심층학습 모델 커스터마이징과 추론을 위한 웹 서비스 플랫폼

노재원, 조상영, 임승호
한국외국어대학교 컴퓨터공학부
harryroh2003@gmail.com, sycho@hufs.ac.kr, slim@hufs.ac.kr

Web Service Platform for Customizing and Inference of Deep Learning Model

Jaewon Roh, Sang-Young Cho, Seung-Ho Lim
Division of Computer Engineering, Hankook University Foreign Studies

요 약

기계학습 모델의 전체 구조를 쉽게 파악하고 추론할 수 있으며, 추론 과정 중에 멈춰서 중간결과를 확인할 수 있는 디버깅, 그리고 customizing 까지 지원하여 기계학습에 더 익숙해지고 더 나아가, 실제로 활용해보는 GUI Platform 구현

1. 서론

인공 신경망(Artificial Neural Network)은 여러 응용분야에서 성공적인 결과를 내고 다양한 응용 분야로 적용되면서 네트워크 모델 개발이 활발이 이루어지고 있다[1]. 최근 공개 소스나 프레임워크 형태로 심층학습 모델을 사용할 수 있어서 많은 개발자들이 심층학습과 결합한 어플리케이션을 제작할 수 있게 되었다[2].

일반 사용자들도 심층 신경망(Deep Neural Network: DNN)의 학습을 용이하게 할 수 있도록 심층 신경망 학습 및 시각화 플랫폼도 개발되어 제공되고 있다. NVIDIA 의 DIGITS 는 DNN 모델을 학습하기 위한 데이터, 모델 개발, 시각화 기능을 제공한다[3]. Google 의 TensorBoard 는 기계학습 실험에 필요한 시각화 및 도구를 제공한다[4]. 이러한 플랫폼은 사용자가 DNN 모델을 쉽게 사용할 수 있도록 되어 있지만 모델 내부에 대한 깊은 이해를 갖기 어렵다. 이에 착안하여 사용자가 직접 weight 분포를 보고 수정하여 모델을 커스터마이징 할 수 있는 플랫폼을 제작하게 되었다.

본 논문에서는 CNN(Convolutional Neural Network) 모델의 양자화 시뮬레이션을 위한 웹 서비스 기반의 통합 환경에 대해 다룬다. IoT 의 에지 장치는 자원이 한정되어 있어서 기존 모델을 사용하기 어렵다. 양자화는 모델을 저해상도의 정수 모델로 변환하여 에지 장치에서의 추론 연산을 가능하게 한다. 현재 임베디드 장치나 모바일 제품을 위해 양자화 된 모델을 생성하고 실행시킬 수 있는 크로스 플랫폼 프레임워크인 Tensorflow Lite[3]를 예시로 들 수 있다. 여기서 핵심기능은 기존 텐서플로우 모델의 용량을 줄이고 최적화시키는 것이다. 이를 이용하여 많은 개발자들이 트랜드에 맞게 임베디드 장치나 모바일 장치를 위한 머신러닝 모델을 개발하고 제공할 수 있다. 하지만 이는 오직 텐서플로우 모델에 한해서만 사용할 수 있기 때문에 다양한 플랫폼의 모델에도 적용시키기 위해서 본 논문에서의 양자화 시뮬레이션 환경은 Darknet 을 기반으로 구축되었다. Darknet 의 실수 추론 기능에 반정수 하이브리드 추론 기능과 양자화를 이용한 정수 추론 기능이 추가되어 양자화 성능을 검사할 수 있다. 양자화 시뮬레이션 엔진을 사용하기 위하여 개발된 웹 서비스 플랫폼은 웹 브라우저 인터페이스를 통하여 세가지 추론 동작을 실행하면서 검증할 수 있도록 하였다. 이를 위하여 신경망 모델 및 추론의 시각화 및 분석 도구를 포함하며 양자화 추론을 위한 데이터 증가 및 실행 관리 모듈을 구현하였다.

2. 본론

2-1 웹 서비스 요구사항

웹 서비스 플랫폼의 목적은 심층학습 모델의 최적 양자화를 위한 GUI 환경을 제공하는 것이다. 특히, CNN 실수 모델을 기반으로 추론 정확도 감소를 최소화하면서 정수 모델로 변환할 수 있는 환경을 제공한다. 이를 위하여 실수 모델이 추론과 양자화된 정수 모델의 추론 진행과 결과를 비교할 수 있어야 한다.

이러한 환경을 제공하기 위한 요구사항은 다음과 같다.

첫째, CNN 네트워크 모델의 실수와 정수 추론 기능이 필요하다. 기존의 실수 모델의 양자화를 통해 구해진 정수 모델을 추론하여 그 결과를 실수 추론과 비교하여 양자화의 적합도를 구할 수 있다. 양자화는 각 층(Layer) 단위로 이루어지기 때문에 각 층에서의 연산 결과를 상세히 비교하기 위해서는 전체 추론 과정의 단계별 수행이 가능하여야 한다.

둘째, 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 커스터마이징 하는 기능이 필요하다.

세째, 전체 모델과 각 층에서의 연산 결과를 시각화하고 통계적으로 분석하는 기능이 필요하다. 심층학습의 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.

2-2 웹 서비스 구현

첫번째 요구사항에 대해 기존 실수 추론(FLOAT) 기능, 반정수 추론(HYBRID) 기능, 정수 추론(INT) 기능, 층별 추론(STEP) 기능, 교차 추론(CROSS) 기능을 제공한다. 추론 기능을 제공하는 서버단의 추론 엔진은 실수 추론을 지원하는 Darknet 플랫폼을 기반으로 만들어 졌다. 여기에 Convolution 층만 정수로 수행하는 Yolo2 Lite 를 이식하여 HYBRID 기능을 추가하고 8-비트 정수 연산을 수행하도록 하여 INT 기능을 추가하였다.

STEP 기능은 일반적인 추론의 경우 입력 이미지와 추론 형식(을 주면 끝까지 추론하여 하나의 결과를 보여주는데 반하여 추론하는 과정에서 원하는 층에 멈춘 후 해당 층의 정보, 중간결과, 입력, 출력, 가중치의 시각적/통계적 정보를 확인할 수 있다. 이때 필요에 따라 가중치를 직접 수정하여 현재 층을 수정된 값으로 다시 한번 더 추론할 수 있고 얼마나 달라졌는지 중간결과를 통해서 확인하는 디버깅 기능이 있다. CROSS 기능은 FLOAT 또는 INT 로 추론을 진행하다가 특정 층에서 멈추고 다른 형식으로 한 층을 추론하는 기능이다. 이 기능은 특정 층에서 정수 연산의 정확도를 실수 연산과 비교할 때 유용한다.

시각화 기능은 서버로부터 모델의 정보가 있는 JSON 객체를 받으면 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. 또한 중간 결과를 이미지 형태로 보여준다.

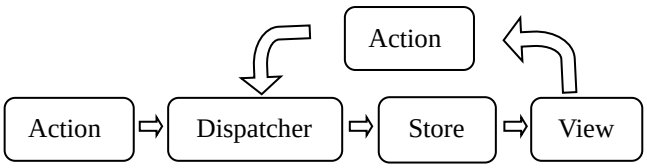
파일 변환기는 Darknet 모델 파일을 추론 엔진 모델 파일로 변환하며 양자화를 통해 정수 가중치 파일도 만든다. 양자화 알고리즘은 현재 log2 분포와 KL 확산 알고리즘을 사용한다. CNN 실수 모델을 입력하면 양자화를 통하여 정수 가중치 파일 생기며 파일 관리자에 의하여 관리된다. 파일 관리자는 각 사용자별로 학습된 모델, 양자화된 모델, 추론 과정 및 결과 파일을 관리한다. 각 사용자 별로 파일이 독립적으로 관리되기 때문에 다중 사용자 환경을 지원하고 있으며 로그인을 통하여 독립적으로 웹 서비스를 사용할 수 있다.

부가적으로 입력 이미지 데이터 집합을 확장하기 위한 데이터 증가(Augmentation) 기능을 가지고 있어서 하나의 이미지를 Rotate, Blur, Flip, Flop 의 변형을 통해 5 개의 이미지로 만든다.

전체적 구현을 위하여 HTML, CSS, Bootstrap 라이브러리를 사용하였다. 서비스 특성상 많은 데이터들이 전위단(Front-end)과 후위단(Back-end)을 이동하며 이에 따라 UI 가 자주 변경되어야 하므로 React[7] 라이브러리를 사용하였다.

2-3 전위단 기능 구현을 위한 상태 관리

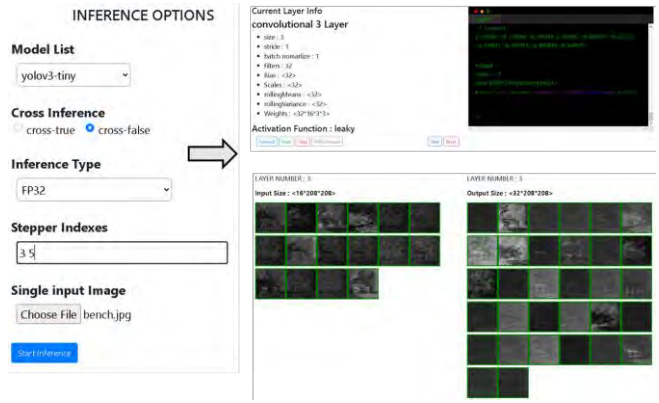
React 에서는 State 라는 것을 통해 UI 를 원하는 방향으로 동작시킬 수 있다. 일반적인 State 는 컴포넌트 안에서 선언되며 범위는 해당 컴포넌트이며 오직 자식 컴포넌트만 State 를 공유할 수 있다. 하지만 모든 컴포넌트들에게 공유되어야 할 State 가 필요할 수 있는데 이를 위해 Flux Pattern [8]을 활용했다. State 들을 Store 라는 곳에 저장해두고 컴포넌트들이 갱신하고 읽는 구조이다.



(그림 1) Flux Pattern

그림 1 에서 View 가 컴포넌트에 해당하는데, 많은 컴포넌트에서 검증되지 않는 데이터나 방법으로 Store 들을 갱신할 수 있으므로, 이를 방지하기 위해 Dispatcher 에서 갱신 Action 들을 관리한다.

현재 전위단에서 userStore, aliveStore, socketStore, editLayerWeightStore 의 네 가지 Store 가 있다. userStore 는 후위단에 GET 이나 POST 요청을 보낼 때 username 에 대한 정보가 여러 컴포넌트에서 필요한 경우가 많아 Store 에서 관리한다. aliveStore 의 목적은 웹 서비스의 모든 서비스에서 로그인이 되어있어야 하기 때문에 모든 컴포넌트에서 로그인 여부를 바로 확인하기 위함이다. 전위단의 거의 모든 서비스가 후위단과 Socket 또는 Socket-stream 통신을 한다. 이를 위한 하나의 Socket 객체를 socketStore 에 저장한다. 웹 서비스 중에서 가중치를 수정하는 서비스가 있었는데 editLayerWeightStore 에서 수정된 가중치들을 저장한다. 그림 2 는 구현된 웹 서비스의 정수 추론 동작 예를 보여준다.



(그림 2) 정수 추론 STEP 동작

3. 결론

심층학습 양자화 모델의 최적화를 지원하는 웹 서비스를 개발하였다. 웹 서비스는 실수와 정수 추론 및 양자화 기능을 지원하고 있다. 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 가중치 편집이 가능하도록 하였다. 부수적으로 데이터 증가 기능 및 파일 관리 기능을 가지고 다중 사용자를 동시 지원한다. 표 1 에서와 같이 YOLO 시리즈 모델에 대하여 동작을 검증하였다.

<표 1> YOLO 모델의 각 서비스에 대한 test 결과

기능	yolov2	yolov3	yolov3-tiny
Visualizer	O	O	O
FileManager	O	O	O
Normal Inference (FP, HYB)	O	O	O
Stepping Inference (FP, HYB)	O	O	O
Cross Inference	X	X	O
All INT8 inference	X	X	O
Augmentation	O		

에지 장치를 위한 심층학습 모델에 대한 생성부터 추론까지 전반적인 과정을 진행하는 웹 서비스 환경이 구축되었으며 다양한 양자화 알고리즘을 실험할 수 있다.

향후에는 더 다양한 모델을 지원할 수 있도록 개선이 필요하다. 시각적인 요소가 중요하기 때문에 UI/UX 에 대한 연구와 더불어 많은 사용자들에게 안정적인 서비스를 제공할 수 있도록 시스템 안정화가 필요하다.

"본 연구는 과학기술정통신부 및 정보통신기획평가원의 SW 중심대학지원사업의 연구결과로 수행되었음 "(2019-0-01816)

참고문헌

[1] 박상욱, "인공지능 기술 및 시장 동향", 한국정보통신학회지, 19 권, 2 호, pp. 11-22, 2018.
[2] G. Nguyen, S. Dlugolinsky, M. Bobák et al. "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 AutoML Workshop, 2015.
[4] TensorBoard(online): www.tensorflow.org/tensorboard
[5] Tensorflow Lite(online) : https://www.tensorflow.org/lite
[6] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", CVPR, 2016.
[7] React library(online): www.reactjs.org
[8] Flux Pattern(online): facebook.github.io/flux/