PyTorch s'ouvre à l'API Android Neural Networks

Connexion établie entre PyTorch et l'API Android Neural Networks (NNAPI). La jonction n'est encore que partielle. Elle s'inscrit dans une série de fonctionnalités expérimentales destinées à tirer parti de l'accélération matérielle pour l'inférence sur les terminaux mobiles.

> *[Announcing at PyTorch DevDay Livestream: https://t.co/uNSqJwDA0e]*
>
> *NNAPI support for PyTorch allows Android apps to run computationally intensive NN on the most powerful/efficient parts of the chips in mobile phones, including DSPs and NPUs. Learn more: https://t.co/KUeIBLI32t*
>
> *— PyTorch (@PyTorch) November 12, 2020*

Sur la partie GPU, les API Vulkan (Android) et Metal (iOS) sont mises à contribution. La passerelle avec la NNAPI vise les autres accélérateurs – essentiellement les NPU et les DSP. Facebook – créateur de PyTorch – a mené des tests avec un modèle intégré dans l'application Messenger. En l'occurrence, celui qui gère les arrière-plans immersifs (voir ci-dessous).

Sur un smartphone Google Pixel 5, le recours à la NNAPI a réduit de moitié la latence mesurée avec un cœur CPU. Sur un Pixel 4, il l'a divisée par près de quatre. Moyennant une sollicitation plus importante du processeur.

	Normalized latency		
	Pixel 3	Pixel 4	Pixel 5
Executing using 1 CPU core	1.000	1.000	1.000
Executing using 2 CPU cores	0.604	0.535	0.639
Executing using 4 CPU cores	0.373	0.337	0.443
Executing using NNAPI	0.721	0.275	0.494
CPU Consumption with NNAPI offload	32%	73.3%	46.5%

Le différentiel est plus significatif sur l'architecture MobileNetV2 (vision par ordinateur).

	Normalized Latency		
	Pixel 3	Pixel 4	Pixel 5
Executing using 1 CPU core	1.000	1.000	1.000
Executing using 2 CPU cores	0.554	0.478	0.546
Executing using 4 CPU cores	0.294	0.261	0.377
Executing using NNAPI	0.292	0.095	0.197
CPU Consumption with NNAPI offload	33.6	73.3	46.5

L'utilisation de ce modèle avec la NNAPI a requis quelques étapes de préparation, listées dans ce tutoriel. De manière générale, quelques adaptations sont nécessaires. PyTorch présente effectivement des différences sémantiques avec la NNAPI. Par exemple sur l'organisation des poids synaptiques en mémoire. Ou sur la représentation des opérations de suréchantillonnage.

Pour ce « premier jet », le tandem PyTorch-NNAPI ne fonctionne qu'à partir d'Android 10. La prise en charge d'Android 8 et d'Android 9 est sur la feuille de route, sans échéance définie. Même chose pour les modèles qui exploitent le contrôle de flux ou encore le *fallback* automatique sur CPU.

Illustration principale © ProStokStudio – shutterstock.com