



Plus grand n'est pas forcément mieux

Dimensionnement du matériel pour une ML/AI optimale à la périphérie

Par : Yann LeFaou est directeur associé de la division Touch and Gesture (TXFG) de Microchip.

L'apprentissage machine (ML) et l'intelligence artificielle (AI, dont le ML peut être considéré comme un sous-ensemble) ont toujours été mis en œuvre sur des plateformes informatiques à haute performance et, plus récemment, dans le nuage. Aujourd'hui, cependant, ces deux technologies sont de plus en plus déployées dans des applications où le traitement est effectué à proximité de la source de données. Idéal pour les appareils IoT, un tel traitement en périphérie signifie que moins de données doivent être envoyées vers le cloud pour être analysées. Parmi les avantages, citons l'amélioration des performances grâce à un temps de latence réduit et une meilleure sécurité.

La ML/AI fait passer le traitement de pointe au niveau supérieur en rendant possible l'inférence à la source. Cela permet à un appareil IoT, pour rester dans cet exemple, d'apprendre et de s'améliorer à partir de l'expérience. Les algorithmes analysent les données pour rechercher des modèles et prendre des décisions éclairées. La machine (ou l'IA) apprend de trois manières : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage renforcé.

Dans le cadre de l'apprentissage supervisé, la machine utilise des données étiquetées pour la formation. Par exemple, une caméra de sécurité intelligente peut être formée à l'aide de photos et de clips vidéo de personnes debout, marchant, courant ou transportant des boîtes. Les algorithmes d'apprentissage supervisé comprennent la régression logistique et Naïve Bayes, avec un retour d'information nécessaire pour continuer à affiner le(s) modèle(s) sur la base duquel (desquels) les prédictions seront faites.

L'apprentissage non supervisé utilise des données non étiquetées et des algorithmes tels que le K Means Clustering et l'analyse des composantes principales pour identifier des modèles. Cette méthode est idéale pour la détection des anomalies. Par exemple, dans un scénario de maintenance prédictive ou dans une application d'imagerie médicale, la machine signalerait les situations ou les aspects de l'image qui sont inhabituels, sur la base du modèle "habituel" que la machine a construit et qu'elle maintient.

L'apprentissage renforcé est un processus d'essai et d'erreur. Comme pour l'apprentissage supervisé, un retour d'information est nécessaire mais, au lieu de simplement corriger la machine, le retour d'information est traité comme une récompense ou une pénalité. Les algorithmes comprennent Monte Carlo et l'apprentissage Q.



Dans les exemples ci-dessus, l'élément commun est la vision intégrée, rendue "intelligente" par l'ajout de ML/AI et toute autre application peut bénéficier d'une inférence à la source basée sur la vision. En outre, la vision intégrée intelligente peut utiliser des parties du spectre non visibles par l'œil humain, telles que l'infrarouge (utilisé pour l'imagerie thermique) et les UV.

Fournissez à un système ML/AI edge d'autres données, telles que des relevés de température et des niveaux de vibration provenant de transducteurs, et les dispositifs IoT industriels peuvent non seulement jouer un rôle important dans la stratégie de maintenance prédictive d'une organisation, mais ils peuvent également fournir des alertes précoces en cas de défaillance inattendue et ainsi contribuer à la protection des machines, des produits et du personnel.

Systèmes embarqués

Comme nous l'avons mentionné dans notre introduction, la ML/AI nécessitait initialement des ressources informatiques considérables. Aujourd'hui, en fonction de la complexité de la machine, des composants davantage associés aux systèmes embarqués (tels que les appareils IoT) peuvent être utilisés pour mettre en œuvre la ML et l'IA.

Par exemple, la détection et la classification d'images peuvent être mises en œuvre dans des réseaux de portes programmables (FPGA) et sur des unités de microprocesseurs (MPU). En outre, des applications relativement simples, telles que la surveillance et l'analyse des vibrations (pour la maintenance prédictive, par exemple), peuvent être mises en œuvre dans une unité de microcontrôleur (MCU) de 8 bits.

En outre, alors que la ML/AI nécessitait initialement des scientifiques des données hautement qualifiés pour développer des algorithmes de reconnaissance des formes - et des modèles pouvant être automatiquement mis à jour afin de pouvoir faire des prédictions - ce n'est plus le cas aujourd'hui. Les ingénieurs en systèmes embarqués, déjà familiarisés avec le traitement de pointe, disposent désormais du matériel, des logiciels, des outils et des méthodologies nécessaires pour concevoir des produits basés sur la ML/AI. En outre, de nombreux modèles et données pour l'entraînement des machines sont disponibles gratuitement et de nombreux fournisseurs de circuits intégrés proposent des environnements de développement intégrés (IDE) et des suites de développement pour accélérer la création d'applications de ML/AI.

Prenons, par exemple, l'IDE MPLAB X de Microchip. Ce logiciel intègre des outils qui aident les ingénieurs à découvrir, configurer, développer, déboguer et qualifier des conceptions intégrées pour de nombreux appareils de l'entreprise. Un plug-in de la suite de développement de l'apprentissage automatique permet aux modèles d'apprentissage automatique d'être flashés directement sur le matériel cible. Cette suite utilise ce que l'on appelle l'apprentissage automatique (AutoML), qui est le processus d'automatisation de nombreuses tâches itératives et chronophages, telles que le développement et l'entraînement de modèles (voir la figure 1).

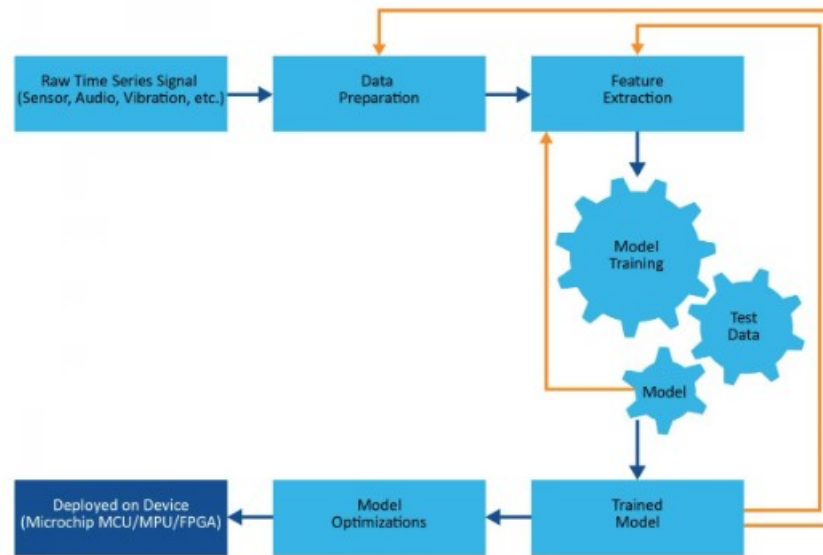


Figure 1 - Le développement de modèles de ML/AI est un processus itératif

Bien que ces étapes itératives puissent être automatisées, l'optimisation de la conception est une autre affaire. Même les ingénieurs ayant l'expérience de la conception d'applications de traitement de pointe peuvent éprouver des difficultés avec certains aspects de la ML/AI. De nombreux compromis devront être faits entre les performances du système (largement déterminées par la taille/complexité du modèle et le volume de données), la consommation d'énergie et le coût. En ce qui concerne les deux derniers points, la figure 2 illustre la relation inverse entre les performances (requises) et le coût, et indique la consommation d'énergie pour les types de dispositifs Microchip utilisés dans les applications typiques d'inférence ML.

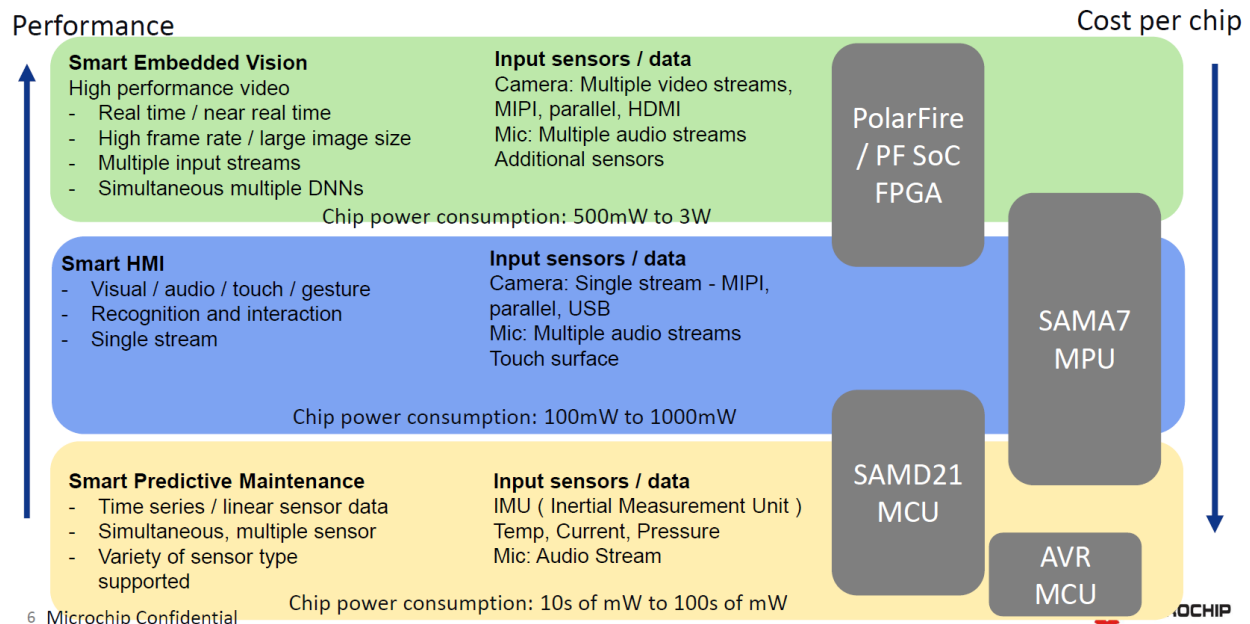


Figure 2 - Ci-dessus, exemples de types d'appareils et d'applications d'inférence ML dans lesquelles ils sont généralement utilisés.

Concevoir pour les petits formats

Comme nous l'avons mentionné, même les MCU 8 bits peuvent être utilisés pour certaines applications de ML. La popularité de tinyML, qui permet d'exécuter des modèles sur des micros aux ressources limitées, est l'un des éléments qui rendent cela possible et qui contribuent grandement à faire entrer le ML/AI dans la communauté des ingénieurs.

Pour s'en convaincre, il suffit de considérer qu'un MCU ou MPU haut de gamme destiné à des applications de ML/AI fonctionne généralement entre 1 et 4 GHz, nécessite entre 512 Mo et 64 Go de mémoire vive et utilise entre 64 Go et 4 To de NVM pour le stockage. Il consomme également entre 30 et 100 W. tinyML, quant à lui, est destiné aux MCU qui tournent entre 1 et 400 MHz, ont entre 2 et 512 Ko de RAM et utilisent entre 32 Ko et 2 Mo de mémoire non volatile pour le stockage. La consommation d'énergie se situe généralement entre 150 μ W et 23,5 mW, ce qui est idéal pour les applications qui sont alimentées par des batteries, qui dépendent fortement de l'énergie récoltée ou qui sont alimentées par induction.

Les clés de la mise en œuvre de tinyML résident dans la capture et la préparation des données, ainsi que dans la génération et l'amélioration des modèles. La capture et la préparation des données sont essentielles si l'on veut que des données significatives (un ensemble de données) soient disponibles tout au long du processus de ML (voir figure 3).



Figure 3 - Déroulement du processus d'apprentissage automatique.

Les machines supervisées (et semi-supervisées) ont besoin d'ensembles de données à des fins de formation. Un ensemble de données est une collection de données déjà classées dans un certain ordre. Ainsi, pour notre exemple de caméra de sécurité intelligente, la machine serait entraînée à l'aide de photos de personnes debout, marchant et courant (entre autres). L'ensemble de données peut être créé à la main, mais beaucoup sont disponibles dans le commerce, comme MPII Human Pose, qui comprend environ 25 000 images extraites de vidéos en ligne.

Cependant, l'ensemble des données doit être optimisé pour l'utilisation. Trop de données rempliront rapidement la mémoire. Si les données sont insuffisantes, la machine ne parviendra pas à faire des prédictions ou fera des prédictions erronées ou trompeuses.

En outre, le modèle ML/AI doit être de petite taille. À cet égard, une méthode de compression populaire est l'"élagage de poids", dans laquelle le poids des connexions entre certains neurones du modèle est fixé à zéro, ce qui signifie que la machine n'a pas besoin de les inclure lorsqu'elle fait des déductions à partir du modèle. Les neurones peuvent également être élagués. La quantification est une autre technique de compression. Elle réduit la précision des paramètres, des biais et des activations d'un modèle en convertissant les données dans un format de haute précision, tel que la virgule flottante 32 bits (FP32), en un format de moindre précision, par exemple un entier 8 bits (INT8).

Avec un ensemble de données optimisé et un modèle compact, il est possible de cibler un MCU approprié. À cet égard, il existe des cadres qui facilitent cette tâche. Par exemple, le framework/flux TensorFlow permet de choisir un nouveau modèle TensorFlow ou de recycler un modèle existant. Ce modèle peut ensuite être compressé dans un tampon plat à l'aide du convertisseur TensorFlow Lite, chargé sur la cible et quantifié.

Résumé

La ML et l'IA utilisent des méthodes algorithmiques pour identifier des modèles/tendances dans les données et faire des prédictions. En plaçant la ML/AI à la source des données (la périphérie), les applications peuvent faire des déductions et prendre des mesures sur le terrain et en temps réel, ce qui rend le système global plus efficace (en termes de performances et d'énergie) et plus sûr.



Grâce à la disponibilité de matériel adapté, d'IDE, d'outils et de kits de développement, de cadres, d'ensembles de données et de modèles libres, les ingénieurs peuvent développer assez facilement des produits de traitement de pointe basés sur la ML/AI.

Les ingénieurs en systèmes intégrés et l'industrie dans son ensemble vivent une période passionnante. Cependant, il est important de ne pas faire de l'ingénierie à outrance. Beaucoup de temps et d'argent peuvent être consacrés au développement d'applications ciblant des puces dotées de plus de ressources que nécessaire, qui consomment donc plus d'énergie et sont plus coûteuses.

A propos de l'auteur :

Yann LeFaou dirige une équipe qui développe des technologies tactiles capacitives et dirige également l'initiative d'apprentissage automatique (ML) de l'entreprise pour les microcontrôleurs et les microprocesseurs. Il a occupé plusieurs autres postes techniques et marketing au sein de Microchip, et a notamment dirigé les activités de marketing mondial de la société pour les technologies tactiles capacitives, d'interface homme-machine et d'appareils électroménagers. M. LeFaou est diplômé de l'ESME Sudria en France.