

Hands on Deep Learning

Martin Vielvoye - 2021

Deep Learning

en Pratique

ML/DL en local

- Les langages très populaires en machine learning (**Python** et **R**) sont des langages très faciles d'accès, à installer sur tous les OS et à maintenir.
- Malheureusement, plusieurs facteurs freinent l'entraînement de d'algorithmes de ML/DL en local :
 - Demande de mémoire forte selon les volumes de datasets
 - Certaines opérations de nettoyages ou d'entraînement sont très coûteux pour un CPU de machines "grand public"
 - La nature itérative de l'entraînement fait que chaque étape prends du temps, rendant l'itération et l'expérience frustrante mais possible

ML/DL dans le cloud

- De très nombreux services proposent aujourd'hui de délocaliser les calculs dits "coûteux" vers les datacenters.
- Plusieurs avantages :
 - Accessibilité et accompagnement
 - Un ratio coût-efficacité rentable
 - Une persistance de données
 - Une forte scalabilité/extensibilité
 - Un hardware state-of-the-art qu'il ne faut pas maintenir
 - Accès à du hardware d'accélération (GPU, TPU, Ordinateurs Quantique)

ML/DL dans le cloud

- **Cloud computing** (bas niveau):

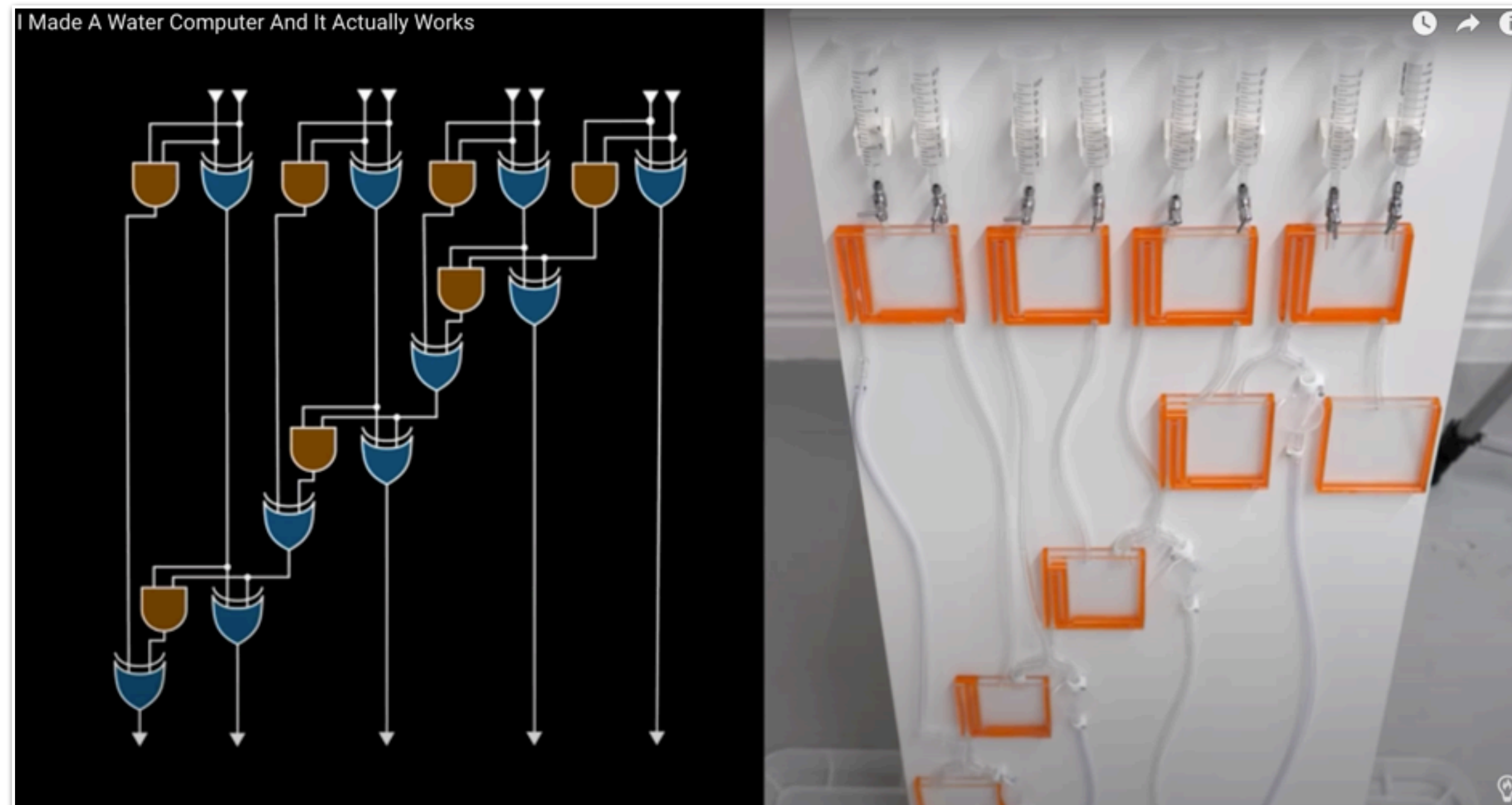
- Amazon Web Service (*AWS*)
- Azure Microsoft
- IBM Cloud
- Google Cloud Platform (*GCP*)
- Apache Hadoop

- **Services MLOps** (haut niveau):

- AWS AI Service
- Azure AI
- IBM Watson
- Vertex AI
- Apache Spark

Un hardware adapté : Théorie

- Le type de hardware qui compose une machine, influe énormément sur les performances de ses calculs.
- Dès lors qu'on peut construire les portes logiques universelles (**NAND** et **NOR**), quel que soit le procédé, on peut construire un ordinateur.



Title : "I Made a Water Computer And it Actually Works"

Youtube Channel: Steve Mould

Un vrai challenge d'un point de vue ingénieur/mécanique mais le résultat est extrêmement lent à calculer!

Mais est-ce que l'ordinateur carbone/silicium est la solution la plus rapide de calcul?

"Nature is quantum, goddamn it! So if we want to simulate it, we need a quantum computer."

Richard Feynman

Un hardware adapté : Pratique

- Sur-simplification de CPU vs GPU vs TPU
 - **CPU** : Calcule très vite mais de manière linéaires (une opération à la fois).
 - **GPU** : Calcule moins vite mais de manière parallèle (énormément de petites opérations à la fois).
 - **TPU** :
 - Similaire au GPU mais réduit la taille de chaque valeur en 8-bit réduisant la mémoire nécessaire et le temps de calcul
 - Accélération de calcul et d'efficacité énergétique pour les calculs de tenseurs (multiplication de matrice)

Deep Learning

en Frameworks

ML Frameworks

Tensorflow

- Haut niveau et Bas niveau
- Fait pour Python, mais beaucoup de support pour d'autres langages
- Très bon support d'images, de tableaux et de modèles de DL
- Accélération GPU et TPU
- Optimisé pour Google Collab

- Pas forcément beginner-friendly sans Keras
- Permet d'expérimenter avec des modèles très bas niveau

ML Frameworks

PyTorch

- Haut niveau et Bas niveau
- Interface en Python sur du C++
- Graph et modèles dynamique, permet de changer un modèle pendant son entraînement (et ne pas tout recommencer de zéro)
- Optimisé pour Google Collab
- Assez user-friendly, permet d'utiliser des debuggers
- Optimisé pour le prototypage, la production et le déploiement rapide

ML Frameworks

Keras

- Haut niveau
- Permet d'être implémenté sur plusieurs outils (Tensorflow, Theano, Microsoft Cognitive Toolkit, ...)
- Permet d'accélérer l'exploration et la parallélisation de calculs
- L'un des frameworks les plus beginner-friendly
- A plus de mal sur les opérations bas niveau et les modèles très profonds

ML Frameworks

Theano

- Bas niveau
- Librairie spécialisée dans les expressions mathématiques de matrices à plusieurs dimensions
- Possède une rapidité d'exécution proche de fonctions écrites en C
- Cette rapidité est un réel avantage dans les cas d'usage avec beaucoup de données

ML Frameworks

MLlib Spark

- Haut niveau et Bas niveau
- Librairie d'Apache Spark, un framework spécialisée dans la big data et le calcul distribué.
- Possède des fonctions utilitaires de hauts niveau facile d'accès ainsi que des optimisations bas niveau
- Optimisé pour une intégration fluide et scalable avec les outils d'Apache Spark qui sont très populaire dans les entreprises-tech.

ML Tools

Scikit Learn

- Haut niveau
- Construit sur plusieurs package comme NumPy, SciPy et matplotlib
- Complètement OpenSource et avec des licenses très libres
- Possède beaucoup de taches de ML (classification, clustering, régression, decision tree, etc)
- Très bonne documentation et une communauté très active

Deep Learning

en Démonstration