

SONY
Vision and Sensing
Application SDK モデル量子
化 機能仕様書

Version 0.1.0

2022 - 11 - 10

Table of Contents (目次)

更新履歴	1
用語・略語	2
参照資料	3
想定ユースケース	4
機能概要、アルゴリズム	5
操作性仕様、画面仕様	7
目標性能	13
制限事項	14
その他特記事項	15
未決定事項	16

更新履歴

Date	What/Why
2022/11/10	初版作成

用語・略語

Terms/Abbreviations	Meaning
MCT	モデルを量子化するためのオープンソースソフトウェア
Keras	AIモデルのフォーマットの一つ
TFLite	TensorFlow Liteのこと AIモデルのフォーマットの一つ
イテレーション	(1回あたりの)学習

参照資料

- ◆ Reference/Related documents (関連資料)
 - ◆ Model Compression Toolkit (MCT)
 - https://github.com/sony/model_optimization

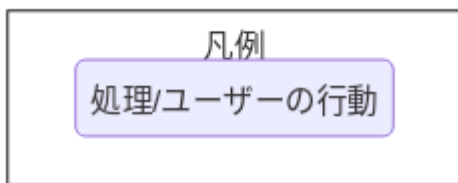
想定ユースケース

- ◆ モデルの量子化を行いたい
量子化を行うことでモデルのサイズを抑え、ターゲットエッジAIデバイスにデプロイできるようにしたい
- ◆ 量子化前と後のモデルを使用して推論実行し精度を確認したい

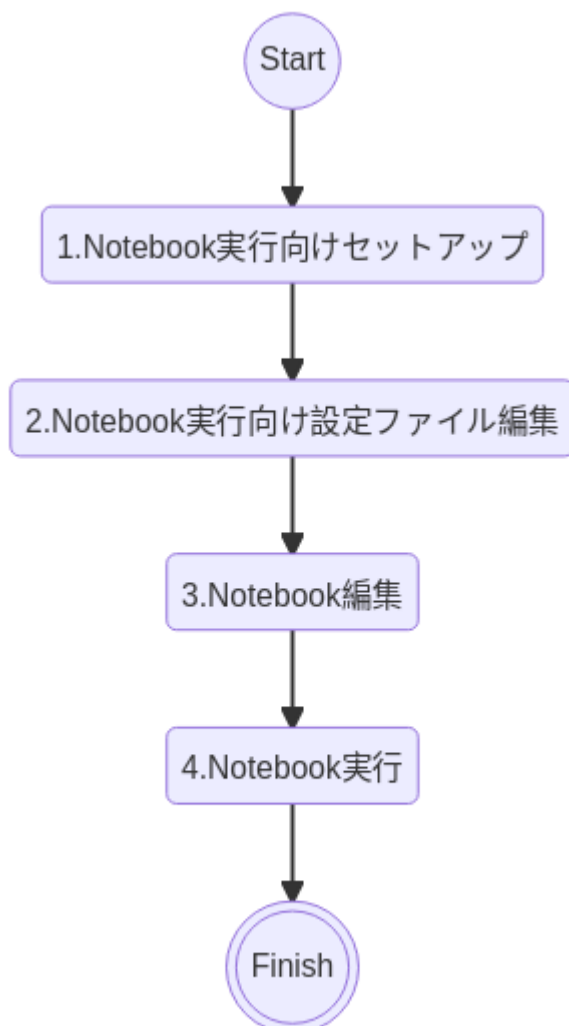
機能概要、アルゴリズム

Functional Overview

- ◆ SDKにて下記のフローでImage ClassificationのAIモデル(Keras)を量子化しAIモデル(TFLite)に変換できる
- ◆ 量子化前と後のAIモデルで推論実行し、推論実行結果の統計値(Top1 accuracy)を取得できる
- ◆ SDKにてサポートするAIモデルは、MCTの [supported-features](#) に準拠する



◆ フロー概要



◆ フロー詳細

1. Notebook実行向けセットアップ

- 変換対象となるAIモデル(Keras)を用意する
- 量子化のキャリブレーションに使用するため、AIモデルのtrainingに使用した画像を用意する
- 推論評価時に入力として使用するため、AIモデルのvalidationに使用する画像とそのground truth情報を用意する

2. Notebook実行向け設定ファイル編集

- 設定ファイル[configuration.json](#)を編集してNotebook実行時の設定を行う

3. Notebook編集

- 使用するAIモデルに応じてNotebook内のcalibration用preprocessing処理部の実装を修正する

4. Notebook実行

- AIモデル(Keras)を量子化しAIモデル(TFLite)に変換し、推論評価するNotebookを実行する

操作性仕様、画面仕様

How to start each function

1. SDK環境を立ち上げ、Topの `README.md` をプレビュー表示する
2. SDK環境Topの `README.md` に含まれるハイパーリンクから、`tutorials` ディレクトリの `README.md` にジャンプする
3. `tutorials` ディレクトリの `README.md` に含まれるハイパーリンクから、`quantize model`ディレクトリの `README.md` にジャンプする
4. `quantize model`ディレクトリの `README.md` に含まれるハイパーリンクから、`image classification`ディレクトリの `README.md` にジャンプする
5. `image classification`ディレクトリの各ファイルから各機能に遷移する

Notebook実行向けセットアップ

1. 変換対象となるAIモデル(Keras)を用意する
 - ◆ 変換対象となるAIモデル(Keras)を、SDK実行環境に格納する
2. 量子化のキャリブレーションに使用するため、AIモデルのtrainingに使用した画像を用意する
 - ◆ AIモデルのtrainingに使用した画像(300ファイル程度)が含まれるフォルダを、SDK実行環境に格納する
3. 推論評価時に入力として使用するため、AIモデルのvalidationに使用する画像とそのground truth情報を用意する
 - ◆ AIモデルのvalidationに使用する画像が含まれるフォルダを、SDK実行環境に格納する
 - ◆ AIモデルのvalidationに使用する画像のground truth情報ファイルを、SDK実行環境に格納する
 - ground truth情報ファイルを作成する場合は、下記の形式で作成する
 - validationに使用する画像をファイル名で昇順にソートした順に、一行ごとに画像のground truthのidを記載する
 - 例:idとラベル、各画像ファイルが下記の場合、下記のground_truth.txtとなる

idとラベル

```
0 : car  
1 : bike  
2 : human
```

各画像ファイル

```
bike1.JPG  
bike2.JPG  
car1.JPG  
human1.JPG  
human2.JPG
```

ground_truth.txt

```
1  
1  
0  
2  
2
```



後述の「実行ディレクトリ」について、image classificationを実行する場合は `quantize_model/image_classification` ディレクトリとなる。

Notebook実行向け設定ファイル編集

1. 実行ディレクトリの設定ファイル(`configuration.json`)を編集する



特別な記載がある場合を除き、原則として省略は不可。



特別な記載がある場合を除き、原則として大文字小文字を区別する。

Configuration	Meaning	Range	Initial	Remarks
---------------	---------	-------	---------	---------

source_keras_model	変換元となるAIモデル(Keras)パス。KerasのSaved Model形式のフォルダまたはh5形式のファイルを指定する	絶対パスまたはNotebook(*.ipynb)からの相対パス	未指定(空文字)	
dataset_image_dir	量子化の際にキャリブレーションを行うためのデータセット画像を格納したディレクトリ	絶対パスまたはNotebook(*.ipynb)からの相対パス	./images	
batch_size	量子化の際にキャリブレーションを行う画像を小分けにして重みやバイアスなどの特徴を見つけるセット枚数	1以上かつ、 dataset_image_dir に含まれる画像枚数以下	50	
input_tensor_size	AIモデルの入力テンソルのサイズ(画像の一辺のピクセル数)	AIモデルの入力テンソルに準拠	224	
iteration_count	量子化時のイテレーション回数	1以上	10	
output_dir	変換結果AIモデルの出力先となるディレクトリ	絶対パスまたはNotebook(*.ipynb)からの相対パス	./output	
evaluate_image_dir	推論実行時に入力する画像を含むディレクトリ	絶対パスまたはNotebook(*.ipynb)からの相対パス	./evaluate/images	
evaluate_image_extension	推論実行時に入力する画像の拡張子	文字列	JPEG	

<code>evaluate_ground_truth_file</code>	推論実行時に入力する画像についてのground truth情報ファイルのパス	絶対パスまたはNotebook(*.ipynb)からの相対パス	<code>./evaluate/ground_truth.json</code>	
<code>evaluate_result_dir</code>	推論実行結果の統計情報を保存するディレクトリ	絶対パスまたはNotebook(*.ipynb)からの相対パス	<code>./evaluate/results</code>	

Notebook編集

1. 実行ディレクトリの量子化実行用Notebook(*.ipynb)を開く
2. Notebookの中のcalibration用preprocessing処理部 (`FolderImageLoader` の引数 `preprocessing=[resize, normalization]`) を編集する
 - ◆ 使用するAIモデルの学習時のpreprocessing処理に相当する処理となるよう、編集する

Notebook実行

1. 実行ディレクトリの量子化実行用Notebook(*.ipynb)を開き、その中のPythonスクリプトを実行する
 - ◆ その後下記の動作をする
 - 実行ディレクトリの`configuration.json`存在をチェックする
 - エラー発生時はその内容を表示し、中断する
 - `configuration.json` `source_keras_model`、`dataset_image_dir` の存在をチェックする
 - エラー発生時はその内容を表示し、中断する
 - `configuration.json` の下記の内容を読み取り、MCTへ必要な設定を行い、AIモデル(Keras)を量子化し変換する
 - `configuration.json` `source_keras_model`
 - `configuration.json` `dataset_image_dir`
 - `configuration.json` `batch_size`
 - `configuration.json` `input_tensor_size`
 - `configuration.json` `iteration_count`

- MCTなどの外製ソフトでエラー発生時は、外製ソフトが出力するエラーを表示し、中断する
- `configuration.json` `output_dir` に、MCTで量子化したAIモデル(TFLite)ファイル `model_quantized.tflite` と、TensorFlow標準機能でTFLiteに変換したAIモデル(TFLite)ファイル `model.tflite` を出力する
 - `output_dir` で指定するディレクトリがなければ作成し、そこに出力する
- 変換中はNotebookに下記のような表示をする(`iteration_count` が10の場合)

```

0%|          | 0/10 [00:00<?, ?it/s]
...
30%|██████    | 3/10 [00:15<00:35, 5.10s/it]
...
100%|██████████| 10/10 [00:50<00:00, 5.07s/it]
```

- `configuration.json` `output_dir`、`evaluate_image_dir`、`evaluate_ground_truth_file` の存在をチェックする
 - エラー発生時はその内容を表示し、中断する
- `configuration.json` の下記の内容を読み取り、tflite interpreterへ必要な設定を行う
 - `configuration.json` `output_dir`
 - `configuration.json` `evaluate_image_dir`
 - `configuration.json` `evaluate_image_extension`
 - `configuration.json` `evaluate_ground_truth_file`
 - `configuration.json` `evaluate_result_dir`
- 元のAIモデル(Keras)、TensorFlow標準機能でTFLiteに変換したAIモデル(TFLite)、MCTで量子化したAIモデル(TFLite)の3種のAIモデルで推論実行し、統計情報を表示する
- 統計情報を、`evaluate_result_dir` 配下に `results.json` ファイルとして保存する
- TensorFlowなどの外製ソフトでエラー発生時は、外製ソフトが出力するエラーを表示し、中断する
- AIモデル(TFLite)の推論実行中は下記のような表示をする(画像数が10の場合)

```

0%|          | 0/10 [00:00<?, ?it/s]
...
40%|██████    | 4/10 [00:03<00:05, 1.08it/s]
...
100%|██████████| 10/10 [00:09<00:00, 1.08it/s]

```

- AIモデル(Keras)の推論実行中はTensorFlowライブラリによるログを表示する
- 処理中でもNotebook Cell機能のStop Cell Executionで中断できる

目標性能

- ◆ SDKの環境構築完了後、追加のインストール手順なしに、AIモデル(Keras)を量子化しAIモデル(TFLite)に変換できること
- ◆ UIの応答時間が1.2秒以内であること
- ◆ 処理に5秒以上かかる場合は、処理中の表現を逐次更新表示できること

制限事項

◆ なし

その他特記事項

◆ なし

未決定事項

◆ なし