

Rapport : A Survey on Distributed Machine Learning

AKAMBA MANI Crescence Catherine

November 29, 2023

1 Techniques d'augmentation de stockage

L'apprentissage automatique s'intéresse de plus en plus aux applications complexes. Ce qui entraîne un ensemble d'opérations et de calculs complexes donc une grande charge de travail. Il est donc primordial d'accélérer la charge de travail:

Il existe deux façons différentes et complémentaires d'accélération des charges de travail : ajout de plusieurs ressources dans une seule machine (vertical scaling ou scaling up) et ajout de plusieurs noeuds au système (horizontal scaling ou scaling out)

1. Scaling up

La méthode la plus commune dans les solutions scale-up est l'ajout des GPUs programmables. Les GPUs comportent un grand nombre de threads matériels. Par exemple, les Nvidia Titan V Tesla V100 ont un total de 5120 coeurs, ce qui les rend 47 fois plus rapides pour l'apprentissage profond qu'un serveur classique (à savoir un Intel Xeon E5-2690v4). La prolifération des GPGPU (General-Purpose GPU, c'est-à-dire des GPUs capables d'exécuter du code arbitraire) a conduit les éditeurs à concevoir des produits personnalisés qui peuvent être ajoutés aux machines conventionnelles comme accélérateurs et ne remplissent plus aucun rôle dans le sous-système graphique de la machine. Par exemple, la série de GPU Nvidia Tesla est destinée au calcul hautement parallèle et conçue pour être déployée dans des supercalculateurs et clusters.

Comme autre accélérateur, nous avons les circuits intégrés spécifiques à une application (Application Specific Integrated Circuits ASICs), qui implémentent les fonctions spécialisées grâce à une conception hautement optimisée. Nous pouvons également citer l'Epiphany qui est un processeur à usage spécial conçu avec une architecture MIMD qui utilise un ensemble de processeurs, chacun accédant à la même mémoire, pour accélérer l'exécution de l'opération à virgule flottante.

2. Scaling out

Augmentation des performances par ajout de plus d'équipements

2 Architecture pour l'apprentissage automatique distribué

L'architecture automatique distribuée est composée de trois couches dépendantes: apprentissage automatique , parallelisme et la topologie

1. Apprentissage automatique

L'apprentissage automatique est une technologie qui apprend à prendre des décisions ou à faire de prédictions à partir de données. On distingue trois principaux d'apprentissage automatique à savoir : l'apprentissage supervisé l'apprentissage non supervisé l'apprentissage semi-supervisé l'apprentissage par renforcement

Il est constitué de deux phases fondamentales: la phase d'entraînement qui consiste à entraîner le modèle à partir d'un jeu d'entraînement et la phase de test qui consiste à évaluer de nouvelles observations.

2. Parallélisme

En matière de distribution, il existe deux manières fondamentales différentes de répartir le problème sur plusieurs machines:
la parallélisation des données et la parallélisation du modèle.

Dans l'approche parallélisation des données, les données sont partitionnées autant de fois qu'il y a des noeuds(worker-node) dans le système et chacun de ces noeuds applique le même algorithme sur leur ensemble de données respectifs(different). Le même modèle est disponible pour tous les noeuds de sorte qu'un seul résultat cohérent émerge naturellement.

Dans l'approche parallélisation du modèle, les copies exactes de l'intégralité de l'ensemble des données sont traitées par les noeuds. Chaque noeud applique une partie du modèle sur sa copie de données. Le modèle est donc l'ensemble de toutes les parties du modèle. Les résultats de tous les modèles sont alors regroupés à l'aide des méthodologies d'assemblage.

3. Topologie

La topologie représente la structure dans laquelle les noeuds sont organisés. Les systèmes centralisés (figure 3(a)) utilisent une approche strictement hiérarchique de l'agrégation, qui s'effectue en un seul point central. Les systèmes decentralisés permettent une aggregation intermédiaire, soit avec un modèle repliqué qui est constamment mis à jour lorsque l'aggregat est diffusé à tous les noeuds, comme des topologies aborescentes dans lesquelles chaque noeud ne doit communiquer qu'avec ses parents et ses enfants, soit avec un modèle partitionné qui est réparti sur plusieurs serveurs de paramètres (tous les paramètres du modèle sont stockés dans une réserve sur chaque serveur de paramètres) . Les systèmes entièrement distribués se composent d'un réseau de noeuds indépendants qui élaborent ensemble la solution et où aucun rôle spécifique n'est attribué à certains noeuds. Il s'agit d'une topologie Peer-to-peer, où chaque noeud possède sa propre

copie des parametres du modele et les noeuds communiquent directement entre eux.

Communication Lorsque l'apprentissage automatique distribue est utilise , l'objectif est d'obtenir la meilleur precision au cout de calcul et de communication le plus bas . Pour planifier et equilibrer la charge de travail, trois aspectss doivent etre pris en compte:

- Identifier les taches qui peuvent etre executees en parallele
- Decider de l'ordre d'execution des taches
- Assurer une repartition equilibre de la charge entre les noeuds disponibles

Une fois ces trois questions reglees, les informations entre noeuds doivent etre communes de la maniere la plus efficace possible. Il existe plusieurs techniques permettant d'entrelacer le calcul parallele et la communication.

- **Le modele BSP (Bulk Synchronous Parallel)** est le modele le plus simple dans lequel les programmes assurent la coherence en se synchronisant entre chaque phase de calcul et de communication. L'avantage est que les programmes de machine learning BSP serialisables sont garantis de produire une solution correcte. L'inconvenient est que les noeuds ayant termine doivent attendre a chaque synchronisation que tous les autres noeuds terminent. ce qui entraîne des surcharges si certains noeuds progressent plus lentement que d'autres.
- **Le modele SSP (Stale Synchronous Parallel)** relache la surcharge de synchronisation en permettant aux noeuds les plus rapides d'avancer pendant un certain nombre d'iterations. Si ce nombre est depasse, tous les noeuds sont mis en pause. L'avantage du SSP est qu'il offre toujours de solides garanties de convergence des modeles. Un inconvenient, est que lorsque la stagnation devient trop importante , les taux de convergence se deteriorent.
- **La methode ASP (Approximate Synchronous Parallel)** limite le degre d'imprecision d'un parametre.
- **Le modele BAP/TAP (Barrierless Asynchronous Parallel / Total Asynchronous Parallel)** permet aux noeuds de communiquer en parallele sans attendre l'un ou l'autre. L'inconvenient est que le modele peut converger lentement ou meme se developper de maniere incorrecte.

3 Quelques solutions d'apprentissage automatique distribués

- MapReduce : MapReduce est un framework (et une architecture sous-jacente) de traitement de donnees developpe par Google pour

traiter les données dans un environnement distribué.

- Apache Spark
- Caffe 2
- CNTK
- etc...