A-01

組み込みシステムにおける TensorflowLite の性能評価

Performance evaluation of TensorflowLite in embedded system

甲藤　裕大 †　　中本 幸一 †
Yudai Katto　　Yukikazu Nakamoto

1. はじめに

従来の組み込みシステムは特定の用途のみに使用されていたが，マイクロプロセッサの進化等により多機能的を持たせられることが可能になった．さらに汎用 OS が組み込みシステムでも利用されるようになり機械学習が組み込みシステムでも適用などが可能となってきている．組み込みシステムでは利用場面により PC と比べ使用可能メモリの大きさ，実行時間遅れが問題となる場合がある．

本稿では組み込みシステム向けの機械学習ライブラリである TensorflowLite を用いて，組み込みシステムにおける機械学習の実行性能を評価する．

2. Tensorflow，TensorflowLite について

2.1 Tensorflow

Tensorflow とは Google が開発した機械学習向けに開発されたオープンソースプラットフォームである．このプラットフォームはニューラルネットワークモデルの構築，訓練，推論をクラウド，デバイス上などで実行することが可能である．また対応プログラミング言語も幅広く C，C++，Java，Python などが使用可能である．

2.2 TensorflowLite

TensorflowLite は Android デバイス，組み込み Linux などのモバイルデバイス向けに高速で機械学習を可能とするライブラリであり、Tensorflow と同様に C，C++，Java，Python がプログラミング言語として使用可能である．図 1 に示すように Tensorflow を用いて構築，訓練したニューラルネットワークモデルを FlatBuffers ファイルに変換し，.tflite ファイルを用いて推論を行う．Tensorflow にて作成したモデルから TFLite モデルへの変換は TFLiteConverter を用いて行う．FlatBuffers とは Google が開発したクロスプラットフォームであり，GoogleProtocolBuffer のように使用時にパースしないなどの特徴があり、ファイル利用時の高速化が可能となっている．TensorflowLite では，この FlatBuffers ファイルへの変換時にモデルの重みを 32 ビット浮動小数点から 8 ビット整数に量子化することでモデルの軽量化，推論速度の高速化を行うことが可能である．FlatBuffers ファイルへの変換時に式 1[1] のように概算することで量子化を行っている．ここで realvalue は元の数値，int8value は 8 ビット量子化後の数値，zeropoint は 128（-1 -127 を 0 127 で表現するため），scale は式 2 に示すように算出される．この量子化によってモデルパラメータが浮動小数点 32 ビットから整数 8 ビットへと変化していることなどから約 1/4 ほどにモデルサイズが削減される．[3]

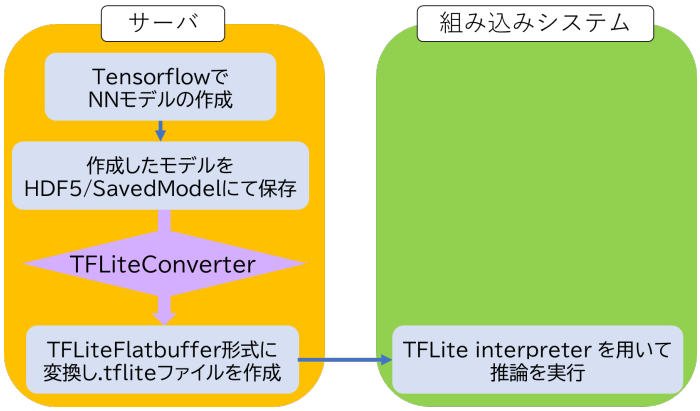

図 1　TensorfloLite の適応方法

$$realvalue = (int8value - zeropoint) \quad (1)$$

$$scale = realvalueMaxrange - realvalueMinrange \quad (2)$$

TensorflowLite は他にも ARMCPU の NEON 命令拡張を行うことによる推論の効率化などにより推論精度の向上，高速化を行っている．

3. 評価準備

3.1 評価方法

評価手順を以下に示す

（1） Tensorflow を用いてニューラルネットワークの一つ，CNN モデルを構築・訓練する．構築・訓練は GoogleColaboratory(以下 Googlecolab と呼ぶ) を用いて行う．訓練，推論時のデータは chromedriver を用いて Google の画像検索から 3 種類（人形，箱，球）それぞれ 300 枚とし，合計 900 枚のデータをダウンロードして用いた．ここで訓練データは全体の 4/5 である 720 枚，テストデータは全体の 1/5 である 180 枚を用いた

（2） 作成したモデルを TFLiteConverter を用いて.tflite ファイ

† 兵庫県立大学，University of Hyogo

ルへと変換する．この時モデルの重み変換を float 型，int 型それぞれ変換する

(3) 変換したファイルを組み込みデバイス上にて推論させ性能を評価する．

組み込みデバイスには RaspberryPi を用いた．表 1 に推論時の実行環境，Googlecolab の環境を示す．

表 1 推論時の実行環境

使用機器	RaspberryPi4B	Googlecolab
CPU	BroadcomBCM2711	Intel(R)Xeon(R)
GPU	-	Tesla T4
クロック周波数	1.50GHz	2.00GHz
メモリサイズ	4GB	13.3GB

3.2 使用言語・モデル

評価に使用したプログラミング言語は Python を用いた．CNN モデルは組み込み向け CNN とされる MobileNet，また比較のため DenseNet を用いた．

3.3 評価項目

性能評価として以下の項目を評価する

- 入力画像のデータサイズを変更させ MobileNet を構築・訓練させたときの推論精度・時間の評価
- MobileNet，DenseNet それぞれの推論精度・時間の評価
- MobileNet，DenseNet それぞれのモデルサイズ，モデルパラメータ

4. 結果

4.1 入力サイズごとの結果

入力データを変化させたときの結果を表 2 に示す．ここで比較のため Googlecolab 上にて Tensorflow での推論性能の結果も示す．ここで TF は Tensorflow，TFLite は TensorflowLite を示し，入力サイズは H × W の値を表記している．また推論時間はテストデータ 180 枚実行したときの画像 1 枚当たりの平均推論時間を示している．

結果として，Tensorflow と TensorflowLite を比較すると推論精度はほぼ同等であった．推論時間は TFLite の int8 と float32 を比べると最大 2 倍ほどの差であった．

表 2 入力サイズごとの推論性能結果

TF or TFLite	入力サイズ	推論精度	推論時間
TF(Googlecolab)	32 × 32	62.98 %	45.61ms
	112 × 112	80.11 %	44.06ms
	224 × 224	84.42 %	45.54ms
TFLite(Raspberrypi) int8 重み	32 × 32	61.54 %	6.917ms
	112 × 112	79.12 %	98.79ms
	224 × 224	83.51 %	397.1ms
TFLite(Raspberrypi) float32 重み	32 × 32	62.64 %	15.69ms
	112 × 112	79.67 %	139.1ms
	224 × 224	82.97 %	418.4ms

4.2 モデルごとの結果

表 3 には CNN モデルごとの推論精度，時間を，表 4 にはモデルのデータサイズを示す。入力データサイズは 112 × 112 としている．また TensorflowLite のモデルサイズ，パラメータ測定は Netron[4] を用いて測定した．

表 3 モデルごとの推論性能

TF or TFLite	モデル	推論精度	推論時間
TF(Googlecolab)	Mobilenet	80.11 %	44.06ms
	DenseNet	91.72 %	61.76ms
TFLite(Raspberrypi) int8 重み	Mobilenet	79.12 %	98.79ms
	DenseNet	91.21 %	195.0ms
TFLite(Raspberrypi) float32 重み	Mobilenet	79.67 %	139.1ms
	DenseNet	90.11 %	502.6ms

表 4 モデルごとのパラメータ、モデルサイズ

TF or TFLite	モデル	モデル サイズ	モデル パラメータ
TF	MobileNet	12.9MB	3.23M
	DenseNet	28.1MB	7.04M
TFLite int8 重み	MobileNet	3.23MB	3.19M
	DenseNet	7.18MB	6.95M
TFLite float32 重み	MobileNet	12.7MB	3.19M
	DenseNet	27.8MB	6.95M

表 3 より DenseNet の推論精度は MobileNet 同様 Tensorflow と TensorflowLite は同等であり，TensorflowLite の int8 と float32 を比べると DenseNet の方が推論時間が 2 倍以上差があり MobileNet に比べ推論時間への量子化の影響が大きかった．表 4 より Tensorflow と TensorflowLite の float32 のモデルサイズはほとんど同じであった．int8 と比べると量子化に伴い 1/4 倍ほどとなっていた．モデルパラメータは Tensorflow，TensorflowLite では 0.04M，0.09M ほどの差であった．

5. まとめ

まとめを以下に示す。

- TF, TFLite を比較すると推論精度は大きく下がらなかった．
- モデルパラメータ数が大きいネットワークの方が推論精度が高いが，量子化による推論時間の影響が大きかった．

参考文献

[1] Benoit Jacob, et al.: Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference The IEEE Conference on Computer Vision and Pattern Recognition、pp. 2704 - 2713, 2018.

[2] https://google.github.io/flatbuffers/

[3] https://www.tensorflow.org/lite/performance/model_optimization?hl=ja

[4] https://netron.app/