Présentation du TinyML

Mark Patrick (Mouser Electronics)

Dans cet article, nous allons nous pencher sur l'apprentissage automatique (ML en anglais pour « machine learning ») propulsé par des microcontrôleurs de faible puissance et à faible consommation d'énergie. Il s'agit d'une nouvelle approche de l'apprentissage automatique baptisée TinyML. L'apprentissage automatique a pénétré de nombreux aspects de notre vie quotidienne, tant chez soi qu'au travail ou entre les deux. De nombreuses applications d'apprentissage automatique nécessitent une puissance de calcul importante pour traiter des données scientifiques ou financières complexes. Or, les applications ML conçues pour l'Internet des objets (IoT) et d'autres applications en périphérie (edge) n'ont qu'une faible capacité de calcul et une faible connectivité.

L'IA et le ML font déjà partie de nos vies

Il n'y a pas si longtemps encore que parler à sa montre-bracelet ou à un gadget dans le style du Communicator de Star Trek relevait de l'esprit fécond des auteurs de science-fiction. C'est devenu aujourd'hui un geste presque banal. Nous parlons à nos applications pour smartphones, au système d'infodivertissement de notre voiture et à l'enceinte intelligente dans notre salon.

L'intelligence artificielle (IA) – la science qui vise à créer des ordinateurs capables de penser, de détecter, de reconnaître et de résoudre des problèmes – est devenue la pierre angulaire de l'informatique et de la science des données d'aujourd'hui. L'apprentissage automatique est un domaine d'application de l'IA et concerne l'utilisation par des ordinateurs d'algorithmes leur permettant d'apprendre et d'améliorer des méthodes pour réaliser une tâche sans qu'ils aient été explicitement programmés pour l'accomplir.

Le monde qui nous entoure porte déjà l'empreinte du ML : prévisions météorologiques, recherche d'itinéraires, encarts publicitaires dans vos applications de médias sociaux… De nombreux domaines de la recherche scientifique dépendent désormais du ML pour parcourir des pétaoctets de données afin d'en dégager des tendances.

Fonctionnement de l'apprentissage automatique

Des exemples d'apprentissage automatique que nous venons de citer, nous ne percevons en général que le résultat. Peu d'entre nous ont eu l'occasion d'admirer de près la complexité d'opérations telles que la recherche d'un nouveau trou noir ou le calcul du nombre de permutations d'événements météorologiques passés en vue pour déterminer les prévisions météorologiques du jour. De telles tâches complexes s'appuient sur de grands ensembles de données, plusieurs algorithmes candidats et une puissance de calcul importante. En creusant un peu le sujet, on découvre qu'il existe différentes catégories d'apprentissage automatique, chacune convenant plus particulièrement à des tâches spécifiques. Sans entrer dans les détails, il s'agit des méthodes d'apprentissage supervisé, non supervisé, semi-supervisé et par renforcement.

Le réseau de neurones est un élément essentiel de toute application de ML. Pour résumer grossièrement, un réseau neuronal dans un modèle mathématique imite la fonction des neurones dans le cerveau humain. Le modèle utilise des algorithmes pour déduire les probabilités d'un résultat. Par exemple, « la probabilité que cette image représente un chien est de 95% » en résultat d'une opération de reconnaissance d'image. Il existe différents types de réseaux de neurones. Des termes comme réseaux de neurones convolutifs (CNN) ou réseaux de neurones récurrents (RNN) ne vous sont peut-être pas inconnus. Chaque type de réseau neuronal possède un ensemble différent de couches interconnectées, ce qui le rend plus adapté à des tâches spécifiques.

L'apprentissage supervisé consiste à alimenter le réseau neuronal en données d'entraînement en vue de permettre à un algorithme d'en déduire des résultats. Prenons pour exemple la reconnaissance d'images, une tâche qui convient particulièrement aux CNN. Pour identifier différents types de fruits, il faut fournir à l'algorithme du réseau neuronal des milliers d'images étiquetées représentant différents types de fruits, sous différents angles et à différents degrés de maturité. L'algorithme recherche alors des

caractéristiques discernables qui l'aident à identifier les différents types de fruits. Cette phase d'entraînement est itérative et il est alors possible qu'il faille encore affiner l'algorithme afin d'obtenir les probabilités les plus élevées lorsqu'il est mis en présence d'un ensemble de données de test (ici, des images).
Une fois que l'algorithme du réseau neuronal a atteint son meilleur niveau de performance avec l'ensemble de données de test, le modèle est prêt à être déployé. Une fois entré en phase de déploiement, aussi appelée inférence, le modèle déduit les résultats à partir d'une probabilité.
Pour entrer en interaction avec, par exemple, notre enceinte intelligente, nous utilisons généralement un mot ou une phrase de déclenchement – « OK, Google » – pour que l'appareil sorte de veille et se mette à nous écouter. Une enceinte intelligente ne dispose évidemment pas des capacités de calcul d'un centre de données. Elle se contente d'enregistrer de courts fichiers audio et de les envoyer dans le cloud, où le processus d'inférence déterminera la nature de notre demande. Ce que l'enceinte fait d'elle-même, c'est détecter le mot ou la phrase de déclenchement. C'est un excellent exemple d'apprentissage automatique simple et c'est précisément à cela que sert le TinyML !

L'apprentissage automatique et l'IoT industriel

La liste des applications susceptibles de tirer avantage de l'apprentissage automatique s'allonge de façon exponentielle à mesure que cette technologie prolifère dans notre société. Cependant, de nombreuses applications industrielles n'ont rien à voir avec le « big data », mais s'occupent de la manière de rendre les lignes de production plus efficaces. Une panne inattendue sur une ligne de production peut s'avérer très coûteuse. Imaginez par exemple qu'un moteur tombe en panne durant la transformation d'aliments réfrigérés. La production serait mise à l'arrêt et les matières premières seraient perdues. C'est pourquoi de nombreuses entreprises ont adopté un régime de maintenance prédictive afin de planifier des temps d'arrêt prévisibles et ainsi éviter que de telles pannes surviennent. La surveillance basée sur l'état des moyens de production tels que les moteurs et les actionneurs aide à prévoir quand, par exemple, un roulement de moteur commence à montrer des signes d'usure excessive. Les algorithmes d'apprentissage automatique utilisés dans les capteurs périphériques IIoT peuvent identifier quand la vibration d'un moteur s'écarte de sa fréquence habituelle, ce qui peut être un signe annonciateur suffisant.
Rien que dans le domaine de l'industrie, les possibilités d'applications de l'apprentissage automatique sont innombrables, mais plusieurs défis techniques se dressent sur leur chemin. Contrairement aux applications big data, de simples applications comme les capteurs périphériques IIoT ne disposent que d'une infime partie des ressources de calcul et de mémoire disponibles dans un centre de données. Ensuite, une seule usine peut devoir être équipée de centaines de capteurs. L'économie d'échelle est donc un facteur à prendre en compte au même titre que les dimensions physiques de l'appareil, la disponibilité d'une alimentation électrique appropriée et le type de connectivité (filaire ou sans fil). Dans une majorité des cas aussi, l'algorithme de réseau neuronal associé à un capteur de vibrations, par exemple, requiert moins de ressources qu'un algorithme utilisé dans le cadre de la recherche dans l'espace lointain. Cela autorise donc les développeurs de systèmes embarqués à chercher le moyen d'exécuter des modèles de réseau sur des microcontrôleurs à faible consommation alimentés par batterie.

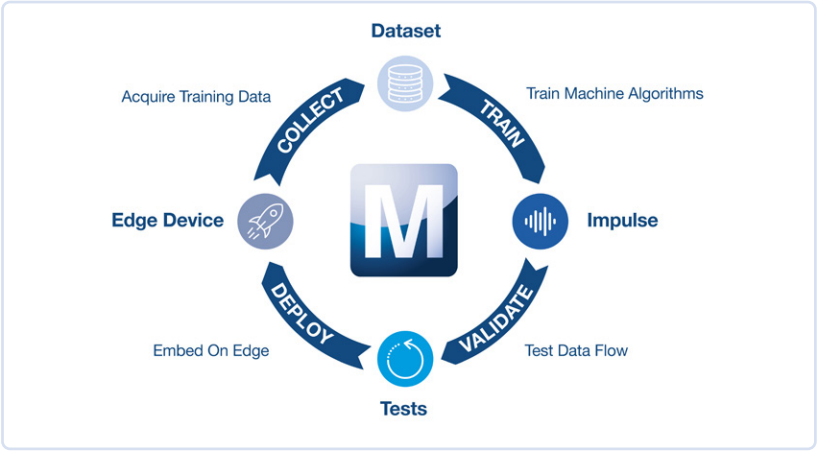

Figure 1: Étapes d'entraînement et d'inférence.

TinyML : l'apprentissage automatique en périphérie

La plateforme informatique utilisée pour entraîner un modèle de réseau neuronal ne doit pas nécessairement être la même que celle utilisée en phase de déploiement. On peut ainsi avoir recours à une plateforme moins gourmande en ressources, mais ce n'est là pas la moindre des difficultés techniques à prendre en compte. L'algorithme est-il capable de s'exécuter au bon moment ? Comment l'appareil communiquera-t-il avec le système hôte pour avertir le personnel opérationnel ? Enfin, les développeurs de systèmes embarqués ne sont généralement pas des spécialistes des systèmes de données et il leur faut, à eux aussi, un certain temps d'apprentissage pour maîtriser les concepts de l'apprentissage automatique et travailler avec des réseaux neuronaux.

230062-04

À propos de l'auteur

En tant que directeur marketing technique de Mouser Electronics pour la région EMEA, Mark Patrick est responsable de la création et de la diffusion du contenu technique - un contenu essentiel à la stratégie de Mouser visant à soutenir, informer et inspirer son public d'ingénieurs.
Avant de diriger l'équipe de marketing technique, Patrick faisait partie de l'équipe de marketing achat de la région EMEA et jouait un rôle essentiel dans l'établissement et le développement des relations avec les principaux partenaires et fournisseurs. En plus d'avoir occupé divers postes dans les départements techniques et marketing, Patrick a travaillé pendant huit ans chez Texas Instruments, dans les services support et ventes techniques.
Ingénieur expérimenté, passionné de synthétiseurs vintage et de motos, il n'hésite pas à les réparer. Patrick est titulaire d'un diplôme d'ingénieur en électronique avec mention très bien de l'université de Coventry.