modèle doit avoir accès à des données. Les clients de cette entreprise doivent donc partager leurs données en toute transparence, ce qui soulève une importante préoccupation en matière de confidentialité.

En effet, une entreprise malveillante pourrait conserver les données des clients ou pourrait divulguer des informations les concernant. C'est pour cela que le federated learning est une solution à toute cette problématique.

Le federated learning est une philosophie d'apprentissage d'un modèle dans lequel les clients d'une entreprise n'auraient pas à envoyer leurs données à cette dernière. Le principe est simple, l'entreprise, que l'on va désormais appeler le serveur, va transmettre à tous ses clients un modèle de base avec des poids aléatoires (sur chaque neurone) qui répond à la problématique des clients (détection, prédiction). Pour avoir une idée simple dans un premier temps, chaque client va entraîner le modèle reçu par le serveur avec ses propres données puis envoyer les nouveaux poids des neurones générés après l'entraînement au serveur. Le serveur va donc recevoir une multitude de poids venant de plusieurs clients. Il va réaliser un calcul mathématique, tel que la moyenne des poids, afin d’obtenir un modèle qui serait théoriquement efficace pour toutes les bases de données des clients variés.

Il faut donc comprendre que le federated learning apporte beaucoup d'avantages. D'une part, il permet la protection des données et le respect des réglementations RGPD ce qui est une première en intelligence artificielle. D'un autre côté, il permet, pour l'élaboration d'un modèle, de disposer de plusieurs clients (donc de plus de données variées) ce qui rend le modèle extrêmement robuste notamment face à de nouvelles données qui pourraient s'ajouter (si le serveur ajoute un nouveau client par la suite).

Le volume de données est une problématique importante en intelligence artificielle. Un modèle a besoin de beaucoup de données pour être très performant, mais la machine qui entraîne ce modèle doit être très puissante pour pouvoir faire tous les

calculs. Puisque le federated learning implique l'apprentissage du modèle sur chaque client avec leurs propres bases de données, cela revient à répartir la puissance de calcul nécessaire entre chaque client tout en conservant l'énorme volume de données.

On peut dire que l'apprentissage fédéré inverse l'approche traditionnelle. Il permet l'apprentissage automatique sur des données distribuées en déplaçant l'information vers les données, au lieu de déplacer les données vers l'information.

Maintenant, rentrons beaucoup plus en détail sur le fonctionnement du federated learning. En réalité, pour entraîner un modèle avec du federated learning cela est plus compliqué que ce que l'on a expliqué précédemment. Il ne suffit pas simplement que les clients s'entraînent sur leurs données et envoient les poids au serveur.

L'apprentissage est séparé en round, chaque round, un certain nombre de clients, sont sélectionnés aléatoirement pour entraîner le modèle sur leurs données. Lors du premier round, les poids du modèle sont aléatoires, mais une fois le premier round terminé, les clients sélectionnés pour l'apprentissage, on finit d'entraîner le modèle sur leurs données, ils envoient les poids du modèle au serveur central. Le serveur va alors effectuer un calcul mathématique sur ces poids. Dans notre cas d'application, nous utiliserons le calcul qui se nomme fed average, qui consiste à faire la moyenne des poids pour chaque neurone.

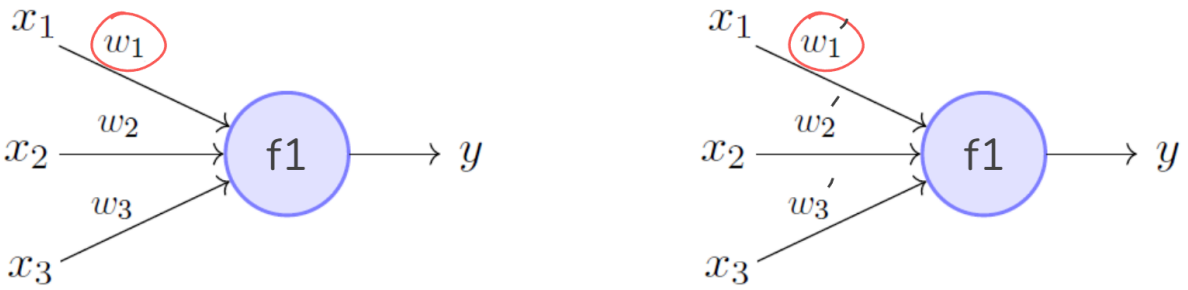


Figure 10 - Schéma montrant deux neurones similaires de deux clients différents avec des poids distincts

Il est crucial de comprendre qu'en parlant de faire la moyenne des poids, pour deux neurones F et F' situés au même emplacement dans le réseau de neurones mais sur deux clients différents, prendre la moyenne équivaudrait à créer un nouveau modèle identique avec comme nouveau poids :

w1* = (w1+w1')/2 , w2* (w2+w2')/2

Et ainsi de suite pour chacun des poids du modèle.

À la fin du calcul, on obtient donc un nouveau modèle dont chacun des poids correspond à la moyenne de tous les modèles des clients sélectionnés pour l'entraînement. Une fois cela terminé, le serveur va sélectionner de nouveaux clients pour faire une évaluation. Cette évaluation permet de générer des métriques afin d'observer l'évolution de la performance du modèle. Une fois le round terminé, ce processus est répété en sachant qu'à chaque fois que des clients sont sélectionnés que ce soit pour l'apprentissage ou pour la validation, ils reçoivent du serveur les nouveaux poids du modèle. Ainsi, le nombre de rounds peut être déterminé par la stratégie ou peut s'arrêter lorsque les poids du modèle convergent.

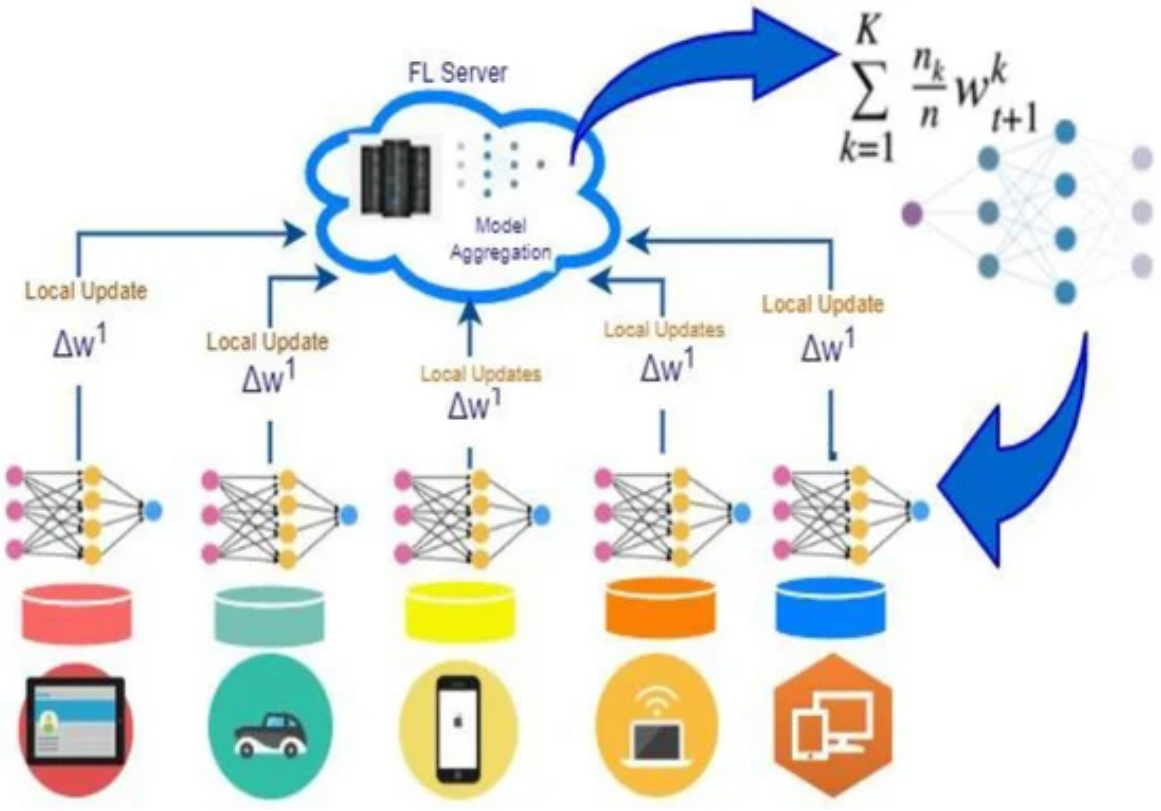


Figure 11 - Schéma explicatif du federated learning

Le serveur central va suivre ce que l'on nomme une stratégie. La stratégie en federated learning comporte des paramètres importants qui vont définir comment va se dérouler l'apprentissage. Certains paramètres sont indispensables comme le nombre de clients minimum pour l'entraînement ou la validation.

Un paramètre, très important, demande le pourcentage de clients utilisés pour l'entraînement et la validation à chaque round. En effet, à chaque round, le serveur ne va pas demander à tous les clients de s'entraîner, car cela ne serait pas très optimisé. Prenons un exemple : Imaginons qu'il y ait cent clients. Supposons que la

stratégie prévoit qu'à chaque cycle, dix pour cent des clients seront sélectionnés de manière aléatoire pour l'entraînement et cinq pour cent pour la validation. S'il y a donc plus de dix rounds programmés dans la stratégie, dans le meilleur des cas tous les clients auront servi d'entraînement au moins une fois. D'un autre côté, il est préférable d'entraîner peu de clients à chaque fois, mais avec un grand nombre de rounds que l'inverse puisque qu'à chaque round le serveur envoie le nouveau modèle avec des poids de plus en plus performants. Le pourcentage de clients est à définir à l'avance par le data scientist en fonction de beaucoup de paramètres notamment le nombre de données que dispose chaque client, le nombre total de clients et la puissance de calcul nécessaire.

La stratégie demande un nombre minimum de clients disponibles par rapport au nombre total de clients pour effectuer un federated learning. La stratégie permet aussi l'acquisition des métriques afin d'avoir un rendu visuel sur l'avancement de la performance du modèle.

La particularité principale de la stratégie est de définir comment va s'effectuer le calcul mathématique une fois que le serveur aura récupéré l'ensemble des poids des modèles entraînés par les clients à chaque round. Il existe une multitude de calculs de stratégie comme fed average que l'on a expliqué précédemment. Si fed average fait la moyenne des poids de chaque neurone, il ne prend pas en compte la quantité de données que contient chaque client. En effet, si un client possède dix fois plus de données qu'un autre, dans un cas normal le modèle sera dix fois plus influencé par les données de ce client.

Or avec le federated learning, la stratégie de fed average effectue la moyenne des poids de chaque neurone entre les deux modèles de ces deux clients. Cela revient à ne pas prendre en compte le fait qu'un des deux clients possède davantage de données, ce qui pourrait permettre un entraînement plus efficace du modèle.

C'est pourquoi il existe une autre stratégie qui se nomme fed average pondéré. Cette stratégie est particulièrement intéressante, car elle prend en compte la différence de quantité de données de chaque client afin de la pondérer lors du calcul de la moyenne. Un client qui aura plus de données verra les poids de son modèle plus influent dans le calcul.

Lors de la validation, si un client obtient une précision satisfaisante (par exemple quatre-vingt pour cent), tandis qu'un autre client, ayant nettement moins de données, retourne une précision faible (par exemple vingt pour cent), cette méthode ne se contentera pas de calculer la moyenne (80 + 20)/2 = 50%. Elle prendra en considération le fait que le premier client dispose par exemple de dix fois plus de données. Il existe de nombreuses autres stratégies envisageables dans ce contexte.

On pourrait citer FedProx qui n'est pas beaucoup différent de fed average, cette méthode ajoute un terme de régulation au problème d'optimisation fédéré afin de mieux prendre en compte les différences locales entre les clients.

FedMA (Federated Model Averaging), au lieu de simplement agréger les mises à jour de chaque client par moyenne, prend également en considération la similarité entre les modèles locaux afin de pondérer plus efficacement les mises à jour.

FedDyn (Federated Learning with Dynamic Selection) est une méthode qui sélectionne dynamiquement les clients à chaque round en fonction de leur performance locale ou de la qualité de leurs données.

FedRamp consiste à atténuer le déséquilibre des données entre les clients en introduisant une phase de rééquilibrage des données. En effet, chaque client a son propre ensemble de données locales qui peuvent être différentes en taille, en qualité, et même en distribution par rapport aux autres clients.

Il faut bien sûr choisir une stratégie en fonction des paramètres (type de modèle, nombre de clients, qualité/quantité de la donnée) et des problématiques. Une fois la stratégie bien établie et codée, il suffit de lancer un apprentissage fédéré pour créer un modèle tout en garantissant la confidentialité des données de chaque client.

Choix technologiques

Durant la première phase de développement du projet, il était important de faire des choix technologiques juste et en adéquation avec les attendus du projet. Tout d'abord, l'un des premiers choix à faire se porte sur le choix de la bibliothèque (framework) principal. Parmi les frameworks existants, les plus populaires pour le deep learning et l'apprentissage automatique sont Tensorflow et Pytorch.

Tensorflow présente de nombreux avantages, d'abord par le fait que ce soit un framework très adopté et largement utilisé dans l'industrie de nos jours. De ce fait, il dispose d'une grande communauté de développeurs actifs et de nombreux outils et ressources accessibles pour des débutants. C'est un framework flexible qui est adapté pour construire et déployer des modèles d'intelligence artificielle sur diverses plateformes tel que le web ou encore mobile. Sa documentation rend ce framework facile à prendre en main. Souvent accompagnés d'exemples concrets avec du code pour mieux imager les concepts clé de ce dernier. Tensorflow est un framework développé par Google et qui a été rendu open source en février 2017. Toutefois, la bibliothèque comporte également certains inconvénients, lorsqu'il s'agit de développer certaines fonctionnalités complexes, elle peut très rapidement devenir difficile à utiliser de par sa syntaxe déroutante. On pourra également noter que selon les types d'entraînements que l'on souhaite appliquer à un modèle d'intelligence artificielle, Tensorflow peut présenter des lacunes en termes de performances de rapidité et de ressources consommées par la mémoire de l'ordinateur comparé à Pytorch.

Pytorch est une bibliothèque Python open source développée par Meta. Cette bibliothèque, très populaire auprès des chercheurs présente de grande qualité de flexibilité notamment au niveau de la manipulation des modèles d'intelligence artificielle. La communauté de Pytorch, moins nombreuse que celle de Tensorflow reste très active et propose de nombreuses ressources et des moyens de support. La grande différence avec son principal concurrent se fait au niveau de son graphe de calcul qui est dynamique. Cela permet de rendre les débogages et les résultats plus intuitifs pour l'utilisateur.

Après avoir effectué des recherches préalables, le choix de la bibliothèque s'est porté vers Tensorflow. Plus accessible pour les débutants et d'après les recherches effectuées, elle semble être avantageuse pour réaliser de l'apprentissage fédéré.

Les performances en termes de précision pour l'entraînement d'un modèle utilisant CNN sont en faveur de Tensorflow. Malgré le fait que Pytorch soit plus rapide purement en termes de temps d'exécution, nous avons fait le choix de privilégier la précision à la vitesse d'exécution, car la base de données utilisée ne possède pas

une grande quantité de données et donc le temps d'exécution n'est pas un paramètre prioritaire dans le cas de notre étude. Les exemples étaient également plus nombreux ainsi que sa communauté vaste et active et donc le choix s'est porté vers cette bibliothèque.

Feature	PyTorch	TensorFlow
Datasets and pre-trained models in torchtext, touch audio, and torchvision	**Library of Datasets and pretrained models**	Datasets and pre-trained models in torchtext, torchaudio, and torchvision
Deployment	TorchServe for serving machine learning models	TensorFlow Serving and TensorFlow Lite for model deployments
Model Interpretability	PyTorch Captum	tf-explain
Privacy-Preserving Machine Learning	PyTorch Opacus for differentially private model training	TensorFlow Federated for federated machine learning
Ease of Learning	Requires intermediate proficiency in Python	Relatively easier to learn and use

Figure 12 - Tableau comparatif entre différentes bibliothèques d'IA

Un autre choix technologique crucial dans ce projet est celui de la bibliothèque d'apprentissage fédéré. Il existe de nombreuses bibliothèques, mais il était important de faire un choix en adéquation avec le reste des technologies adoptées pour mener le projet. En effet, la bibliothèque d'apprentissage fédérée doit être choisie en fonction du framework précédemment choisi. Les critères sur lesquels il faut se baser pour choisir la bibliothèque la plus adaptée pour réaliser de l'apprentissage fédéré sont les suivantes :

- Fonctionnalités de base : la bibliothèque doit offrir les fonctions de base comme l'entraînement côté client, l'agrégation côté serveur et une communication efficace.
- Le support des modèles : en intelligence artificielle et plus particulièrement en deep learning il existe de nombreux modèles d'apprentissage. Il est important

de prendre cet élément en considération et être sûr que la bibliothèque supporte le ou les modèles utilisés dans le projet.
- Extensibilité : Il s'agit de la capacité à pouvoir ajouter des algorithmes d'agrégation personnalisés.
- Méthodes de confidentialité : certaines bibliothèques offrent des méthodes supplémentaires pour renforcer le caractère privé des données.
- Utilisation commerciale : il peut être important de prendre ce paramètre en compte si le projet envisage d'être commercialisé. Certaines bibliothèques offrent des avantages (déploiements, suivis et orchestration des modèles) qui peuvent se révéler intéressants pour une utilisation commerciale.

Basées sur les différents paramètres cités précédemment, plusieurs bibliothèques parmi les plus populaires ont été retenues. Chacune présente différents avantages et certains inconvénients.

NVFlare :

Il s'agit d'un framework développé par l'entreprise Nvidia et destiné à une utilisation "business". Cette bibliothèque supporte une variété de modèles et a été conçue pour développer des produits commerciaux avec de nombreuses fonctionnalités. Il est notamment spécialisé dans la transition des modèles de "Machine Learning" (ML) centralisés aux modèles dit "Federated Learning" (FL). Il est donc hautement flexible, mais nécessite des efforts importants pour comprendre et adopter l'architecture.

FATE :

FATE est une bibliothèque prête pour l'industrie développée par WeBank. Elle propose de nombreux modèles pré-construits pour l'apprentissage, les statistiques et le prétraitement. Néanmoins, après de multiples recherches, on peut observer que cette dernière semble moins flexible que d'autres bibliothèques et une documentation qui semble incomplète dans sa version anglaise.

Flower :

Il s'agit d'une bibliothèque d'apprentissage fédéré flexible et facilement compréhensible. Elle semble idéale pour la recherche et possède des capacités d'extension et de personnalisation importantes. Adapté pour des projets d'étude et de recherche. Cependant, elle présente des lacunes en termes de fonctionnalités avancées et supplémentaires. La documentation est accessible aux débutants et complète.

Pysyft :

C’est un framework open-source développé par le projet OpenMined. Cette bibliothèque est en cours de développement. Elle présente de nombreux aspects positifs, notamment pour la protection de la vie privée et la fédération des données.

En vue du cahier des charges, notre choix s’est porté vers Flower. Effectivement, au vu des contraintes de temps imposées par le projet, il était judicieux de se tourner vers une solution capable d’être mise en place rapidement qui est reconnue auprès des développeurs de la communauté. Cette bibliothèque propose des fonctionnalités de bases et des fonctionnalités avancées qui permettent de ne pas être bridés dans le cas d’un projet de recherche. Elle peut être déployée dans un environnement utilisant Tensorflow ou Pytorch ce qui est notre principale contrainte. Toutefois, des technologies telles que NVFlare ou Pysyft sont notables de par les grandes possibilités qu’elles offrent.

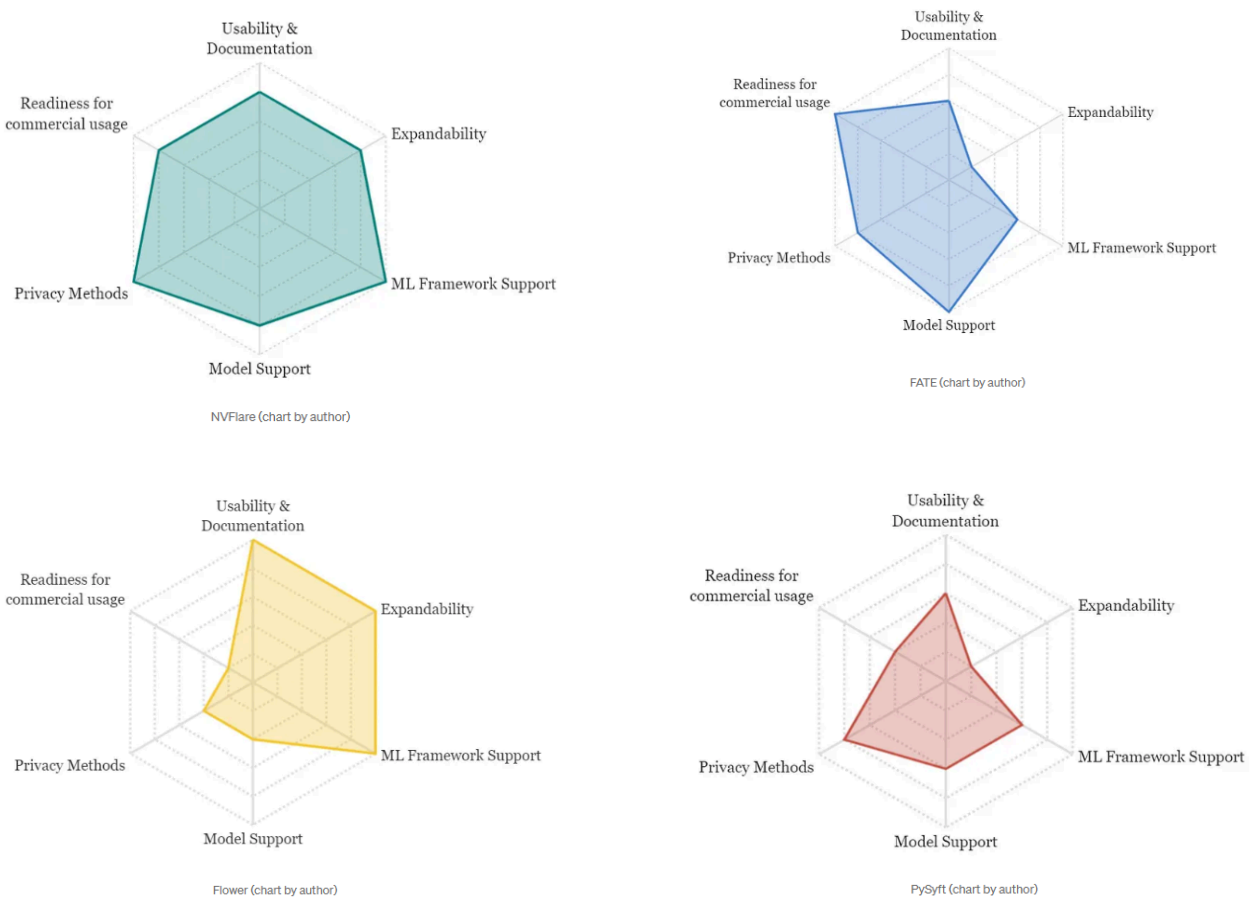


Figure 13 - Diagrammes des performances des librairies pour le federated learning

Le choix technologique de la plateforme s’est porté vers Google Collab. Il s’agit d’une plateforme développée par Google qui propose un hébergement Jupyter Notebook ergonomique. Cela permet de développer dans un environnement cloud,

car Google met à la disposition des utilisateurs des ressources processeurs, mémoire vive, et même carte graphique gratuitement. Utiliser cette solution permet plusieurs avantages : tout d'abord vis-à-vis de la gestion des fichiers. L'ensemble des fichiers utilisés pour développer son hébergé dans un espace cloud dédié à chaque utilisateur et qui peut être partagé entre les membres de l'équipe. Les bibliothèques sont également hébergées par la plateforme. Tout ceci permet d'économiser les ressources de sa machine locale. Pourtant, cette plateforme à également ses limites. La version gratuite offerte par Google ne donne pas accès à des ressources illimités et cela peut être vite contraignant lorsqu'il s'agit d'exécuter des scripts qui peuvent être gourmands en ressources, ce qui est habituellement le cas lorsque l'on réalise de l'intelligence artificielle sur des bases de données importantes.

Le choix s'est porté vers Google Collab plutôt qu'un environnement de développement classique tel que Visual Studio Code, car la gestion des fichiers de manière centralisée et les ressources mises à disposition par la plateforme nous ont paru plus pertinentes.

Développement de la solution

Premier modèle et base de données

Revenons au contexte. Notre objectif était de développer une intelligence artificielle entraînée par la technique du federated learning pour lire du texte sur une image, ce qu'on appelle également un OCR (Optical Character Recognition). Dans un premier temps, avant même d'entraîner un modèle avec du federated learning, il nous a fallu entraîner un modèle sur une base de données qui permettait de reconnaître des caractères sur une image. En intelligence artificielle, il existe un type de modèle qui permet d'analyser des images. Ce type de modèle se nomme CNN (Convolution Neural Network) et comme expliqué dans la partie sur les OCR, ce modèle permet de reconnaître les patterns, les formes afin d'en faire des déductions.

C'est pourquoi nous sommes partis sur une architecture d'un CNN simple comme présenté ci-dessous.

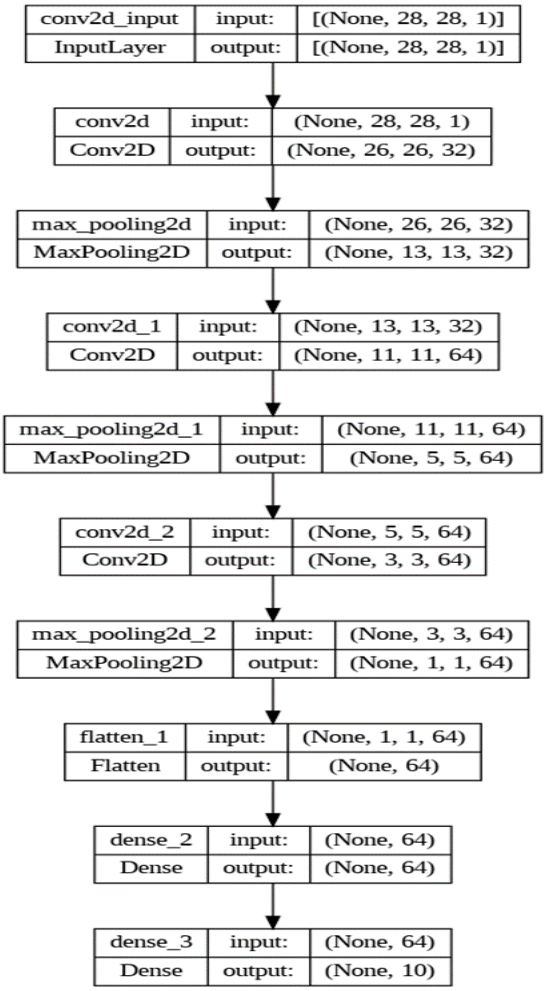


Figure 14 - Architecture d'un CNN

Ce schéma nous enseigne que les dimensions en entrée correspondent au nombre de pixels (dans ce cas, une image de vingt-huit par vingt-huit pixels), tandis que la dimension en sortie de la dernière couche de neurones correspond au nombre de classes à prédire (dans ce cas, dix). Ces paramètres sont à changer en fonction de la base de données utilisée pour l'apprentissage. Pour ce qui est des couches intermédiaires, également appelées couches cachées, une explication plus détaillée est fournie dans la section précédente sur l'OCR.

Nous avons ainsi opté pour l'entraînement de ce modèle en utilisant la base de données MNIST, célèbre en intelligence artificielle, notamment dans le domaine de la vision par ordinateur. Son nom provient de Modified National Institute of Standards and Technology et elle est issue d'une version modifiée et normalisée de la base de données NIST, spécialisée dans la reconnaissance de chiffres manuscrits. Cette base de données comprend soixante-dix mille images de chiffres manuscrits allant de zéro à neuf, en noir et blanc. Chaque image mesure vingt-huit par vingt-huit pixels et est divisée en deux ensembles de données : soixante mille pour l'apprentissage et dix mille pour la validation.

Cette base de données est très pratique et efficace en raison de son grand nombre de données, de la faible dimension des images et du petit nombre de labels, soit dix au total. Nous avons entraîné le modèle CNN sur cette base de données et avons obtenu des résultats très satisfaisants, dépassant les quatre-vingt-douze pour cent de précision.

Le problème étant que MNIST est une base de données simple, pour faire de l’OCR, il nous faudrait une base de données comportant du texte. C'est pour cela que nous avons travaillé un long moment sur une base de données appelée IAM.

IAM, Institute of Computer Science and Applied Mathematics est une base de données contenant un ensemble d'images de textes avec leurs labels. Très efficace pour la reconnaissance optique de caractère, cette base de données était trop complexe à analyser par un simple modèle CNN présentée ci-dessus. De plus, contrairement à MNIST, les dimensions des images sont plus grandes et variables ce qui complexifie grandement la tâche. En effet, redimensionner une image peut faire perdre de l'information sur cette dernière.

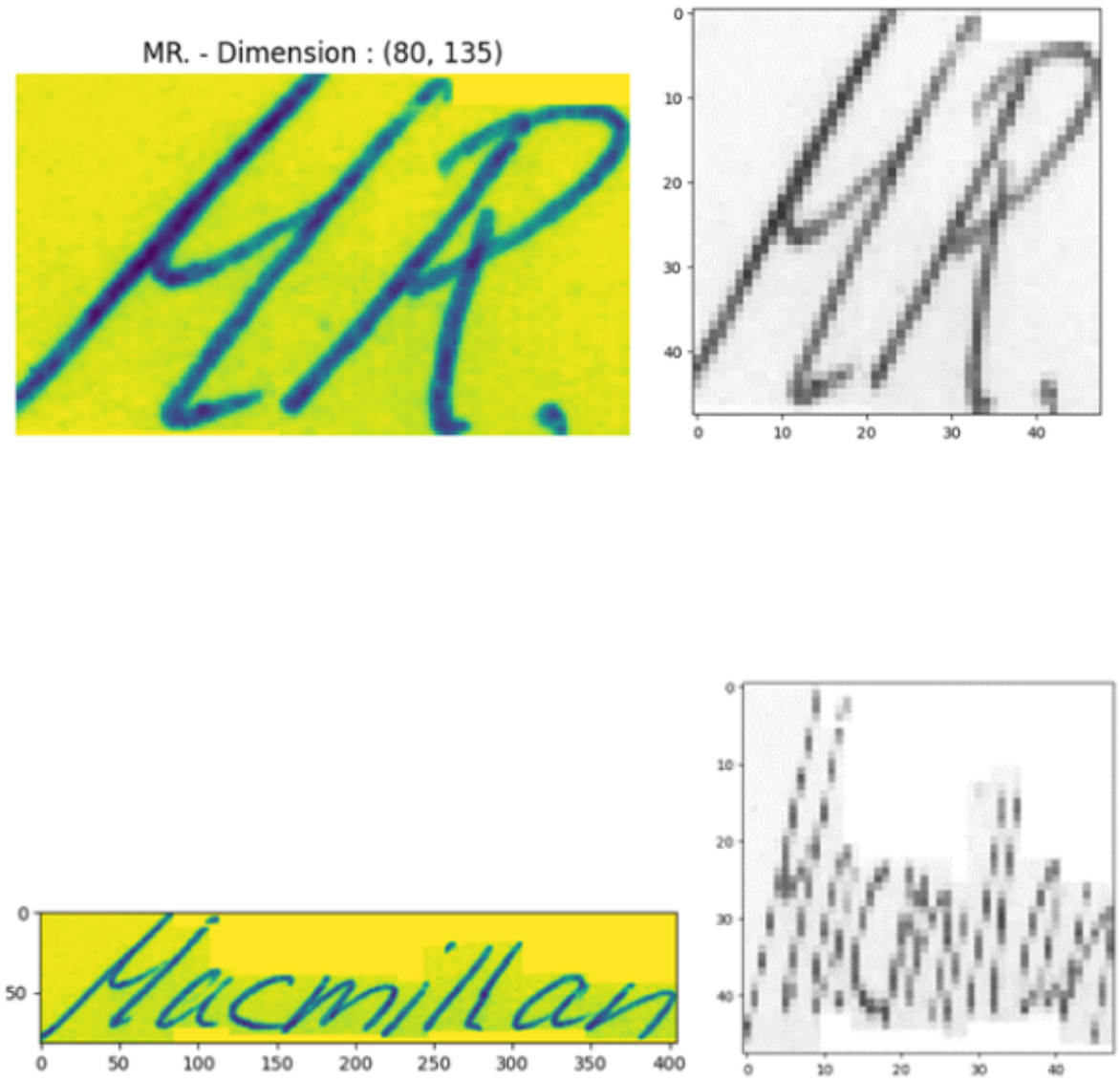


Figure 15 - Images montant l'impact d'un reshape sur une image

Comme vous pouvez le constater ci-dessus, nous avons redimensionné des images de dimensions variables en quarante-huit par quarante-huit pixels. Sur la première image, aucune perte d'information n'est perceptible, contrairement à la deuxième image qui était initialement beaucoup plus large.

Aussi, cette base de données recense des mots et pas des lettres, il y a donc une infinité de labels ce qui complexifie encore plus la tâche pour le modèle. Dû au grand nombre de contraintes de cette base de données et le peu de puissance de calcul à disposition pour notre projet, nous avons donc décidé d'abandonner l'idée d'entraîner un modèle sur la base de données IAM et de poursuivre avec MNIST.

Nouveau modèle

Nous avons envisagé l'idée de consacrer plus de temps à une autre base de données plus intéressante. Par conséquent, nous avons opté pour le remplacement du modèle CNN simple par un CNN plus sophistiqué. Puisque Google Colab ne dispose pas de performances importantes pour créer un modèle avec de nombreuses couches, nous avons pris la décision d'opter pour le transfer learning. L'explication complète du transfer learning se trouve dans la partie qui porte son nom dans le compte-rendu. Nous avons donc opté pour le célèbre modèle VGG16.

VGG16 a été présenté dans un article de recherche intitulé "Very Deep Convolutional Networks for Large-Scale Image Recognition", par Karen Simonyan et Andrew Zisserman en 2014. Le modèle VGG-16 est une architecture de réseau neuronal convolutive (CNN) proposée par le Visual Geometry Group (VGG) de l'Université d'Oxford. Il se caractérise par sa profondeur, composé de seize couches, dont treize couches convolutives et trois couches entièrement connectées. Le modèle est réputé pour sa simplicité et son efficacité, il obtient d'excellentes performances dans diverses tâches de vision par ordinateur notamment la classification d'images et la reconnaissance d'objets.

L'architecture du modèle se compose d'une pile de couches convolutives suivies de couches de mise en commun maximale, dont la profondeur augmente progressivement. Cette conception permet au modèle d'apprendre des représentations hiérarchiques complexes des caractéristiques visuelles, ce qui conduit à des prédictions robustes et précises. L'ImageNet Large Scale Visual Recognition Challenge (ILSVRC) est une compétition annuelle dans le domaine de la vision artificielle où les équipes s'attaquent à des tâches telles que la localisation d'objets et la classification d'images. VGG16 a obtenu les meilleurs classements dans les deux tâches, en détectant des objets de deux cents classes et en classant des images dans mille catégories. Nous avons donc décidé d'utiliser ce modèle très performant pour y faire du transfert learning afin de lui apprendre à détecter des caractères.

Maintenant que nous avons le modèle ainsi que la base de données, il nous faut entraîner ce modèle avec la technique du federated learning. Pour cela nous avons utilisé la bibliothèque Flower qui permet notamment de créer des simulations. En effet, il nous est impossible de créer des clients réel sur Google collab. La simulation offerte par Flower nous permet de générer des clients virtuels, ce qui simplifie considérablement notre travail. La fonction simulation prend en compte des arguments comme le nombre de clients, le nombre de rounds, les ressources nécessaires pour la simulation (cpu, gpu) ainsi que la base de données qui va être répartie chez chaque client. Le dernier paramètre le plus important est bien évidemment la stratégie.

Nous avons décidé d'utiliser la stratégie fed average qui semble la plus adaptée dans notre cas d'utilisation. Dans un souci de performance, nous avons opté pour cinq clients. À chaque round deux clients seront utilisés pour l'entraînement et un pour la validation. Enfin nous avons stipulé dans la stratégie une fonction évaluation fédérée afin d'obtenir par la suite un graphique de l'avancement de l'accuracy à chaque round sur la base de validation, ainsi qu'une fonction d'évaluation globale qui évalue le modèle à chaque fin de round sur l'ensemble de la base de données. Ce qui n'est théoriquement pas possible dans un cas d'application réelle vu que le serveur ne peut pas avoir accès aux données de chaque client.

Pour notre simulation, nous avons décidé de n'utiliser que les 5000 premières données de MNIST répartis sur cinq clients. Pour faire au plus simple, le premier client possède les 1000 premières images, le deuxième les images de mille à deux-mille ainsi de suite. Il y a aussi un super client qui détient les 5000 premières données afin de pouvoir faire l'évaluation globale comme expliquée précédemment.

La dernière étape est de reshape nos données. En effet, MNIST a pour dimension de base 28x28x1 pixels. La troisieme dimension d'une image correspond a la couleur c'est-à-dire que si elle est égale à un alors l'image sera en noir et blanc, si la dimension est de trois, alors l'image sera en couleur RGB (chaque dimension correspond à une teinte de couleur Red Green Blue). Nous avons dû modifier la dimension des images de 28x28x1 à 48x48x3 afin de permettre au modèle VGG16 de les traiter de manière appropriée.

Voilà maintenant que tout cela est fait, il ne nous reste plus qu'à entraîner le modèle et observer les résultats.

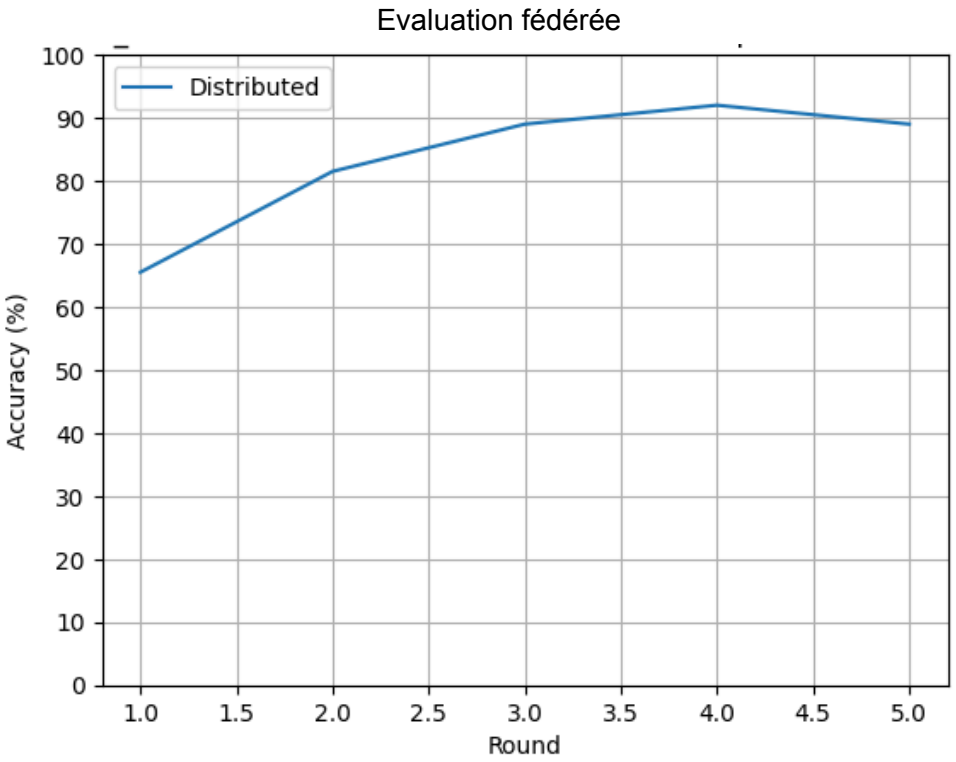


Figure 16 - Graphique montrant l'évolution de l'accuracy en fonction des round dans le cas de la validation fédérer

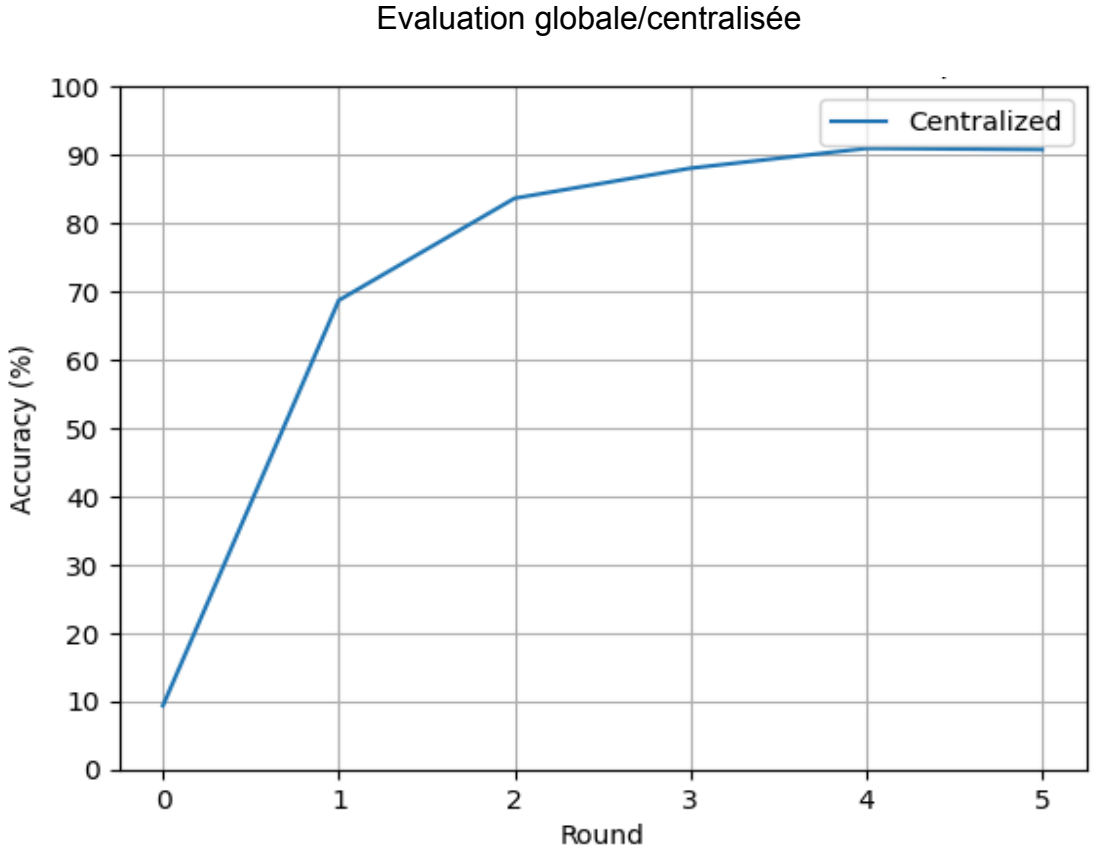


Figure 17 - Graphique montrant l'évolution de l'accuracy en fonction des rounds dans le cas de la validation sur l'ensemble du jeu de données

Nous avons donc obtenu deux figures. Comme expliqué précédemment, la première correspond à l'évaluation fédérée (le client choisit pour la validation) et la deuxième à l'évaluation centralisée (sur l'ensemble de la base de données). Dans un cas réel, le data scientist analysera la première figure car il ne disposera pas de la deuxième. Dans notre situation nous allons principalement nous intéresser à la deuxième figure car elle est plus représentative d'une réelle validation sur un ensemble de données.

À la fin du cinquième round nous obtenons une précision de 90% ce qui est très satisfaisant, mais qu'en est-il du même modèle entraîné sur la même base de données sans utiliser le federated learning. Nous avons donc expérimenté et obtenu un résultat de 92% ce qui est légèrement mieux mais sans respecter la confidentialité des données.

On pourrait penser que le federated learning est moins efficace qu'un apprentissage classique, mais examinons cela dans un contexte réel. Dans un cas réel où chaque client s'entraînait sur sa base de données avec le même modèle pour tous, s’ils font une validation sur l'ensemble des données (les leurs plus celles des autres clients sur lequel ils n'ont pas pu s'entraîner) alors le résultat est nettement plus bas. Nous avons réalisé cette expérimentation en entraînant les cinq clients sur leur dataset de 1000 données et faisons une validation sur l'ensemble des 5000 données. Alors nous avons obtenu un résultat de 72% ce qui n'est pas du tout satisfaisant.

Par la suite, nous avons continué les tests. Nous avons cherché à savoir quelle serait l'accuracy des deux modèles entraîné sur les 5000 premières données de MNIST et valider sur les 5000 suivantes. Ce résultat serait beaucoup plus représentatif car les deux modèles ne vont pas valider sur des données sur lesquelles elles se sont déjà entraînées. Pour le federated learning nous avons obtenu les résultats suivants.

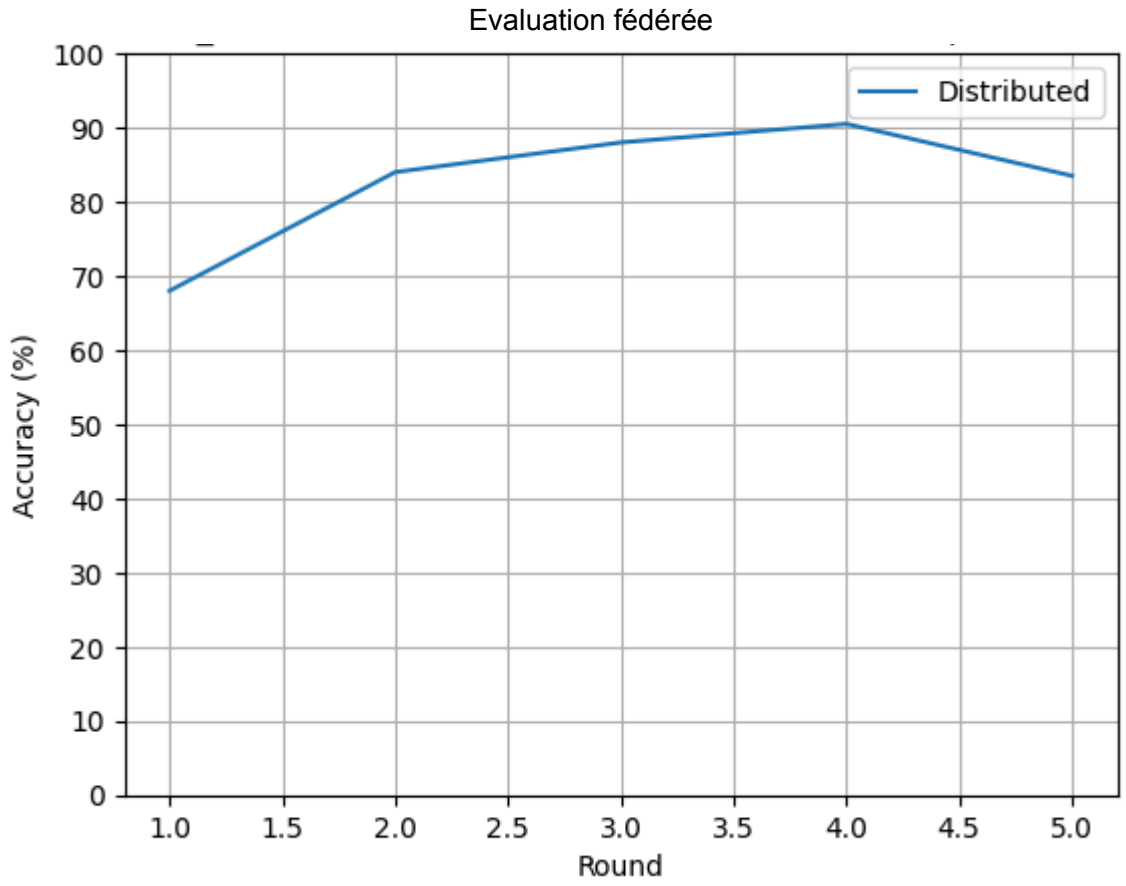


Figure 18 - Graphique montrant l'évolution de l'accuracy en fonction des rounds dans le cas de la validation fédérée sur un nouveau jeu de données

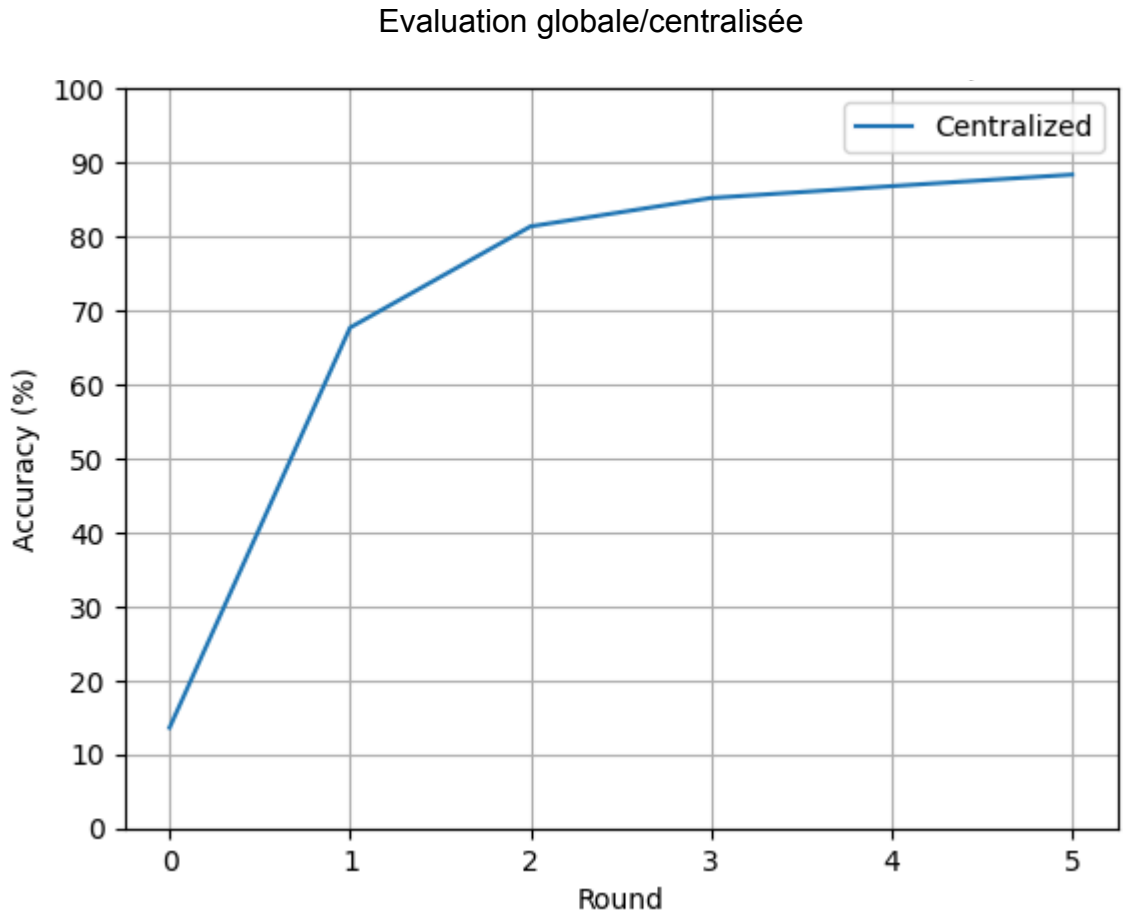


Figure 19 - Graphique montrant l'évolution de l'accuracy en fonction des rounds dans le cas de la validation sur l'ensemble d'un nouveau jeu de données

Pour l'évaluation globale/centralisée, nous avons obtenu 88% d'accuracy. Mais qu'en est-il du modèle entraîner sans federated learning. Nous avons obtenu 88% pour ce modèle ce qui démontre bien que le Federated learning est une solution fiable. Si nous avons obtenu des résultats différents lors de la première expérimentation, cela est dû au fait que lors d'un apprentissage classique, le modèle est plus performant sur des données qu'il a déjà vues contrairement au federated learning qui effectue un calcul sur les poids ce qui peut modifier l'information.

Pour aller encore plus loin, nous avons essayé de voir jusqu'où le federated learning peut aller. Nous avons repris exactement la même situation sauf que nous avons légèrement changé la base de données pour les clients. Étant donné qu'il y a dix labels (nombre de zéro à neuf), nous avons décidé de mettre dans chaque client uniquement deux labels. Pour faire simple, le premier client à 1000 données contenant uniquement des images de zéro et de un, le deuxième client a aussi 1000 données mais cette fois-ci contenant uniquement des images de deux et de trois. Ainsi de suite jusqu'au cinquième client qui contient uniquement des images de huit et de neuf. Notre but ici est d'essayer de montrer que le federated learning peut créer un modèle robuste si chacun des clients qu'il possède ont des données avec

des labels distincts. Dans un premier temps nous avons validé chacun des clients sur l'ensemble de la base de données ce qui a donné sans surprise 20% d'accuracy (parce que chaque client s'est entraîné que sur deux labels). Finalement comme vous pouvez le voir sur cette courbe, les résultats avec le federated learning n'ont pas été si très satisfaisants.

Evaluation globale/centralisée

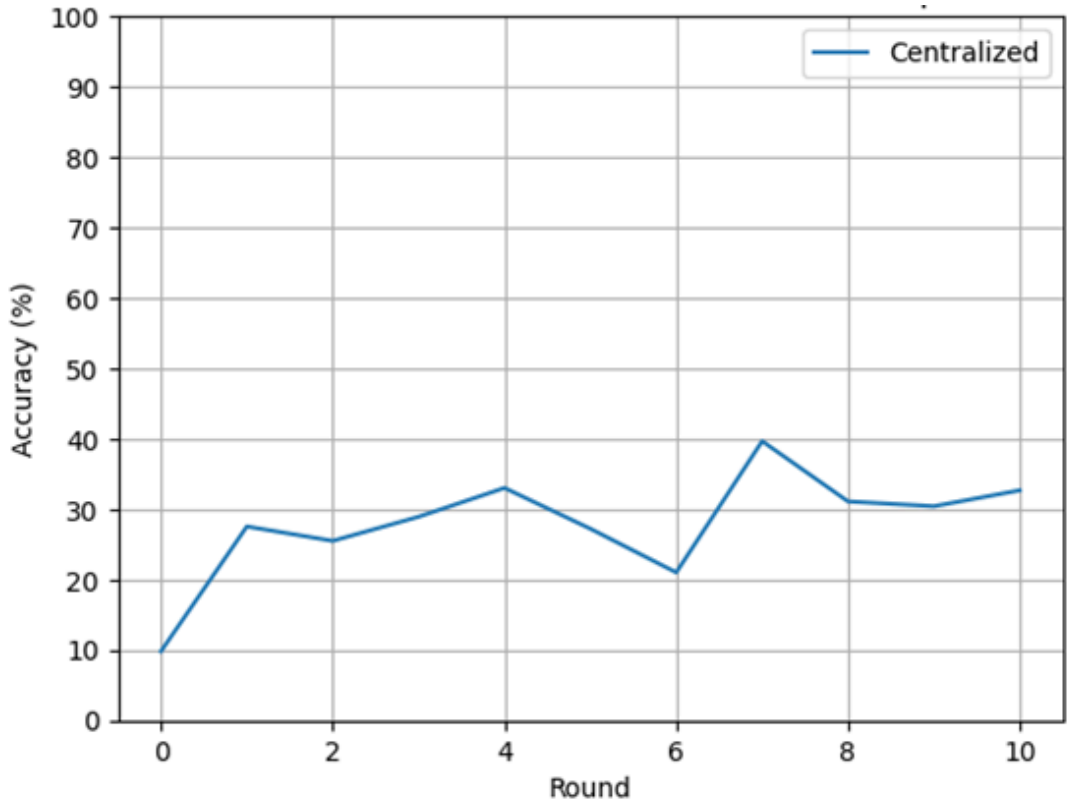


Figure 20 - Graphique montrant l'évolution de l'accuracy en fonction des rounds dans le cas où chaque client ne comporte que deux labels

La trop grande disparité des données sur chaque client doit générer des poids dont les valeurs sont très éloignées. Peut-être qu'avec plus de rounds l'accuracy aurait fini par converger vers des valeurs satisfaisantes.

On peut donc en conclure que l'apprentissage classique reste satisfaisant mais que l'apprentissage fédéré est une nouvelle approche qui permet de résoudre nombre de problèmes telles que la confidentialité de la data, le nombre de données, la puissance de calcul et bien d'autres. Choisir entre un apprentissage classique ou fédéré est à déterminer en fonction des besoins et des exigences des clients.

Un lien vers notre production réalisé :

https://colab.research.google.com/drive/1WBwqDztBBsBW6kgCnNhenMVLOa5dD1KI?usp=sharing

V - Bibliographie

Documentation

- Exemple d'utilisation de Flower avec Tensorflow : https://github.com/adap/flower/blob/main/examples/simulation-tensorflow/sim.ipynb https://github.com/adap/flower/tree/main/examples/advanced-tensorflow
- Stratégie FedAvg Flower : https://flower.ai/docs/framework/ref-api/flwr.server.strategy.FedAvg.html#fedavg
- Modèle de deep learning pour l'extraction de donnée PDF : https://github.com/microsoft/table-transformer?tab=readme-ov-file
- Transfer-learning avec Keras : https://github.com/yoavz/transfer-learning-keras/tree/master
- EasyOCR : https://github.com/JaidedAI/EasyOCR/blob/master/custom_model.md
- Documentation complète d'EasyOCR : https://jaided.ai/easyocr/modelhub/
- Classification d'images a l'aide de VGG16 : https://www.kaggle.com/code/viratkothari/image-classification-of-mnist-using-vgg16
- Template pour la création du site web : https://www.free-css.com/free-css-templates/page296/little-fashion

Article

- Comparatifs sur les différentes bibliothèques de federated learning : https://medium.com/elca-it/flower-pysyft-co-federated-learning-frameworks-in-python-b1a8eda68b0d

- Explication sur le deep learning : https://nanonets.com/blog/table-extraction-deep-learning/
- Federated learning pour la classification de texte : https://www.scitepress.org/Papers/2023/116587/116587.pdf
- Cas d'applications du federated learning : https://www.mdpi.com/2079-9292/11/4/670
- Extraction de donnée à partir d'un tableau : https://arxiv.org/pdf/2110.00061.pdf
- Guide pour mettre en place un OCR : https://medium.com/@adityamahajan.work/easyocr-a-comprehensive-guide-5ff1cb850168

- Explications du fine-tune : https://pub.towardsai.net/how-to-fine-tune-the-craft-model-in-easyocr-f9fa0ac5cc9d#d85d
- Comparatifs sur les frameworks : https://geekflare.com/pytorch-vs-tensorflow/
- Réseau de neurones : https://www.lebigdata.fr/reseau-de-neurones-artificiels-definition
- Federated Optimization strategy https://arxiv.org/abs/1812.06127
- Federated Averaging strategy https://arxiv.org/abs/1602.05629
- Neurones biologiques https://svtdiderot.fr/cordewener/4eme-cordewener/le-systeme-nerveux/

Vidéos

- Explication sur le transfer learning : https://youtu.be/deWrUO4vRxQ?si=y2Rui-ITRor5UFk7
- Animation sur le federated learning : https://www.youtube.com/watch?v=bGuG57PTZyw&ab_channel=650AILab
- Explication sur le deep learning https://www.youtube.com/@MachineLearnia

Autre

- Base de donnée de documents PDF : https://github.com/tpn/pdfs/tree/master
- OCR pour la reconnaissance de tableaux : https://github.com/tesseract-ocr/tesseract/issues/3150
- Solution d'OCR existante : https://github.com/JaidedAI/EasyOCR/issues/317

VI - Glossaire

Outils et plateformes de développement

Visual Studio Code : Visual Studio Code est un environnement de développement intégré (IDE) léger et extensible développé par Microsoft. Il offre des fonctionnalités avancées pour l'édition de code, le débogage, la gestion de version et l'intégration avec une variété d'outils et de langages de programmation.

IDE : IDE est l'acronyme de "Environnement de Développement Intégré". Il s'agit d'un logiciel qui fournit des outils et des fonctionnalités pour faciliter le développement de logiciels, notamment l'édition de code, le débogage, la gestion de version et parfois même le déploiement.

Slack : Slack est une plateforme de communication collaborative en ligne largement utilisée dans les milieux professionnels. Elle permet la communication en temps réel au sein des équipes via des messages textuels, des appels vocaux et vidéo, ainsi que la gestion de projets à travers des canaux thématiques et des intégrations avec d'autres outils de productivité.

Discord : Discord est une plateforme de communication en ligne initialement conçue pour les joueurs, mais qui est également utilisée dans divers contextes, y compris les communautés en ligne et les équipes de travail. Elle permet la communication via des messages textuels, des appels vocaux et vidéo, ainsi que la création de serveurs personnalisés avec des canaux thématiques.

Github : GitHub est une plateforme de développement collaboratif de logiciels basée sur Git. Elle permet aux développeurs de partager et de collaborer sur des projets en utilisant des fonctionnalités telles que le contrôle de version, le suivi des problèmes, la gestion des tâches et les demandes de tirage (pull requests).

OpenMined : OpenMined est une communauté open source qui se concentre sur le développement de technologies de confidentialité et de sécurité pour l'apprentissage fédéré et le calcul sécurisé en utilisant des techniques telles que le chiffrement homomorphique et le chiffrement multi-parties. Cette communauté vise à rendre l'intelligence artificielle plus respectueuse de la vie privée en permettant l'entraînement de modèles sur des données décentralisées sans compromettre la confidentialité des données individuelles.

Open source : "Open source" fait référence à un modèle de développement de logiciels dans lequel le code source est rendu accessible au public, permettant à quiconque de le consulter, de le modifier et de le distribuer. Les projets open source encouragent la collaboration, la transparence et l'innovation collective en permettant à un large éventail de contributeurs de participer au développement du logiciel.

Framework : Un framework est une structure logicielle préconçue qui fournit des outils, des bibliothèques et des conventions pour faciliter le développement d'applications. Les frameworks permettent aux développeurs de gagner du temps en évitant de réinventer la roue pour des tâches courantes, en fournissant une architecture de base et des fonctionnalités prêtes à l'emploi. Ils sont souvent conçus pour un domaine spécifique, comme le développement web, mobile ou l'apprentissage machine, et peuvent être utilisés pour accélérer le processus de développement et assurer une cohérence dans le code.

Apprentissage automatique et réseaux de neurones

OCR : OCR est l'acronyme de "Optical Character Recognition" (Reconnaissance Optique de Caractères). C'est une technologie qui permet de convertir des images ou des documents numérisés contenant du texte en texte éditable et recherchable. L'OCR identifie les caractères dans une image et les convertit en texte brut, ce qui permet aux utilisateurs de traiter et d'analyser plus facilement des documents numérisés, tels que des factures, des formulaires ou des livres.

Transfer learning : Le transfert d'apprentissage (ou "transfer learning") est une technique en apprentissage automatique où un modèle pré-entraîné sur une tâche est réutilisé comme point de départ pour une tâche similaire ou connexe. Plutôt que de construire un modèle à partir de zéro, le transfer learning exploite les connaissances déjà acquises par le modèle pré-entraîné, ce qui peut accélérer le processus d'entraînement et améliorer les performances du modèle, en particulier lorsque les données d'entraînement sont limitées.

CART : CART est l'acronyme de "Classification And Regression Trees" (arbres de classification et de régression). Il s'agit d'une technique d'apprentissage automatique utilisée pour construire des modèles prédictifs en forme d'arbres de décision. Les arbres de décision divisent récursivement l'espace des caractéristiques en sous-ensembles homogènes, en utilisant des règles simples basées sur les caractéristiques des données. Ils sont utilisés à la fois pour la classification, où ils prédisent la classe d'un échantillon, et pour la régression, où ils prédisent une valeur numérique.

Cortex visuel : Le cortex visuel est la partie du cerveau impliquée dans le traitement visuel, où les informations visuelles provenant des yeux sont traitées et interprétées.

Label : Un label est une étiquette ou une catégorie attribuée à un exemple de données dans le cadre de l'apprentissage supervisé. Il représente la réponse attendue du modèle pour cet exemple, souvent exprimée sous forme de texte ou de valeur numérique. Dans la classification, chaque exemple de données est associé à un label correspondant à sa classe, tandis que dans la régression, le label représente la valeur cible à prédire.

RNN : Un réseau neuronal récurrent (RNN) est un type de réseau neuronal artificiel conçu pour traiter des données séquentielles en tenant compte de la dépendance temporelle entre les différentes étapes de la séquence. Contrairement aux réseaux de neurones classiques, les RNN possèdent des boucles internes qui leur permettent de conserver une mémoire à court terme, ce qui les rend particulièrement efficaces pour traiter des données séquentielles telles que du texte, de la parole ou des séries temporelles.

Back-propagation : Il s'agit d'un algorithme utilisé pour entraîner les réseaux de neurones en calculant les gradients de l'erreur par rapport aux poids du réseau, ce qui permet d'ajuster ces poids pour minimiser l'erreur de prédiction.

Gradient : Le gradient représente la direction et le taux de variation maximale d'une fonction par rapport à ses paramètres. En apprentissage automatique, il est utilisé pour déterminer comment ajuster les paramètres d'un modèle afin de minimiser une fonction de perte.

Descente de gradient : La descente de gradient est un algorithme d'optimisation utilisé pour minimiser une fonction en ajustant itérativement les paramètres dans la direction opposée du gradient de la fonction par rapport à ces paramètres, avec un taux d'apprentissage spécifié.

Perceptron : Le perceptron est un modèle de neurone artificiel simple, souvent utilisé pour la classification binaire. Il prend des entrées pondérées, les somme, puis applique une fonction d'activation pour produire une sortie.

Synapse : Une synapse est une connexion fonctionnelle entre deux neurones qui permet le passage d'informations sous forme de signaux électriques ou chimiques. Elle facilite la transmission de l'influx nerveux d'un neurone à un autre dans le système nerveux.

Dendrite : Les dendrites sont de fines extensions branchues situées sur le corps cellulaire d'un neurone. Elles reçoivent des signaux électriques et chimiques provenant d'autres neurones et les transmettent au corps cellulaire pour traitement.

Axone : L'axone est une longue extension du neurone qui transmet les signaux électriques, appelés influx nerveux, depuis le corps cellulaire vers d'autres neurones ou vers des cellules cibles telles que les muscles ou les glandes.

Couche cachée : Une couche cachée dans un réseau de neurones artificiels est une couche qui ne correspond ni aux données d'entrée ni aux prédictions de sortie. Elle effectue des transformations non linéaires sur les entrées, permettant au réseau d'apprendre des représentations plus abstraites et complexes des données.

Gestion des données

Base de données : Une base de données est une collection organisée de données structurées ou non structurées, stockées électroniquement dans un système informatique. Elle est conçue pour permettre la récupération, la modification et la gestion efficace des données en fonction des besoins de l'utilisateur.

Big data : Big data fait référence à des ensembles de données extrêmement volumineux, variés et complexes, qui dépassent les capacités des outils de gestion de données traditionnels pour les stocker, les gérer et les analyser. L'analyse des big data nécessite souvent des techniques et des technologies spéciales pour extraire des informations significatives et des tendances à partir de ces données massives.

GED : La gestion électronique des documents (GED) désigne le processus de création, de gestion, de stockage et de partage de documents électroniques au sein d'une organisation. Elle implique l'utilisation de logiciels et de systèmes pour capturer, indexer, organiser et récupérer les documents de manière efficace et sécurisée.

Nombre binaire : Un système de numération utilisant seulement les chiffres 0 et 1 pour représenter les nombres.

Concepts et méthodologie en intelligence artificielle

IA : L'intelligence artificielle (IA) désigne le développement de systèmes informatiques capables d'effectuer des tâches qui nécessitent normalement l'intelligence humaine, telles que la reconnaissance de la parole, la prise de décision, l'apprentissage et la compréhension du langage naturel.

Data-scientist : Un data scientist est un professionnel qui analyse et interprète des ensembles de données complexes pour en extraire des insights et des tendances, généralement dans le but d'aider les entreprises à prendre des décisions stratégiques basées sur les données.

Orienté objet : Orienté objet est un paradigme de programmation où les programmes sont structurés autour d'objets qui représentent des entités réelles ou conceptuelles, et qui peuvent interagir entre eux en passant des messages.

Accuracy : L'accuracy, ou précision en français, est une mesure d'évaluation commune utilisée dans l'apprentissage automatique pour évaluer la performance d'un modèle de classification. Elle représente le nombre de prédictions correctes divisé par le nombre total d'échantillons.

Algorithme de marge maximum : L'algorithme de marge maximum (Max-Margin Algorithm) est une méthode d'apprentissage supervisé utilisée pour la classification, notamment dans le cadre des machines à vecteurs de support (SVM). Il vise à trouver l'hyperplan qui sépare les données avec la plus grande marge possible entre les classes, ce qui permet une meilleure généralisation et résistance au bruit.

Techniques de machine learning distribué

Fed average : Fed Average est un algorithme d'agrégation utilisé dans l'apprentissage fédéré (federated learning) où les modèles locaux des appareils participants sont moyennés pour former un modèle global. Cela permet de préserver la confidentialité des données tout en bénéficiant de la capacité de généralisation du modèle global.

Fed prox : FedProx est un algorithme utilisé dans l'apprentissage fédéré pour atténuer le biais des appareils locaux en introduisant une régularisation proximale dans la fonction de perte globale. Cela favorise une meilleure convergence vers une solution plus équitable tout en préservant la confidentialité des données locales.

Confidentialité

RGPD : Le RGPD, ou Règlement Général sur la Protection des Données, est une loi de l'Union européenne qui vise à protéger les données personnelles des individus en imposant des règles strictes sur leur collecte, leur traitement et leur stockage par les entreprises et les organisations. Il accorde également aux individus des droits sur leurs propres données, tels que le droit d'accès, de rectification et de suppression.