第10章 Nano/Xavier NX/AGX Orinで

Jetson大研究2：TensorRTを使った最適化と推論実験

土井 伸洋

表1　深層学習の推論処理向け最適化ツール
他社製デバイスに対応しているツールもある

名　称	開発元	対象デバイス
TensorRT	エヌビディア	エヌビディア製GPU，SoC（Jetsonシリーズ）
TensorFlowLite	グーグル	Edge TPU，汎用デバイス※1
OpenVINO	インテル	インテル製CPU，GPU，VPU※2
SNPE SDK	クアルコム	Qualcomm製DSP，GPU，CPU

※1：対応するAndroidデバイス，iOSデバイス，組み込みLinux向けデバイス（ラズベリー・パイなど）
※2：Movidiusシリーズ

前章にて，認識モデルのエッジ・デバイスへの移植と，その際に実施した最適化の効果を述べました．本章ではこの点を掘り下げ，Jetsonシリーズ向けの最適化ツールであるTensorRTの利用法を紹介します．さらに，同じモデルをJetsonシリーズの3機種で動作させ，速度を比較します．

対象となるデバイスは，Jetson NanoとJetson Xavier NX，Jetson AGX Orinです．以降では，それぞれNano，Xavier NX，AGX Orinと表記します．

最適化ツールによる推論処理の高速化

● それぞれのデバイス向けに用意されている

深層学習の推論をCPUやGPUで行う場合，学習済みモデルをそのまま動作させることもできます．しかし，デバイスやアクセラレータに特化した最適化ツールを用いてモデルを最適化すると，デバイスの能力を最大限に活用でき，結果として処理（基本的には推論処理に限る）を高速化できます．

エッジ・デバイスは，数～十数Wという低電力で動作する代わりに，演算能力が限られていますので，最適化は必須となることが多いです．代表的な最適化ツールと対象デバイスを**表1**に挙げます．

今回利用するデバイスはJetsonシリーズです．最適化ツールとしてTensorRTを用います．

● 最適化で行われる具体的な処理

▶量子化

最適化ツールで行われる代表的な処理が量子化です．通常FP32（単精度浮動小数点数）演算で実施される推論処理を，FP16（半精度浮動小数点数）演算化したり，INT8（整数）演算化したりします．演算ビット長が減ることによる演算速度向上の他，モデル容量を削減できるメリットがあります．量子化された演算を実行するには，もちろんデバイスやアクセラレータの側のサポートが必要です．

演算精度の変更による推論精度低下を防ぐため，どのツールでも自動または手動のキャリブレーション手段が提供されています．

▶ハードウェアへの最適化

汎用的な演算で記述された部分をアクセラレータが得意とする演算へ変換します．インテル製CPUであれば，AVX（Advanced Vector Extension）命令を持っていますし，クアルコム製DSPであれば多様なMAC（積和演算ユニット）があります．

TensorFlowやPyTorchで出力できる認識モデルは汎用的な演算で記述されています．最適化ツールはモデルの中で使われている演算を解析し，演算を置き換えたり，複数の演算を融合したりすることで，処理速度向上を図ります．

ターゲットとするエッジ・デバイス

本章で利用するデバイスは，Nano，Xavier NX（**写真1**），AGX Orin（**写真2**）という3台のJetsonです．3台の概略スペックを**表2**に示します．

第9章で実験に用いたNanoと比べると，後発のデバイスによるAI処理速度は圧倒的です．Nanoが浮動小数点数演算（FP32，FP16）でのAI処理にしか対応できないのに対して，Xavier NXとAGX OrinはINT8演算にも対応できるのが大きな要因の1つです．

演算ユニットの主力は，CUDA Coreと呼ばれる汎用の並列計算ユニットですが，これ以外にもAI処理に寄与するユニットが2つ追加されています．