Travail de lecture et de rédaction scientifique sur le Federate Learning

Bal Sébastien

23 mai 2021

Remerciements

Table des matières

CHAPITRE 1

Machine Learning

1.1 En quoi consiste le Machine Learning

Le Machine Learning est présent partout sur la toile, cela va au moteur de recherche comme Google. Nos applications tels que Siri et Alexa. Les fil d'actualités comme Facebook et Twitter. Toutes ces platformes stockent des données sur les utilisateurs afin de comprendre et d'améliorer leurs performances. Ces données serviront à mieux cibler ce que les utilisateurs aiment. La machine pourra ainsi proposer plus facilement des recommandations ou des résultats pour des recherches.

Le Machine Learning fait partie d'une branche de l'intéligence artificielle (IA).Celle ci est utilisée dans l'informatique, elle permet d'enseigner à des machines comment apprendre et agir sans nécissiter une programmation élaboré. De plus, le Machine Learning aborde le concept d'analyse des données qui utilise la construction et l'adaptation de modèles. Ainsi un ordinateur peut "apprendre" par l'expérience qu'il a acquise ce qui semble le plus proche du modèle passé en paramètre. La présence d'algorithmes est nécessaire pour améliorer la capacité de prédiction d'un modèle suivant certains paramètres.

Le Machine Learning peut etre différencié par deux catégories d'algorithmes : supervisés et non supervisés.

L'apprentissage supervisé utilise des données qui sont déjà connues par le model avec une étiquette. A la fin de son entrainement,le model pourra être

capable de retrouver des données dans le même domaine dont les données n'ont pas d'étiquette.

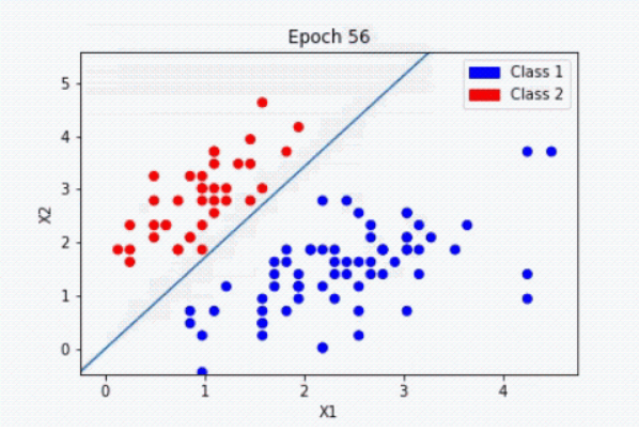

FIGURE 1.1: Modèle supervisé

Sur cette image ci dessus, on peut constater que l'algorithme ajustera sont modèles en fonction de ses paramètres afin de diminuer l'écart entre les résultats attendus et les résultats obtenus. Ce qui diminuera au fil de l'entraînement les marges d'erreurs.

L'apprentissage non supervisé, quand à lui, permet d'entrainer le modèle sans étiquette. La machine cherche parmis les données sans indices et permet de découvrir les tendances.L'apprentissage par renforcement, ce modèle permet d'entrainer la machine avec un objectif bien précis.Le modèle a un système d'échecs et d'erreurs.

1.2 En quoi consiste le Machine Learning

Le Machine Learning est présent partout sur la toile, cela va au moteur de recherche comme Google. Nos applications tels que Siri et Alexa. Les fils d'actualités comme Facebook et Twitter. Toutes ces platformes stockent des données sur les utilisateurs afin de comprendre et d'améliorer leurs performances. Ces données serviront à mieux cibler ce que les utilisateurs aiment. La machine pourra ainsi proposer plus facilement des recommandations ou des résultats pour des recherches.

1.3 Principes des concepts du Machine Learning

1.4 Big Data et Machine Learning

Avec un grand nombre de données les outils analytiques ne savent pas traiter autant de donnés pour l'exploiter à bon escient. Un volume de données trop large empèche une analyse compréhensive. Les corrélations et les relations entre ces données sont trop importantes ce qui complique la tache pour des analystes.

C'est pour cette raison que le Machine Learning est idéal pour du Big Data. Cette technologie pourra extraire des valeurs qui proviennent de cette source de données sans passer par l'intermèdiaire d'un être humain.

Sans le Big Data, le Machine learning et l'intelligence artificielle ne sont rien. Les données sont le moyen pour le Machine Learning d'apprendre et de comprendre comment pensent les humains.

La technologie est capable d'apprendre en allant chercher elle même des ensembles de données pour les analyser et devenir plus précise.

CHAPITRE 2

Deep Learning

2.1 Le concept du Deep Learning

Le Deeo Learning est basé sur un système d'un réseau neuronal inspiré des systèmes cérébraux. Ce type d'apprentissage est d'y supervisé car c'est le développeur qui va décidé sur quel type d'apprentissage il va lancer le Deep Learning. Cette technique à besoin d'énormément de données, on parlera donc de Data Lake.

Pour que le modèle mathématique deviennent performant, il faudra l'entrainer à reconnaitre un élément en particulier. Prenons le cas de la reconnaissance d'un animal. Pour la phase d'apprentissage nous passerons au système plusieurs images d'animaux. On précisera dans la partie d'entrainement les éléments auquels le système devra être conscient.

2.2 L'entraînement du Deep Learning

Pour notre exemple, les boules vertes représente le bon chemin que le système va prendre pour arriver à vérifier le modèle qui était demandé. Les boules bleus sont celles qui ont des caractéristiques avec le modèle mais ne correspondra pas exactement au modèle qui était demandé. Les boules rouges quand à elles, représentent les erreurs que le système a exclu pour pouvoir apprendre le modèle exacte. Les erreurs sont par la suite renvoyées en amont du système pour que le système ajuste son modèle mathématique

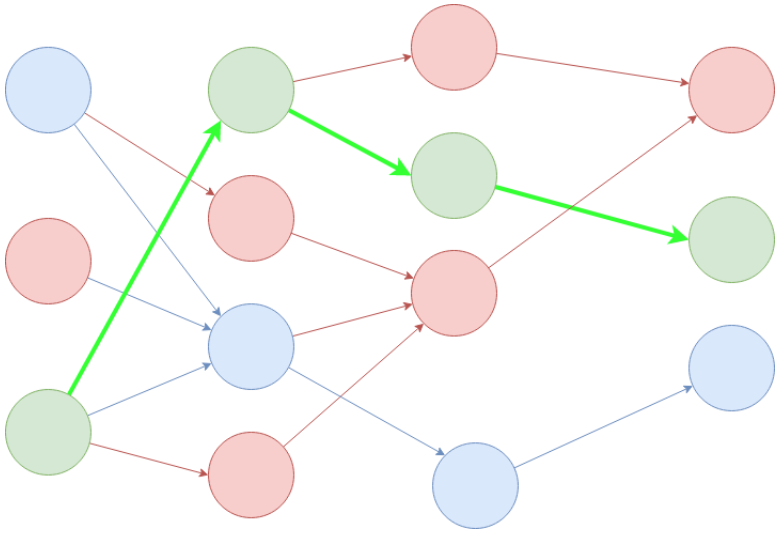

FIGURE 2.1: Autoapprentissage Deep Learning

CHAPITRE 3

Federate Learning

3.1 Définition du Federate Learning

En quelques mots, c'est un apprentissage automatique distribué qui permet d'entraîner un modèle mathématique avec un large groupe de données décentralisées qui se trouvent sur des téléphones portables(pour notre cas ici).

3.2 Algorithme de Federate Learning

3.3 Implémentation

Dans notre modèle, nous allons utilisé le système TensorFlow pour former notre système neuronal. TensorFlow est une bibliothèque open source pour le Machine Learning. C'est un petit couteau Suisse qui contient ici des outils pour permettre de résoudre des problèmes mathématiques.

3.4 Deep Learning, cas d'utilisation

3.4.1 Smart Building

3.4.2 Industrie

Bibliographie

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

[2] Pour la partie sur le Machine Learning https ://www.lebigdata.fr/machine-learning-et-big-data

ANNEXE A

Annexe

TOWARDS FEDERATED LEARNING AT SCALE: SYSTEM DESIGN

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

ABSTRACT

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

1 INTRODUCTION

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

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

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

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

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

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

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