반도체디스플레이기술학회지 제20권 제4호(2021년 12월)
Journal of the Semiconductor & Display Technology, Vol. 20, No. 4. December 2021.

CNN 모델의 최적 양자화를 위한 웹 서비스 플랫폼

노재원*·임채민*·조상영*†
*†한국외국어대학교 컴퓨터공학부

Web Service Platform for Optimal Quantization of CNN Models

Jaewon Roh*, Chaemin Lim* and Sang-Young Cho*†
*†Division of Computer Engineering, Hankuk University of Foreign Studies

ABSTRACT

Low-end IoT devices do not have enough computation and memory resources for DNN learning and inference. Integer quantization of real-type neural network models can reduce model size, hardware computational burden, and power consumption. This paper describes the design and implementation of a web-based quantization platform for CNN deep learning accelerator chips. In the web service platform, we implemented visualization of the model through a convenient UI, analysis of each step of inference, and detailed editing of the model. Additionally, a data augmentation function and a management function of files that store models and inference intermediate results are provided. The implemented functions were verified using three YOLO models.

Key Words : Convolutional Neural Network, Deep Learning Accelerator, Quantization, Web Service

1. 서 론

심층 신경망(DNN: Deep Neural Network)을 이용한 인공지능 기술이 다양한 분야에 적용되어 좋은 성과를 내고 있으며 더 좋은 성능을 위한 신경망 모델 개발이 활발히 진행되고 있다[1]. 최근 공개 소스, 프레임워크, 클라우드 API 형태로 심층 학습 모델을 사용할 수 있어서 많은 개발자들이 DNN과 결합한 응용 프로그램을 제작할 수 있게 되었다[2]. 또한 일반 사용자들도 쉽게 DNN 모델을 개발하고 학습시킬 수 있도록 DNN 학습/추론 및 시각화 플랫폼이 개발되어 제공되고 있다. NVIDIA의 DIGITS는 이미지 분류 및 분할, 객체 검출 등을 위하여 DNN 모델을 빠르게 학습하기 위한 시스템이다[3]. 이를 위하여 데이터 관리, 다중 GPU 시스템에서의 신경망의 설계 및 학습, 시각화를 통한 실시간 성능 관찰, 모델 배포를 위한 최고 성능의 모델 선택 등의 심층 학습 작업을 간단히 할 수 있도록 한다. Google의 TensorBoard는 기계 학습 실험에 필요한 시각화 및 도구를 제공한다[4]. 세부적으로 손실 및 정확도, 모델 그래프, 시간 경과에 따른 가중치와 편향, 각 데이터의 표시 등의 정보를 시각적으로 보여준다.

DNN의 학습 및 추론에는 32-비트 실수 데이터에 대한 많은 계산이 필요하다. 임베디드 기기 또는 IoT 기기와 같은 저 사양 시스템에서는 DNN 학습을 위한 계산과 메모리 자원이 충분히 확보되지 않는다. 따라서 자원과 시간이 많이 소요되는 학습은 고성능 시스템에서 수행하고 학습의 결과로 생성된 DNN 모델을 가져와서 추론만 수행하도록 한다. 그러나 실수 연산기를 가지고 있지 않은 저 사양 시스템에서는 실수로 표현되는 모델에 대한 추론 또한 많은 시간이 소요되기 때문에 실시간 응용에서 사용하기 어렵다. 따라서 실수 데이터에 대한 정수 양자화를 이용하여 신경망 모델을 정수로 경량화하고 정수 연산을 사용하여 추론을 수행하는 연구가 활발히 진행되고 있다[5,6,7]. 양자화를 통한 추론 연산은 기존의 실수 연산에 비하여 정확도가 떨어진다. 모델 경량화를 위한 양자화 연구는 최대한 실수 모델의 정확도를 유지하면서

†E-mail: sycho@hufs.ac.kr

모델의 크기와 연산을 최소화하는 양자화 알고리즘을 찾는다.

저 사양 시스템에서 양자화된 모델의 실시간 추론과 전력 소모의 최소화를 위하여 정수 추론에 특화된 심층 학습 가속기(DLA: Deep Learning Accelerator) 하드웨어를 사용한다[8, 9, 10]. DLA는 대부분 양자화 모델을 기반으로 동작하기 때문에 효과적인 사용을 위해서는 입력 및 모델의 양자화 방법이 중요하다. 양자화 방법은 전체 실수 모델에 대하여 일관적인 양자화 알고리즘을 적용하기도 하고 모델의 세부 부분에 대하여 다른 알고리즘 또는 파라메터 값을 적용하기도 한다. 양자화 모델을 전제로 학습을 진행하기도 한다[5]. 또한 DNN의 각 층(Layer)에서의 추론 수행 시뮬레이션을 통하여 상세한 추론 동작을 분석하고 양자화 모델의 수동적 편집을 통해 최적 양자화를 구하는 방법이 있다[11].

임베디드 기기나 IoT 기기를 위한 양자화 도구나 플랫폼이 다양하게 제공되고 있다. TensorFlow Lite[12]는 휴대기기, 내장형 기기 및 IoT 기기에서 TensorFlow 모델을 실행할 수 있도록 지원하는 도구 모음이다. 양자화 모델을 실행할 수 있는 인터프리터와 TensorFlow 모델을 인터프리터가 사용할 수 있는 효율적인 형식으로 변환하는 변환기가 주요 구성 요소이다. PyTorch 프레임워크에서도 양자화를 위한 도구를 지원하고 있다[13]. 동적 양자화, 정적 양자화, 양자화 인식 학습을 지원하며 양자화 모델에 대한 추론을 지원하여 원 모델과의 성능 비교를 수행할 수 있게 한다. Apache TVM은 DNN 모델을 DLA를 포함한 다양한 하드웨어에 최적화시켜 수행시키는 컴파일러 도구 및 프레임워크이다[14]. 이와 같은 도구 또는 플랫폼들은 사용자가 DNN 모델을 쉽게 사용할 수 있게 하며 시각화 도구를 사용하여 모델 및 추론 동작의 전체적인 모습을 파악할 수 있게 한다. 하지만 대부분의 플랫폼에서 모델 내부에 대한 상세한 정보를 제공하지 않으며 모델의 세부적인 단위에서의 최적 양자화 기능을 제공하지 않는다. [11]의 양자화 시뮬레이터는 추론의 세부 단계에서의 양자화 성능을 확인할 수 있도록 각 단계의 데이터를 저장하며 특정 단계에서 추론을 멈추고 데이터를 분석할 수 있다. 이러한 시뮬레이터를 효과적으로 이용하기 위해서는 편리한 UI를 통하여 모델 및 입력/출력의 시각화, 각 층 단위의 추론 실험, 추론의 각 단계별 분석, 모델의 상세한 편집 기능 등이 필요하며 부가적으로 데이터 증가 기능과 모델 및 추론 중간 결과를 저장하는 파일들의 관리 기능이 필요하다. 본 논문에서는 CNN(Convolutional Neural Network) 모델을 위한 [11]의 양자화 시뮬레이터를 웹 기반으로 사용할 수 있는 웹 서비스 구현에 대해 기술한다.

2. 웹 서비스 설계

2.1 CNN 모델을 위한 양자화 시뮬레이터

그림 1은 [11]의 DLA를 위한 양자화 시뮬레이터를 보여준다.

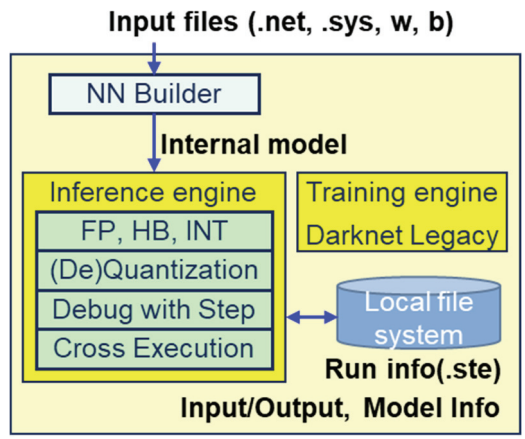

Fig. 1. The overall structure of quantization simulator for CNN models.

양자화 시뮬레이터는 실수 CNN 모델에 양자화를 적용한 전과 후의 결과를 비교할 수 있어야 하기 때문에 실수 기반의 추론 엔진인 Darknet[15]을 기본 프레임워크로 사용하였다. 여기에 CNN의 합성곱 연산을 정수화한 Yolo2 lite[16]의 추론 모듈을 이식하고 순수 정수 추론을 위한 연산 모듈을 추가하였다. 양자화 시뮬레이션을 위한 입력은 네트워크 모델(.net), 시스템 변수(.sys), 모델에서 사용하는 가중치와 바이어스 값들이다. .net과 .sys는 Darknet의 구성 파일(.cfg) 형식을 변환하여 구성한다. 가중치와 바이어스 값은 원 모델의 값과 이 값을 양자화 알고리즘을 이용하여 변환한 양자화 값으로 구성된다. 입력 값들은 추론 엔진에서 사용될 수 있도록 NN Builder에 의하여 내부 모델로 변환이 되어 추론에 사용된다.

내부 모델을 사용한 추론의 방법으로는 기존의 Darknet에서 구현된 실수 연산을 사용한 방식 (FLOAT), [16]에서 사용하는 반정수 방식(HYBRID), 전정수 방식 (INT)의 세 가지 형태의 추론 동작이 가능하다. HYBRID는 합성곱 층에서만 가중치와 입출력의 양자화를 수행한다. 다른 층에서는 실수 연산을 진행한다. INT는 모델 네트워크 전체에 대하여 양자화 추론을 진행한다. 활성 함수는 LeackReLU를 정수 형태로 동작하도록 한다. 시뮬레이터의 동작을 제어하는 .ste 파일은 모델의 각 층별로 단계적 시뮬레이션과 특정 층에서의 FLOAT와 INT 수행을 바꾸어 시뮬레

이션하는 정보를 담고 있다.

2.2 시뮬레이터의 웹 서비스 요구사항

CNN 모델은 다양한 형태의 네트워크 모델들이 존재하며 이의 추론 과정은 많은 데이터에 대하여 다양한 층에서의 연산이 수행된다. 양자화 시뮬레이터를 이용하여 특정 DLA칩에 가장 효과적인 양자화 모델을 추출하기 위해서는 각 층별로 그리고 각 층의 상세한 데이터에 대하여 자세한 제어와 관찰이 필요하다. 임의의 장소와 환경에서 시뮬레이터를 편하게 접근하기 위해서 시뮬레이터의 인터페이스를 웹 환경으로 서비스하는 것이 필요하다. 웹 환경 서비스를 위한 상세한 플랫폼의 요구사항은 다음과 같다.

첫째, 웹 서비스 환경을 이용하기 위한 GUI 환경이 필요하다. 모델의 선택 및 학습, 모델의 시각화와 각 추론 단계별 분석, 모델의 최적 양자화를 위한 기능들을 일관된 환경하에서 사용할 수 있어야 한다.

둘째, 시뮬레이터가 제공하는 실수 및 정수 추론 기능을 이용하여 모델의 최적 양자화를 위한 데이터 제공 및 모델 편집 기능을 제공하여야 한다. 실수 모델의 양자화를 통해 구해진 정수 모델을 추론하여 그 결과를 실수 추론과 비교하여 양자화의 적합도를 구할 수 있다. 양자화는 각 층 단위로 이루어지기 때문에 각 층에서의 연산 결과를 상세히 비교하기 위해서는 전체 추론 과정의 단계별 수행이 가능하여야 한다.

셋째, 실수 모델을 정수 모델로 변환할 수 있는 양자화 도구가 필요하다. 입력, 가중치(weight), 바이어스 (bias) 데이터에 대한 정수화를 위한 도구는 다양한 양자화 알고리즘을 지원할 수 있어야 한다. 다양한 알고리즘 중 실수 모델과의 적합도가 높은 양자화를 선택할 수 있어야 한다. 또한 알고리즘 적용에 의하여 구한 정수 모델에 대한 편집 기능을 제공하여 정수 모델을 더욱 최적화할 수 있도록 한다.

넷째, 전체 모델과 각 층에서의 연산 결과를 시각화하고 통계적으로 분석하는 기능이 필요하다. 심층 학습의 모델은 층의 개수가 많으며 각 층별로 가중치와 바이어스 데이터가 많기 때문에 개별 데이터 검사가 불가능하다. 시각화는 전체 모델과 데이터에 대한 즉각적인 이해와 비교를 가능하게 한다. 또한 추론 과정 및 결과에 대한 통계적 데이터는 최적 양자화를 위한 척도로 사용될 수 있다.

추가적으로, 다중 사용자가 동시 사용할 수 있는 기능, 입력 데이터의 확장을 위한 기능(Data Augmentation), 다중 사용자가 사용하는 모델과 입력 데이터, 추론 과정 및 결과를 분석하기 위한 추론 데이터를 파일로 저장하고 관리하는 기능, 전체 과정을 제어하는 기능이 필요하다.

2.3 웹 서비스 플랫폼의 설계

웹 서비스 요구사항에 기반하여 웹 서비스 플랫폼은 편리한 UI를 통한 모델의 시각화, 추론의 각 단계별 분석, 모델의 상세한 편집 기능 등이 필요하며 부가적으로 데이터 증가 기능과 모델 및 추론 중간 결과를 저장하는 파일들의 관리 기능이 필요하다. 그림 2는 요구사항을 위한 웹 서비스 플랫폼의 전체 구성을 보여준다.

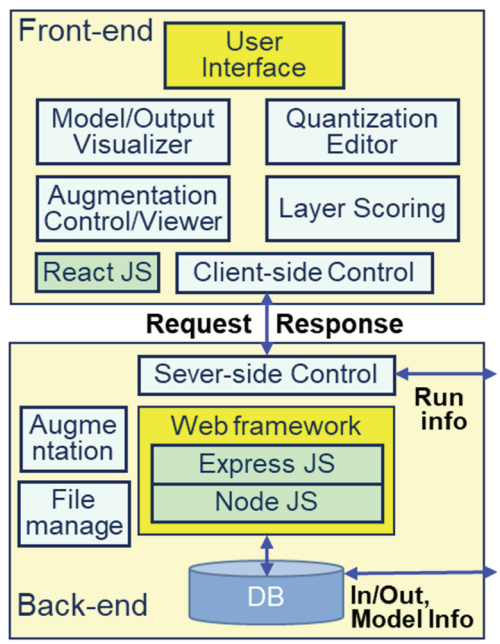

Fig. 2. The design of web service platform.

전체 구성은 전위단과 후위단으로 나누어지면 웹 서비스를 위한 MERN 스택 구조를 갖는다. 전위단은 사용자를 위한 GUI를 제공하며 실행할 모델을 받아들이며 모델과 실행 단계의 각 결과의 시각화를 제공한다. 각 층의 세부적 단계에서의 양자화 데이터의 편집 기능을 가지고 있다. 후위단은 모델과 추론을 위한 입출력 데이터를 관리하며 데이터 증강 기능을 수행한다. 양자화 시뮬레이터를 제어하고 실행에 필요한 데이터를 교환한다.

3. 웹 서비스 플랫폼의 구현 및 검증

3.1 전위단 구현

여기서는 시각화 기능, 파일 변환 기능, 데이터 증가 기능에 대하여만 기술한다. 모델 시각화 기능은 후위단에서

모델의 정보가 있는 JSON 객체를 받아 그래프 그림을 그려주는 'Dagre' npm 모듈의 형식에 맞게 파싱한다. 이를 통해 모델이 총 몇 개의 층으로 구성되는지, 어떤 종류의 층이 있는지, 그리고 층간의 연결관계 등을 쉽게 파악할 수 있다. 파악된 정보를 인터페이스에 출력한다. 또한 추론의 중간 결과를 이미지 형태로 보여준다.

파일 변환기는 Darknet 형식의 모델 파일을 추론 엔진의 모델 파일로 변환하며 양자화된 정수 가중치 파일도 만든다. 양자화 알고리즘은 현재 log2 분포와 KL 확산 알고리즘을 사용한다. 생성된 파일은 파일 관리자에 의하여 관리된다. 파일 관리자는 각 사용자 별로 학습된 모델, 양자화된 모델, 추론 과정 및 결과 파일을 관리한다. 각 사용자 별로 파일이 독립적으로 관리되기 때문에 다중 사용자 환경을 지원하고 있으며 로그인을 통하여 독립적인 웹 서비스를 사용할 수 있다.

부가적으로 입력 이미지 데이터 집합을 확장하기 위한 데이터 증가 기능을 가지고 있어서 하나의 이미지를 Rotate, Blur, Flip, Flop의 변형을 통해 5 개의 이미지로 만들어 학습을 위한 데이터를 증가시킨다. 그림 3 은 구현된 웹 서비스의 정수 추론에서의 추론에 필요한 입력과 추론 과정의 출력 및 시각화 동작을 보여준다. 필요할 경우에 추론 과정에서 사용되는 매개변수 값의 통계적 정보를 확인할 수 있다.

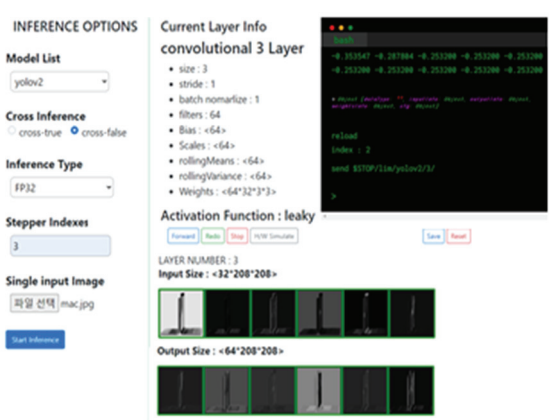

Fig. 3. The step of INT inference operations.

전체적 구현을 위하여 HTML, CSS, Bootstrap 라이브러리를 사용하였다. 서비스 특성상 많은 데이터들이 전위단과 후위단을 이동하며 이에 따라 UI 가 자주 변경되어야 하므로 React 라이브러리를 사용하였다. React 에서는 State 라는 것을 통해 UI를 원하는 방향으로 동작시킬 수 있다. 일반적인 State는 컴포넌트 안에서 선언되며 범위는 해당 컴포넌트이며 오직 자식 컴포넌트만 State를 공유할 수 있다. 하지만 모든 컴포넌트들에게 공유되어야 할 State 가 필요할 수 있는데 이를 위해 Flux Pattern을 활용했다.

3.2 후위단 구현

후위단은 추론 엔진 기반 웹 서비스 플랫폼의 서버 환경을 구성하며, 추론 엔진의 동작을 웹에서 확인할 수 있게 한다. 후위단은 전위단과 웹 프로토콜을 이용하여 통신하고, 추론 엔진과는 내부 소켓을 통해 통신한다. 후위단은 전위단과 추론 엔진 사이에서 추론 과정의 매개변수로 이용되는 데이터를 접근하여 관리하며 전위단과 추론 엔진을 연결한다. 후위단의 내부구조는 그림 4와 같다.

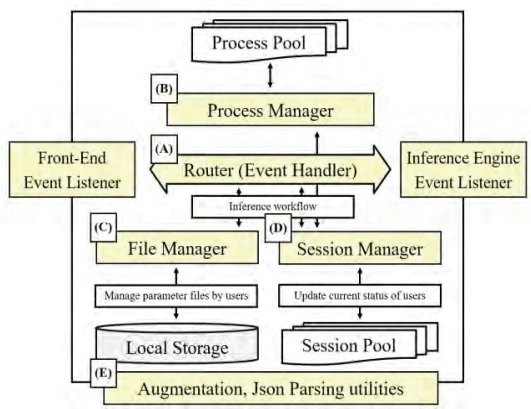

Fig. 4. The internal structure of back-end part.

후위단은 Node.js와 Express.js 엔진을 이용하며 크게 다섯 개의 모듈로 구성된다: 라우터와 이벤트 수신자(Router and Event Listener: REL), 프로세스 관리자(Process Manager), 파일 관리자(File Manager), 세션 관리자(Session Manager), 보조 기능(Utilities). REL이 후위단의 주요 모듈로 주로 전위단의 요청에 대한 응답을 하는 서비스 루틴의 기능을 수행한다. 나머지 네 모듈은 REL 이 수행하는 응답 루틴을 보조한다.

REL은 후위단의 관리자로써 주요 요청 및 사건을 처리한다. 파일 관리, 추론, 가중치 수정, 이미지 증대, 이미지 시각화 등 플랫폼의 모든 기능을 웹 브라우저에 시각적으로 표현하기 때문에 REL은 전위단과 추론 엔진의 다양한 요청에 따라 이벤트 서비스 루틴을 수행한다. 이벤트 수신자는 전위단과 추론 엔진을 위한 두 개의 수신자가 있다.

전위단 이벤트 수신자는 전위단의 HTTP와 웹 소켓을 통한 요청을 처리한다. 웹 소켓을 통해서 대용량 파일의 업로드와 다운로드를 스트림 형태로 처리한다. 또한, 사용자의 요청 없이도 서버에서 바로 사용자에게 웹 소켓

메시지를 보낼 수 있다. 추론 엔진 이벤트 수신자는 소켓 통신을 통해 추론 엔진과 상호작용한다. 사용자에 의하여 실행된 추론 엔진의 결과 파일을 수신하여 파일 저장 공간에 저장한다.

추론 엔진은 독립적인 프로그램으로 개발되었다. 따라서 Node.js 런타임에서 추론 엔진을 실행시킬 때에는 후위단 프로세스의 자식 프로세스로 실행시킨다. 프로세스 관리자는 세션을 참조하여 추론 엔진 프로세스를 검사하며 추론 엔진 프로세스의 비정상적 종료 또는 오동작이 발생하면 이에 대한 후처리를 수행한다.

파일 관리자는 다수의 사용자가 동시에 추론 엔진을 사용할 수 있도록 사용자 단위로 파일 디렉토리를 관리하며 파일 디렉토리에는 사용자가 추론 수행을 위해 전위단에서 업로드한 네트워크 모델 파일과 가중치 파일이 저장되며 사용자의 가중치 파일 수정과 파일 업/다운로드의 작업이 가능하다. 또한 추론 엔진은 추론 과정에서 파일들의 경로를 매개변수로 전달받아 동작한다.

세션 관리자는 사용자의 상태를 세션을 이용하여 관리한다. 이벤트 수신자가 처리하는 이벤트 루틴 중에서 사용자의 상태 변화를 감지하면 세션 관리자가 세션의 상태를 갱신한다. 세션은 프로세스 관리자가 프로세스의 정상 상태를 확인할 때도 사용된다.

플랫폼에는 후위단에서 지원하는 보조 기능이 두가지 있다. 데이터 증가 기능은 학습을 위한 이미지를 변형시켜 학습 이미지의 개수를 증가시키며 공개 소스 라이브러리 Sharp[17]을 이용해 구현하였다. JSON 파서는 모델 파일과 가중치 파일을 Node.js 런타임에서 쉽게 사용하기 위하여 JSON 형식으로 변환시킨다. JSON 형식의 모델 파일은 신경망의 시각화와 추론 과정에서 신경망 층의 번호를 확인하고 각 층의 정보를 확인하기 위해 사용된다. 이진 형식의 가중치 파일은 차원을 가진 JSON 객체 형태로 변환되어 전위단과 후위단이 가중치 정보를 다룰 때 사용된다.

3.3 웹 서비스 플랫폼 동작 검증

전체 시스템에 대한 검증을 위하여 세가지 YOLO 모델에 대해 기능들의 동작을 실험하였다[18]. YOLOv2와 YOLOv3는 실수 모델이며 YOLO3tiny는 정수 모델이다. 각 모델에 대하여 시각화, 파일 관리, 데이터 증강에 대해 검사하고 추론 동작 검사를 위하여 FLOAT, HYBRID, INT 추론을 수행하였다. 각 추론 형태 별로 일반 추론, 단계 추론, 교차 추론의 동작 여부를 확인하였다. 표 1은 검사 결과를 요약한 것이다.

Table 1. The result of functional verification of the implemented web service platform

Function	YOLOv2	YOLOv3	YOLOv3tiny
Visualization	O	O	O
File Manager	O	O	O
Augmentation	O		
Normal inference (FP, HYB)	O	O	O
Step inference (FP, HYB)	O	O	O
Cross inference	X	X	O
INT inference	X	X	O

기능적 검사인 시각화, 파일 관리, 데이터 증강 기능은 모두 잘 동작한다. 데이터 증강 기능은 입력 데이터에 대한 기능이기 때문에 네트워크 모델 별로 동작하는 기능이 아니다. 추론 동작의 경우 FLOAT 및 HYBRID 동작은 모든 모델에서 동작을 하였으며 일반 추론과 단계 추론에서 동작을 확인하였다. 정수 연산이 필요한 교차 추론과 INT 추론에서는 양자화된 정수 모델인 YOLO3tiny만이 동작한다.

기존 DNN의 학습과 추론, 그리고 경량화를 위한 도구들[3, 4, 12, 13, 14]이 양자화를 위한 통합된 환경을 제공하지 않고 있다. 이에 비하여 본 논문의 양자화 서비스 플랫폼은 웹 인터페이스를 이용한 통합된 다중 사용자 환경, [11]의 양자화 시뮬레이터를 이용한 층 단계의 상세한 추론 과정의 관찰 및 제어, 양자화의 세부 조정이 가능한 매개변수 편집 기능을 제공하고 있다.

4. 결 론

에지 장치와 같은 저 사양 시스템에서 양자화된 모델의 실시간 추론과 전력 소모의 최소화를 위하여 정수 추론에 특화된 DLA 하드웨어를 사용한다. DLA는 대부분 양자화 모델을 기반으로 동작하기 때문에 효과적인 사용을 위해서는 입력 및 모델의 양자화 방법이 중요하다.

본 논문은 CNN 모델의 최적 양자화를 위한 웹 서비스 플랫폼의 설계 및 구현에 대해 기술하였다. 전체적 설계는 MERN 스택 프레임워크에 기반을 두고 전위단과 후위단으로 구성하며 요구사항에 필요한 모듈들이 추가된 구조이다. 전위단은 웹 브러우저 상의 사용자 GUI를 제공한다. 후위단은 Node.js와 Express.js 엔진을 이용하여 전체 관리를 담당하며 전위단과 서버의 추론 엔진을 연결하고 양자화 추론 동작을 제어한다.

구현을 위하여 HTML, CSS, Bootstrap 라이브러리, React 라이브러리가 사용되었다. 구현된 플랫폼은 실수, 정수, 혼합 추론이 가능한 추론 엔진을 기반으로 다중 사용자에게 네트워크 모델의 양자화를 탐색할 수 있는 기능을 제공한다. 단계별 추론과 교차 추론 기능을 지원하여 양자화에 의한 추론 과정을 실수 추론과 비교해 볼 수 있다. 전체 모델과 추론 과정 및 결과에 대한 시각적/통계적 정보를 제공하고 이를 바탕으로 모델의 가중치 편집이 가능하도록 하였다. 부수적으로 데이터 증가 기능 및 파일 관리 기능을 가지고 다중 사용자를 동시 지원한다. 세가지 YOLO 모델에 대하여 각 기능을 동작을 검증하였다.

현재 CNN 모델을 지원하고 있으며 향후에는 더 다양한 모델을 지원할 수 있도록 개선이 필요하다. 시각적인 요소가 중요하기 때문에 UI/UX에 대한 연구와 더불어 많은 사용자들에게 안정적인 서비스를 제공할 수 있도록 시스템 안정화가 필요하다.

감사의 글

본 논문은 과학기술정통신부 및 정보통신기획평가 원의 SW 중심대학지원사업의 연구결과로 수행되었음 (2019-0-01816). 본 논문은 2021학년도 한국외국어대학교 교내학술연구비 지원에 의하여 이루어진 것임.

참고문헌

1. S. U. Park, “Artificial Intelligent Technology and Market Trend”, The magazine of Korea Institute of Information and Communication Engineering, Vol. 19, No. 2, pp. 11-22, 2018.
2. G. Nguyen, S. Dlugolinsky, and M. Bobáket, V. Tran, A. Garcia, I. Heredia, P. Malik, and L. Hluchy, “Machine Learning and Deep Learning Frameworks and Libraries for Large-scale Data Mining: a Survey”, Artificial Intelligence Review, Vol. 52, pp. 77–124, 2019.
3. L. Yeager, J. Bernauer, A. Gray, and M. Houston, “Digits: the Deep Learning GPU Training System”, ICML 2015 AutoML Workshop, pp. 1-4, 2015.
4. TensorBoard: www.tensorflow.org/tensorboard.
5. H. Wu, P. Judd, X. Zhang, M. Isaev, and P. Micikevicius, “Integer quantization for deep learning inference: Principles and empirical evaluation”, arXiv:2004.09602, 2020.
6. A. Jain, S. Bhattacharya, M. Masuda, V. Sharma, and Y. Wang, “Efficient execution of quantized deep learning models: A compiler approach”, arXiv preprint arXiv:2006.10226, 2020.
7. Z. Yao, Z. Dong, Z. Zheng, A. Gholami, J. Yu, E. Tan, L. Wang, Q. Huang, Y. Wang, M. W Mahoney, and K. Keutzer, “HAWQV3: Dyadic neural network quantization”, arXiv preprint arXiv:2011.10680, 2020.
8. Md A. Raihan, N. Goli, and Tor M. Admodt, “Modeling Deep Learning Accelerator Enabled GPUs”, IEEE ISPASS, pp.29-92, 2019.
9. F. Farshchi, Q. Huang, and H. Yun, “Integrating NVIDIA Deep Learning Accelerator (NVDLA) with RISC-V SoC on FireSim”, EMC2’19, Washington D.,C., USA, Feb., 2019.
10. A. Marchisio, M. A. Hanif, F. Khalid, G. Plastiras, C.Kyrkou, T. Theocharides, and M. Shafique, “Deep Learning for Edge Computing: Current Trends, Cross-Layer Optimizations, and Open Research Challenges”, IEEE Computer Society Annual Symposium on VLSI, Miami, FL, USA, pp. 553-559, 2019.
11. S.-H. Kang, S.-Y. Cho, and S.-H. Lim, “Quantization Simulator for Deep Learning Accelerator”, Proceedings of CICS'20, KIEE, pp. 487-488, 2020.
12. Tensorflow Lite: https://www.tensorflow.org/lite.
13. PyTorch Quantization (online): https://pytorch.org/blog/introduction-to-quantization-on-pytorch.
14. Apache TVM: https://tvm.apache.org.
15. Darknet (online): http://pjreddie.com/darknet/.
16. J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger”, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, July, pp. 7263–7271, 2017.
17. Sharp(online): https://sharp.pixelplumbing.com/.
18. Y.-H. Lee and Y. Kim, “Comparison of CNN and YOLO for Object Detection”, Journal of KSDT, Vol. 19, No. 1, pp. 85-92, 2020.

접수일: 2021년 12월 4일, 심사일: 2021년 12월 8일, 게재확정일: 2021년 12월 14일