4月号特集で使ったJetson Orinの活用術

Jetson大実験…int8量子化でモデル容量を半分にしつつ精度低下を防ぐ

土井 伸洋

写真1　自動運転向けアプリケーションが動作している
動画像に対して自分の車が走っているレーン領域を抽出する

前回はtrtexecコマンドによるTensorRTでの最適化を実施しました(FP16量子化，**写真1**)．その結果，フレームワーク組み込みのTensorRTよりもさらに最適化効果を得られました．今回も引き続き最適化(主に高速化)について掘り下げていきます．なお，記事の前提条件については前回の記事を参照ください．

<前回の記事はこちら>
https://interface.cqpub.co.jp/wp-content/uploads/if2306_019.pdf

trtexecコマンドでのint8量子化

● int8量子化も引数を変更するだけで実施できる

前回ではFP16(半精度浮動小数点数)量子化のみを行いましたが，引数を変えることでINT8(整数)量子化も実施できます．コマンドを**リスト1**に示します．Dockerコンテナ内で実施します．

実行は正常に終了し，trtファイルも生成されました．最適化処理にかかった時間はFP16の場合とほとんど変わりませんでした．

リスト1　trtexecコマンドでのint8量子化

```
$ export PATH=/usr/src/tensorrt/bin/:$PATH
# 最適化の実施
$ trtexec --onnx=./model/model_resnet18_seg.onnx ¥
--saveEngine=./models/model_resnet18_seg_INT8.trt ¥
--int8 ¥ # INT8による量子化を指定
--useSpinWait
```

図1　INT8による量子化…推論の失敗

● 推論結果は残念な形に…

早速，生成したモデルを推論プログラムから読み出し，結果を視覚化してみました．残念なことにほとんどのフレームで推論が失敗しました(**図1**)．

ほとんどのフレームで，左右いずれかの白線を検出することができず，レーン領域の抽出にも失敗という結果になりました．実行時間こそFP16のときより低減されましたが，これでは利用することができません．これを低減する手段を講じていきます．

int8量子化における精度低下を防ぐ

● 精度を維持する方法…PTQとQAT

多くのプラットフォームにおいて，int8量子化を行う場合は精度を維持するための追加手順が必要です．大きく分けると以下の2つがあります．

- PTQ (Post Training Quantization)
- QAT (Quantization Aware Training)