L'apprentissage profond sur microcontrôleur est l'avenir de l'edge computing

Il n'est plus nécessaire de disposer de processeurs capables d'effectuer des milliers de milliards d'opérations par seconde (TOPS) pour faire de l'apprentissage automatique (ML). Dans un nombre croissant de cas, les microcontrôleurs les plus récents, dont certains sont dotés d'accélérateurs ML intégrés, sont en mesure de fournir une fonctionnalité ML aux dispositifs en périphérie de réseau. Explications de NXP Semiconductors.

Il y a encore quelques années, on pensait que l'apprentissage automatique (ML, Machine Learning), et même l'apprentissage profond (DL, Deep Learning), ne pouvait se faire que sur du matériel haut de gamme, l'apprentissage et les inférences en périphérie de réseau (edge) étant assurés par des passerelles, des serveurs edge ou des centres de données. L'hypothèse était valable à l'époque, car la tendance à la répartition des ressources informatiques entre le cloud et l'edge n'en était qu'à ses débuts. Mais la situation a radicalement changé grâce aux efforts intenses de recherche et développement réalisés par l'industrie et le monde universitaire.

Le fait est qu'aujourd'hui, il n'est plus nécessaire de disposer de processeurs capables d'effectuer des milliers de milliards d'opérations par seconde (TOPS) pour faire de l'apprentissage automatique. Dans un nombre croissant de cas, les microcontrôleurs les plus récents, dont certains sont dotés d'accélérateurs ML intégrés, sont en mesure de fournir une fonctionnalité ML aux dispositifs en périphérie de réseau.

Non seulement ces dispositifs sont capables de faire de l'apprentissage automatique, mais ils peuvent le faire bien, à faible coût, en consommant très peu d'énergie et en ne se connectant au cloud que lorsque cela est absolument nécessaire. En bref, les microcontrôleurs dotés d'accélérateurs ML intégrés constituent la prochaine étape de l'intégration des traitements directement au sein de capteurs qui génèrent des données exploitables grâce à l'IoT (Internet des objets), comme les microphones, les caméras ou les capteurs qui surveillent des paramètres environnementaux.

AUTEUR

Ali Osman Ors, directeur, AI and Machine Learning Strategy and Technologies, Edge Processing, NXP Semiconductors.

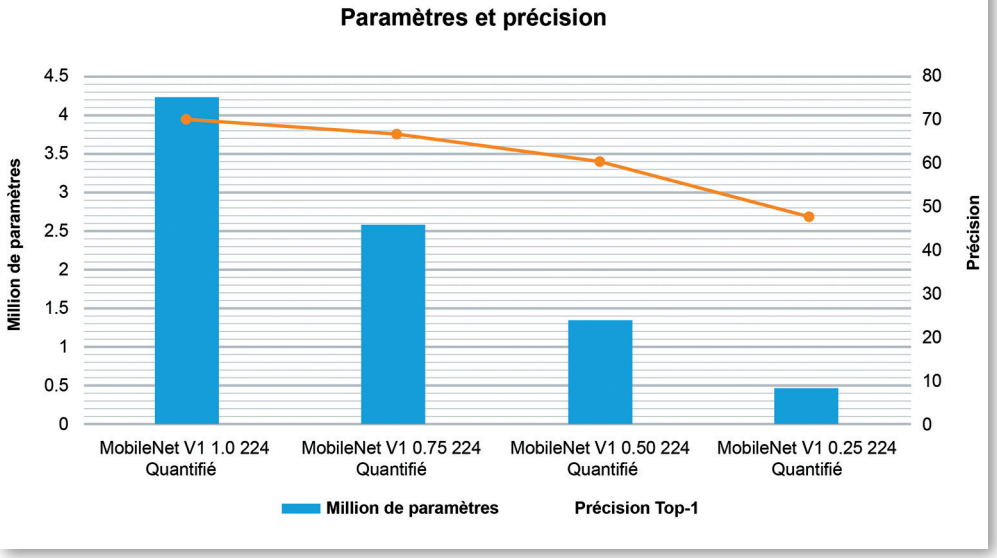
Jusqu'où s'étend la périphérie de réseau ?

Si l'edge est généralement considéré comme le point le plus éloigné d'un réseau IoT (par rapport aux plates-formes hébergées dans le cloud), on l'envisage le plus souvent comme une passerelle évoluée ou un serveur en périphérie de réseau. Mais l'edge ne s'arrête pas vraiment là. Il s'étend jusqu'aux capteurs, à proximité de l'utilisateur. Il devient donc logique de placer autant de puissance analytique que possible près de l'utilisateur, une tâche pour laquelle les microcontrôleurs sont justement parfaitement adaptés.

On pourrait penser que les calculateurs monocartes (SBC) peuvent aussi participer à l'edge computing, car ils sont capables de performances remarquables et, s'ils sont regroupés, ils peuvent même rivaliser avec un petit superordinateur. Mais ils sont encore trop gros et trop coûteux pour être déployés par centaines ou par milliers dans des applications à grande échelle. Ils nécessitent aussi une alimentation externe en courant continu qui, dans certains cas, peut être au-delà de ce qui est disponible, alors qu'un microcontrôleur ne consomme que quelques milliwatts et peut être alimenté par des piles boutons voire même quelques cel-

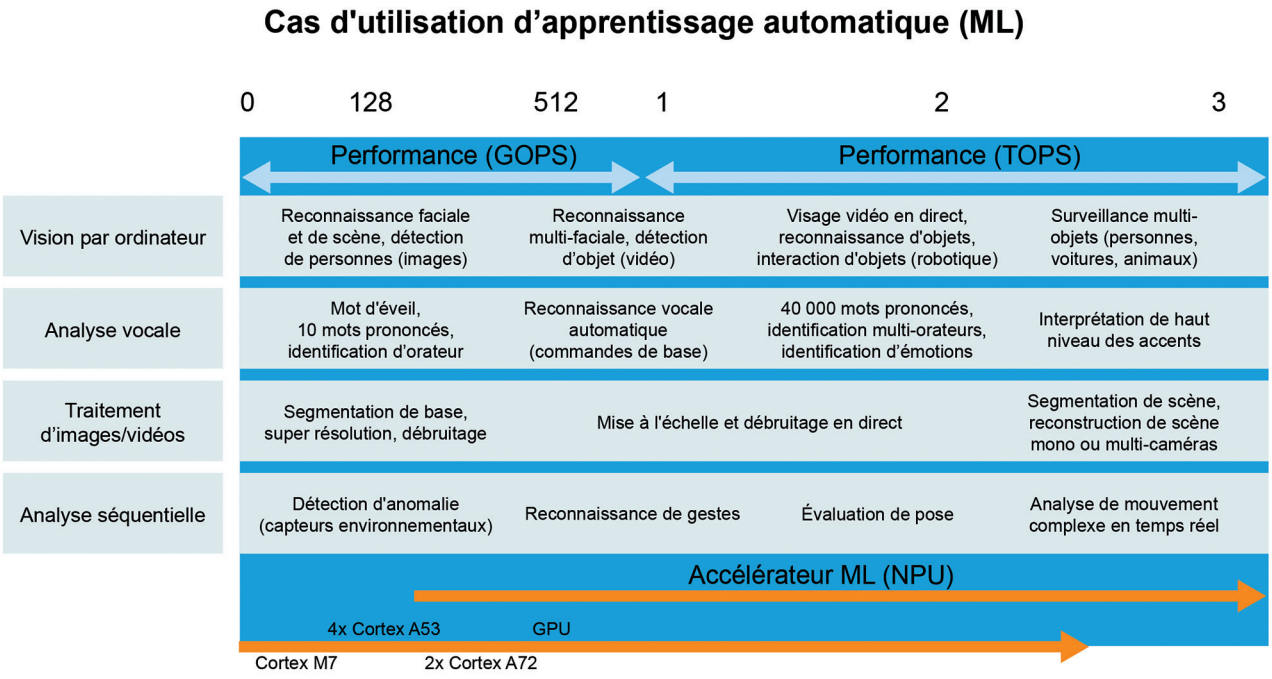
1 NOMBRE DE PARAMÈTRES VS. PRÉCISION

Les exemples de modèles MobileNet V1 avec différents coefficients multiplicateurs montrent un impact radical sur la précision et le nombre de paramètres et de calculs. Toutefois, le simple fait de modifier le multiplicateur de 1,0 à 0,75 n'a qu'un effet minime sur la précision TOP-1, alors que cela a un impact significatif sur le nombre de paramètres et de calculs.



2 CAS D'USAGE DE L'APPRENTISSAGE AUTOMATIQUE

Bien que le TinyML soit un paradigme relativement nouveau, il produit déjà des résultats surprenants en matière d'inférences et d'apprentissage et ce avec une perte de précision minimale. Parmi les exemples récents, citons la reconnaissance vocale et faciale, les commandes vocales et le traitement du langage naturel, et même l'exécution en parallèle de plusieurs algorithmes de vision complexes.

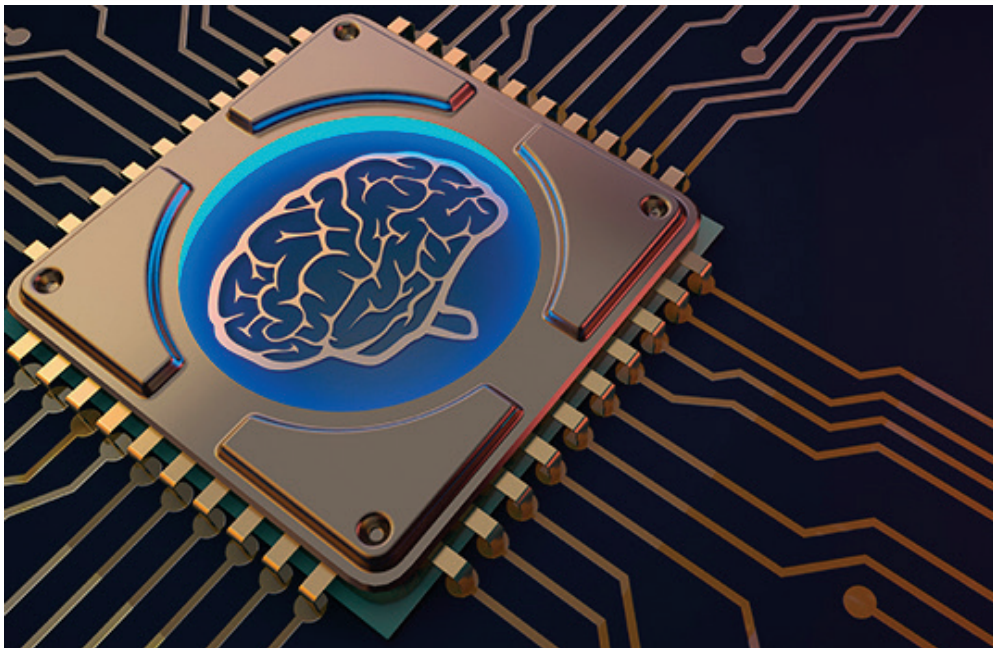


lules photovoltaïques. Il n'est donc pas surprenant que l'utilisation de microcontrôleurs pour faire de l'apprentissage automatique en périphérie de réseau soit devenu un domaine de développement très actif. Ce domaine porte même un nom : « TinyML » (ML compact). L'objectif du TinyML est de permettre l'exécution d'inférences, et à terme d'effectuer de l'apprentissage, sur de petits dispositifs de faible consommation et aux ressources limitées, et notamment sur des microcontrôleurs, plutôt que sur des plateformes de plus grandes dimensions ou dans le cloud. Il faut donc réduire la taille des modèles de réseaux de neurones pour les adapter aux ressources relativement modestes dont disposent ces dispositifs en matière de traitement, de stockage et de bande passante, sans trop réduire les fonctionnalités ou la précision (figure 1).

Une telle approche optimisée en termes de ressources permet aux dispositifs d'ingérer suffisamment de données générées par les capteurs pour faire le travail, tout en ajustant la précision et en réduisant les besoins en ressources. Ainsi, même si certaines données sont toujours envoyées dans le cloud (ou peut-être d'abord vers une passerelle edge, puis vers le cloud), elles seront beaucoup moins nombreuses car une analyse approfondie aura déjà été effectuée.

Un exemple bien connu d'un processus TinyML en action est un système de détection d'objets reposant sur une caméra qui, bien que capable de capturer des images en haute définition, ne dispose que d'un stockage limité et nécessite qu'on réduise la définition. Cependant, si la caméra dispose de fonctions d'analyse embarquées, seuls les objets intéressants sont capturés plutôt que la scène entière, et étant donné que les images pertinentes sont moins nombreuses, la définition supérieure peut être conservée. Cette capacité est généralement associée à des composants plus gros et plus

● L'utilisation de microcontrôleurs pour faire de l'apprentissage automatique (ML) en périphérie de réseau est devenu un domaine de développement très actif. Ce domaine porte même un nom : « TinyML » (ML compact).



puissants, mais la technologie TinyML permet de la mettre en œuvre sur des microcontrôleurs.

Petit mais costaud

Bien que le TinyML soit un paradigme relativement nouveau, il produit déjà des résultats surprenants en matière d'inférences (même avec des microcontrôleurs de puissance modeste) et d'apprentissage (sur des microcontrôleurs plus puissants) et ce avec une perte de précision minimale. Parmi les exemples récents, citons la reconnaissance vocale et faciale, les commandes vocales et le traitement du langage naturel, et même l'exécution en parallèle de plusieurs algorithmes de vision complexes (figure 2).

En pratique, cela signifie qu'un microcontrôleur coûtant moins de 2 dollars, doté d'un cœur Arm Cortex-M7 à 500 MHz et d'une mémoire intégrée de 28 à 128 Ko, est capable de fournir les performances nécessaires pour rendre des capteurs vraiment intelligents.

Même à ce niveau de prix et de performances, de tels microcontrôleurs embarquent de multiples fonctions de sécurité, notamment AES-128, s'accommodent de plusieurs types de mémoires externes, sont dotées d'interfaces Ethernet, USB et SPI, et intègrent ou prennent en charge divers types de capteurs, ainsi que des interfaces Bluetooth, Wi-Fi et audio SPDIF et I²C. Pour un coût supplémentaire, le microcontrôleur sera généralement doté d'un cœur Arm Cortex-M7 à 1 GHz, d'un Cortex-M4 à 400 MHz, de 2 Mo de RAM et d'un accélérateur graphique. La consommation ne sera typiquement que de quelques milliampères à partir d'une alimentation de 3,3 V_{DC}.

Quelques mots à propos des TOPS

Les consommateurs ne sont pas les seuls à utiliser une mesure unique pour définir les performances. Les concepteurs le font tout le temps et les services marketing adorent cette méthode puisqu'en tant que spécification principale, elle simplifie la différenciation entre les appareils... du moins en apparence. Un exemple classique est le processeur, qui a été défini pendant de nombreuses années par sa fréquence d'horloge, ce qui, heureusement pour les concepteurs et les utilisateurs, n'est plus le cas. Utiliser une mesure unique pour évaluer un processeur revient à évaluer les performances d'une voiture en fonction de son seul régime moteur maximum. Cela n'est pas complètement dénué de sens, mais cela n'a pas grand-chose à voir avec la puissance du moteur ou les performances de la voiture, étant donné que de nombreux autres facteurs entrent en ligne de compte dans ces caractéristiques.

Malheureusement, c'est aussi de plus en plus le cas pour les accélérateurs de réseaux de neurones, y compris ceux intégrés à des microprocesseurs ou à des microcontrôleurs hautes performances, qui sont spécifiés pour des milliards ou des milliers de milliards d'opérations par seconde parce que, là encore, c'est un nombre facile à retenir. Mais dans la pratique, les GOPS et les TOPS pris seuls ne veulent pas dire grand-chose et représentent une mesure (sans doute la plus élevée) effectuée dans un laboratoire sans vraiment correspondre à un environnement opérationnel réel. Par exemple, les TOPS ne tiennent pas compte des limitations de bande passante de la mémoire, de la surcharge du processeur, des pré et post-traitements et d'autres facteurs. Si l'on tient compte de tous ces éléments et d'autres, comme les performances sur une carte spécifique en fonctionnement réel, les performances au niveau système seront probablement entre 50% et 60% de la valeur TOPS indiquée sur la fiche technique.

Tout ce que ces chiffres indiquent, c'est en fait le nombre d'éléments de calcul présents dans le matériel, multiplié par leur vitesse d'horloge, plutôt que la fréquence à laquelle les données seront disponibles au moment où le système en aura besoin. Si les données étaient toujours immédiatement disponibles, si la consommation d'énergie n'était pas un problème, si les contraintes de mémoire n'existaient pas et si l'algorithme était parfaitement adapté au matériel, ces résultats seraient beaucoup plus significatifs. Mais le monde réel ne présente pas d'environnements aussi idéaux.

Lorsqu'elle est appliquée aux accélérateurs ML des microcontrôleurs, la métrique a encore moins de valeur. Même si ces minuscules dispositifs ont généralement une valeur de 1 à 3 TOPS, ils peuvent néanmoins offrir les capacités d'inférence suffisantes pour de nombreuses applications ML. Ces dispositifs s'appuient également sur des processeurs Arm Cortex spécialement conçus pour les applications ML à faible consommation. Avec la prise en charge des opérations sur les nombres entiers et flottants et les nombreuses autres caractéristiques du microcontrôleur, il devient évident que le nombre de TOPS, ou toute autre mesure unique, est incapable de définir de manière appropriée les performances, que ce soit seul ou au sein d'un système.

Conclusion

La volonté d'effectuer des inférences sur des microcontrôleurs directement intégrés dans des capteurs ou reliés à ceux-ci, comme des caméras fixes ou vidéo, est en train d'émerger, à l'heure où le domaine de l'IoT évolue pour que davantage de traitements se fassent au niveau edge. Ceci dit, le rythme de développement des processeurs d'application et des accélérateurs de réseaux de neurones embarqués au sein de microcontrôleurs est assez rapide, et des solutions plus performantes apparaissent régulièrement. La tendance est donc à la consolidation de fonctionnalités plus centrées sur l'IA, comme le traitement par réseau de neurones, et d'un processeur d'application dans un microcontrôleur, sans pour autant augmenter sa consommation ou sa taille de manière spectaculaire.

Aujourd'hui, les modèles peuvent être entraînés sur un CPU ou un GPU plus puissant, puis mis en œuvre sur un microcontrôleur à l'aide de moteurs d'inférence comme TensorFlow Lite pour en réduire la taille afin de répondre aux contraintes de ressources du microcontrôleur. Une mise à l'échelle peut être réalisée pour répondre à des exigences plus importantes en matière de ML. Bientôt, il devrait être possible non seulement d'effectuer des inférences, mais aussi de faire de l'apprentissage sur ces dispositifs, ce qui fera effectivement du microcontrôleur un concurrent encore plus redoutable vis-à-vis de certaines solutions de traitement plus lourdes et plus chères. ■