임베디드 시스템에서 가속기 호환을 위한 텐서플로우 라이트 모델 변환

옥기수, 홍성수*
*서울대학교

{ksok, *sshong}@redwood.snu.ac.kr

Tensorflow Lite Model Conversion Compatible with Accelerators for Embedded Systems

Kisu Ok, Seongsoo Hong*
*Seoul National Univ.

요 약

Tensorflow lite는 대표적 심층 학습 프레임워크인 Tensorflow의 경량화 버전으로 자율주행차, 로봇, 드론 등 임베디드 시스템에서 심층 학습 추론을 지원하는 프레임워크이다. Tensorflow lite 는 양자화를 통한 모델 최적화, 프레임워크 경량화 등의 특성으로 임베디드 시스템에서의 심층 학습 추론에 장점이 있다. 이런 임베디드 시스템에서 성능을 최대한 끌어내기 위해서는 가속기와 Tensorflow lite 모델간의 호환이 되어야 한다. 본 논문에서는 임베디드 시스템의 가속기와 호환되는 Tensorflow lite 모델의 특성을 분석하고 가속기 호환성이 있는 Tensorflow lite 모델을 변환하는 방법을 알아낸다. Tensorflow lite 는 임베디드 시스템에서 추론을 위한 자원 소모를 최적화하기 위해 정적 텐서를 포함하는 레이어만 가속기에서 수행하도록 한다. 따라서 가속기 호환성을 위해서는 사전에 모델에 포함되어 있는 동적 텐서를 정적 텐서로 변환해야 한다. 이를 위해서는 해당 심층 학습 모델의 아키텍처를 새로이 빌드하고 사전 훈련된 데이터를 복원하는 과정이 필요하다. 본 논문에서는 Tensorflow 에서 사전 훈련된 객체 탐지 모델을 임베디드 가속기에 호환되는 Tensorflow lite 모델로 변환하고, Qualcomm Automotive AP 환경의 가속기에서 심층 학습 추론을 수행함을 보였다.

Ⅰ. 서 론

TFlite (Tensorflow lite)는 대표적 심층 학습 프레임워크인 Tensorflow 의 경량화 버전으로 자율주행차, 로봇, 드론 등 임베디드 시스템에서 심층 학습 추론을 지원하는 프레임워크이다 [1]. TFlite 는 추론만을 수행하도록 경량화된 프레임워크일 뿐만 아니라 심층 학습 모델을 양자화하는 기능을 제공하여 자원의 제약이 있는 임베디드 장치에서도 심층 학습 추론을 원활히 수행할 수 있도록 지원한다 [2].

이런 임베디드 시스템에서 심층 학습의 추론 성능을 최대한 끌어내기 위해서는 임베디드 시스템의 가속기를 이용하여 TFlite 모델을 수행하여야 한다. 본 논문에서는 임베디드 시스템의 가속기와 호환되는 TFlite 모델의 특성을 분석하고 가속기 호환성이 있는 TFlite 모델을 생성하는 방법을 알아낸다. TFlite 는 임베디드 시스템에서 추론을 위한 자원 소모를 최적화하기 위해 정적 텐서를 지닌 레이어만 가속기에서 수행하도록 설계되었다. 따라서 가속기 호환성을 위해서는 사전에 모델에 포함되어 있는 동적 텐서를 정적 텐서로 변환해야 한다. 이는 모델을 새로이 빌드 및 사전 훈련된 데이터를 복원하는 과정을 필요로 한다. 우리는 가속기에 호환되는 TFlite 모델을 생성하고 Qualcomm Automotive AP 환경의 가속기에서 심층 학습 추론을 수행함을 보였다.

Ⅱ. 가속기 호환 TFlite 모델 변환

이 장에서는 먼저 Tensorflow version 2 를 기준으로 Tensorflow 모델과 TFlite 모델에 대해서 설명한다. 그리고 임베디드 시스템의 가속기와 호환되는 TFlite 모델의 특징 및 모델 변환에 대해 설명한다.

1. Tensorflow 와 TFLite 모델

Tensorflow 모델은 심층 학습 추론을 수행하기 위해 설계된 레이어들의 집합이다. 여기서 레이어는 학습 가능한 변수들로 이루어진 함수로, 하나 이상의 텐서 입력과 하나 이상의 텐서 출력을 갖는다. 텐서는 다차원 배열로 데이터의 흐름을 나타낸다. 이 Tensorflow 모델은 Saved model 이라는 형식으로 저장된다. 이 모델을 배포할 때, 일반적으로 모델 구성을 표기한 pipeline.config 파일과 훈련 데이터인 checkpoint 를 함께 배포하게 된다.

Tensorflow 모델과 달리 TFlite 모델은 FlatBuffer 라는 형식으로 저장된다. TFlite 를 이용하여 모델을 수행하기 위해서는 Tensorflow 모델을 TFlite 모델로 변환하는 과정을 거쳐야 한다.

2. TFlite Delegate 와 가속기 호환 모델 변환

TFlite 모델을 임베디드 시스템에서 수행할 때 옵션으로 TFlite Delegate 를 사용할 수 있다. TFlite Delegate 는 GPU 혹은 DSP 와 같은 디바이스 내의 가속기를 활용하여 TFlite 모델의 하드웨어 가속을 지원하는 라이브러리다. Delegate 는 런타임 환경에서

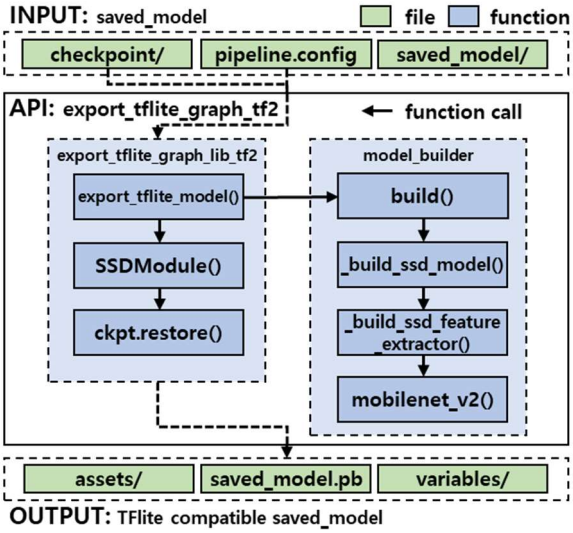

그림 1. TFlite 호환 ssd mobilenet v2 Saved model 생성 API 흐름

TFlite 모델을 분석하여 각 가속기에서의 호환 유무를 판단하며, 실행 가능한 레이어만 가속기에 위임시켜 실행한다.

임베디드 시스템의 성능을 최대한 끌어내기 위해서는 모델을 이루는 가능한 많은 레이어들이 Delegate 를 통해 가속기에서 실행되도록 해야 한다. 하지만 Tensorflow model 을 TFlite API 를 사용하여 TFlite 모델로 바로 변환하면, 대부분의 레이어들은 가속기에서 실행되지 않는다. 우리는 이 원인을 알아내기 위하여 대표적인 객체 탐지 모델인 ssd mobilenet v2 를 Tensorflow Models 의 export_tflite_graph_tf2 API 를 통해 가속기에서 실행가능한 모델로 변환하는 과정을 분석한다.

그림 1 은 우리가 분석한 과정을 보여준다. 첫째로, export_tflite_model()과 SSDModule()이 연속적으로 호출되어 모델의 pipeline.config 로부터 모델의 틀 역할을 하는 메타 모델을 생성한다. export_tflite_model()은 build(), _build_ssd_model(), _build_ssd_freature_extractor(), mobilenet_v2()을 연속적으로 호출하여 mobilenet v2 메타 모델을 작성한다. SSDModule()은 여기에 ssd 레이어를 추가하여 ssd mobilenet v2 의 메타 모델이 생성된다. 이 과정에서 pipeline.config 로부터 이미지 사이즈를 입력으로 받아 모든 동적 텐서를 정적 텐서로 변환하게 된다. 둘째로, chpt.restore()를 불러 메타 모델에 checkpoint 를 이용하여 훈련 데이터를 복원한다. 최종적으로 정적 텐서로만 이루어진 Saved model 이 파일이 생성된다.

분석 결과 동적 텐서를 정적 텐서로 변환함으로써 레이어들을 가속기에서 실행하도록 함을 알아냈다. 이런 과정이 필요한 이유는 동적 텐서가 포함된 레이어는 reshape 하기 위한 지연시간이 가속기에 위임하여 줄일 수 있는 수행시간보다 더 길 수 있기 때문에, 현재 TFlite Delegate 는 동적 텐서가 포함 되어있는 레이어의 경우에는 수행을 지원하지 않는 것으로 추정된다.

표 1. 대상 시스템의 HW 와 SW 구성

HW	Board	Qualcomm SA8195P
	CPU	Kryo 495 octa-core CPU
	GPU	Adreno 680
	DSP	Hexagon DSP
SW	OS	Automotive Grade Linux(AGL)
	framework	Tensorflow r2.7
	GPU library	OpenCL library
	DSP library	Hexagon library v1.20
	Kernel	Linux 4.14.180

표 2. TFlite 호환 모델 벤치마크 결과

Model	GPU	DSP
ssd mobilenet v2	25.1ms	9.6ms

Ⅲ. 대상 시스템에서의 실험

본 논문에서는 설명한 방식으로 변환된 모델의 수행을 확인 하기위해 표 1 에 기술된 하드웨어와 소프트웨어로 구성된 시스템에서 실험하였다. 실험에 활용된 모델은 ssd mobilenet v2 이다. GPU 위임을 확인 하기위한 floating point 32bit 모델과, DSP 위임을 확인 하기위한 integer 8bit 양자화 모델 두 가지 버전으로 변환된 TFlite 모델로 테스트하였다.

벤치마크 결과 floating point 32bit 모델의 레이어 98 개 중 98 개 모두 GPU 에 위임되는 것을 확인했다. Integer 8bit 양자화 모델은 DSP 에서 호환되지 않는 floating point 32bit 텐서가 포함된 레이어 7 개를 제외하고 레이어 98 개 중 98 개 모두 위임되는 것을 확인했다. 표 2 는 TFlite 의 벤치마크 도구를 통해 TFlite 모델이 가속기에 위임되어 실행하여 얻은 추론시간을 나타낸다.

Ⅳ. 결론

본 논문에서는 임베디드 시스템의 가속기와 호환되는 TFlite 모델의 특성을 분석하고 가속기 호환성이 있는 TFlite 모델을 변환하는 방법을 알아낸다. 가속기에서 TFlite 모델을 수행하기 위해서는 정적 요소로 구성된 그래프여야 하고 이를 위해서는 해당 심층 학습 모델의 아키텍처를 새로이 빌드하고 사전 훈련된 데이터를 복원하는 과정이 필요하다. 우리는 가속기에 호환되는 TFlite 모델을 생성하고 Qualcomm Automotive AP 환경의 가속기에서 심층 학습 추론을 수행함을 보였다.

ACKNOWLEDGMENT

이 논문은 2021 년 정부(산업통상자원부)의 재원으로 한국산업기술평가관리원의 지원을 받아 수행된 연구임 (과제번호 20014336, 2021 년 표준아키텍처기반 자율주행 AI SW 플랫폼 및 툴 체인 상용화 기술개발)

참 고 문 헌

[1] UZUN, Dr, and Mehmet Bilban. "Autonomous Vehicle and Augmented Reality Usage." International Journal of Engineering and Management Research (2019).

[2] Rashidi, Mitra. "Application of TensorFlow lite on embedded devices: A hands-on practice of TensorFlow model conversion to TensorFlow Lite model and its deployment on Smartphone to compare model's performance." (2022).