TensorFlow i TensorFlowLite kompatybilność operatorów

Marcin Cyc

Słowem wstępu

Modele są przetwarzane przez konwerter optymalizujący, w którym operacje mogą być pomijane lub łączone.

Zestaw operacji w Lite jest mniejszy.

Nie każdy model można przekształcić.

Zestaw operacji w Lite jest stale rozwijany.

Należy dokładnie rozważyć sposób konwersji oraz optymalizacji operacji aby zbudować model używany przez TensorFlowLite.

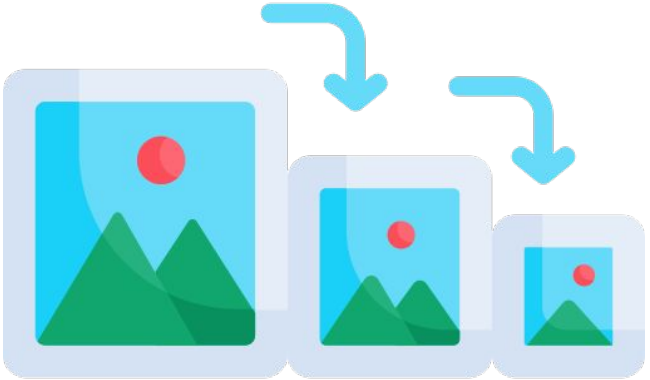
Wspierane typy

Ukierunkowanie na typy **float.32, uint8, int8**

Różnica konwersji ze względu na wykorzystany typ.

Przykład: wymagana informacja o zakresie dynamicznym dla tensorów w skwantyzowanej konwersji.

Kwantyzacja



Zmniejszenie precyzji liczb. Powoduje to:

- mniejszy rozmiar modelu
- szybsze obliczenia

Więcej na temat kwantyzacji:
https://www.tensorflow.org/lite/performance/model_optimization#quantization

Rodzaje kwantyzacji

Rodzaje kwantyzacji w TensorFlowLite:

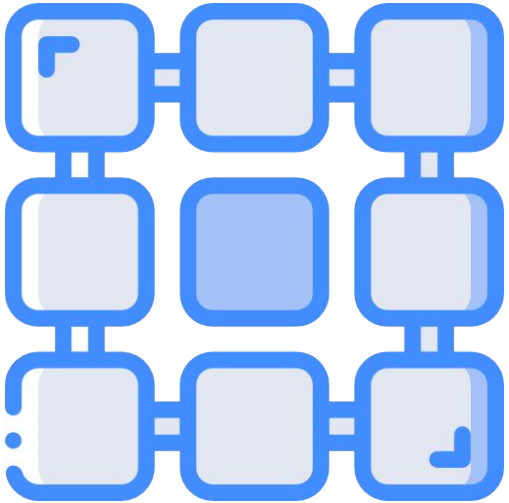
Technika	Wymagania danych	Redukcja rozmiaru	Dokładność	Wspierany hardware
Post-training float16 quantization	Brak danych	do 50%	Nieznaczna utrata dokładności	CPU, GPU
Post-training dynamic range quantization	Brak danych	do 75%	Utrata dokładnośći	CPU
Post-training integer quantization	Nieoznakowana reprezentatywna próbka	do 75%	Mniejsza utrata dokładności	CPU, EdgeTPU, Hexagon DSP
Quantization-aware training	Oznakowana reprezentatywna próbka	do 75%	Mniejsza utrata dokładności	CPU, EdgeTPU, Hexagon DSP

*TPU - jednostka przetwarzająca tensor, układ scalony stworzony przez Google do AI

Format danych i emisja

Obsługiwany format to **NHWC TensorFlow**. (wielkość_próbki x wysokość x szerokość x kanały)

Transmisja w ograniczonej liczbie operacji: **tf.add, tf.mul, tf.sub, tf.div**



Kompatybilne operacje

Poniżej wymienione operacje są zwykle mapowane na ich odpowiedniki w wersji Lite:

- `tf.nn.conv2d - O ile filtr jest stały.`
- `tf.nn.l2_normalize`
- `tf.reshape`
- `tf.nn.max_pool`

i wiele innych.

Pełna lista:
https://www.tensorflow.org/lite/guide/ops_compatibility#compatible_operations

Proste konwersje, ciągłe składanie i łączenie

Jest możliwość przetwarzania operacji bez bezpośredniego odpowiednika.

Dotyczy to operacji, które można:

- usunąć z grafu - if.identity
- zastąpić tensorami - tf.placeholder
- połączyć w bardziej złożone operacje tf.nn.bias_add

Niepełna lista takich operacji:
https://www.tensorflow.org/lite/guide/ops_compatibility#straight-forward_conversions_constant-folding_and_fusing

Nieobsługiwane operacje

Lista we wcześniejszym slajdzie zawiera najpopularniejsze operacje. Jeśli nie ma ta szukanej i popularnej operacji prawdopodobnie nie jest wspierana przez Lite. Taką operacją jest:

- tf.depth_to_space - Układa dane z głębokości (ang. deep) na bloki danych przestrzennych

Operacje TensorFlow Lite

Poniżej wymienione operacje są w pełni zastępują ich odpowiedniki w TensorFlow. Przykłady:

- ABS - wartość bezwzględna wejścia
- CONV_2D - wynik splotu 2D tensora wejściowego
- L2_NORMALIZATION - znormalizowany wektor wzdłuż ostatniego wymiaru
- RESHAPE - nowy kształt tensora wejściowego
- MAX_POOL_2D - każdy wpis jest maksimum w adekwatnym oknie
- operatory porównań - LESS, GREATER, EQUAL, NOT_EQUAL itd.

Pełna lista: https://www.tensorflow.org/lite/guide/ops_compatibility#tensorflow_lite_operations

Podsumowanie

Jeśli tworzymy model, który ma zostać wykorzystany na urządzeniach mobilnych/ o mniejszej mocy obliczeniowej, musimy:

- pamiętać o możliwości konwersji do modelu Lite
- upewnić się, że operacje są mapowanie
- upewnić się, że operacje jeśli nie są mapowanie to można zastąpić je, usunąć lub połączyć w bardziej złożone operacje
- upewnić się, że operacja nie jest na liście niewspieranych operacji
- pamiętać o możliwości kwantyzacji
- wykorzystać wspierany typy danych
- nie nadużywać broadcasting'u