

딥러닝을 위한 인공지능망 표준 포맷 동향



강대기 동서대학교 부교수

1. 머리말

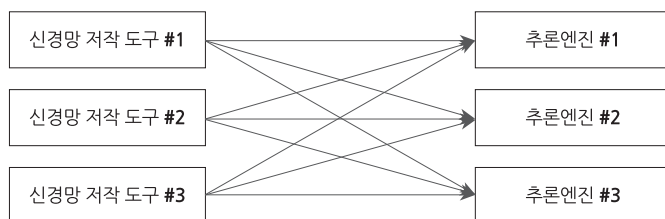
최근 등록이 마감된 NIPS(Neural Information Processing Systems) 2018 국제 컨퍼런스는 신경망 연구로 세계 제일의 컨퍼런스이다. 올해 NIPS 등록은 마치 K-POP 스타의 콘서트처럼 등록이 시작되자마자 12분도 채 안되어 모든 티켓이 다 팔리는 상황이 연출되었다. 게다가 2018년 이후의 컨퍼런스들을 미리 등록할 수 있게 따로 배분해 둔 티켓들조차도 2022년까지가 다 등록되었다고 한다. 이러한 놀라운 사건들은 신경망에 대해 사람들이 큰 관심을 보이고 있음을 알 수 있게 하는 것이다.

분명한 것은 딥 러닝 기술의 등장으로 다시 신경망의 연구에 대한 연구자들과 개발자들 그리고 기업가들의 관심이 높아지기 시작했다는 것이다. 이처럼 다양한 개발 툴들이 난립하는 현 상황에서 신경망에 대한 공통적인 포맷의 중요성은 매우 높다고 볼 수 있다. 본고에서는 대표적인 인공지능망 표준 포맷인 NNEF(Neural Network Exchange Format)와 ONNX(Open Neural Network Exchange)에 대해서 간단히 알아보겠다.

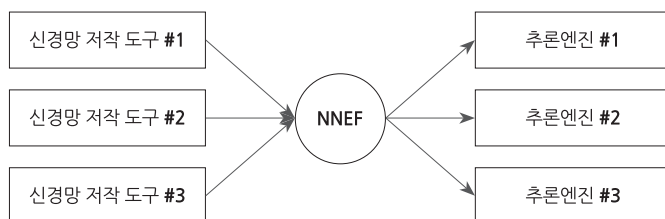
2. NNEF(신경망 교환 포맷)

신경망 교환 포맷(NNEF, Neural Network Exchange Format)은 머신러닝 시스템을 배치(deploy)할 때 생기는 파편화(fragmentation) 문제를 줄이기 위해 제안되었다. 이러한 파편화 문제는 현재 또는 가까운 미래의 기술 상황에서는 다양한 디바이스와 플랫폼들 상에서 다양한 응용 분야에 걸쳐 많은 신경망 학습 도구들과 추론 엔진들이 소수의 신경망을 구성할 수 있기 때문에 발생한다.

NNEF의 목적은 데이터 과학자들과 엔지니어들로 하여금 자신들이 학습에 사용하는 신경망 프레임워크에서 학습된 신경망을 여러 개의 다양한 추론 엔진으로 쉽게 이식할 수 있도록 하기 위한 것이다. 다양한 최종 단말 장비들에 신경망을 배치하기 위해 가장 중요한 것은 장비를 제작하는 사람들이 믿을 수 있는 안정적이고 융통성 있으며 확장가능한 표준을 제시하는 것이다. 따라서 NNEF는 학습된 신경망의 구조, 오퍼레이션, 패러미터들의 완벽한 전체 묘사를 신경망을 생성하기 위한 학습 도구나 신경망을 사용하고 실행하는 추론엔진과는 독립적으로 캡슐화하는 것이다.



[그림 1] 신경망 학습을 위한 저장 도구와 추론 엔진 간에 여러 개의 익스포터(exporter)가 필요한 경우



[그림 2] NNEF를 통한 신경망 저장 도구들과 추론 엔진들 사이의 경제적인 연결

2.1 NNEF 1.0

NNEF 버전 1.0은 잠정적인 버전으로 나왔다가 산업계의 피드백에 근거하여 안정화된 버전으로 공개되었다. 안정화된 버전에서 바뀐 부분은 여러 가지가 있으나 *extent* 타입을 *integer*로 바꾼다든지, 할당문 뒤에 반드시 세미콜론을 오게 한다든지 또는 *squeeze*, *unsqueeze*, *stack*, *unstack*, *slice*, *argmax_reduce*, *prelu*, *RoI* 등과 같은 오퍼레이션들이 추가된 점들 등과 같은 부분들이다. 이에 대한 자세한 내용을 원하는 독자들은 Khronos 그룹의 NNEF Overview¹⁾를 참고하면 된다.

표준 개발자들은 초기 버전에서는 학습된 프레임워크를 임베디드된 추론 엔진으로 보내는 방안에 집중하였다. NNEF 포맷을 소프트웨어 도구 내부로 임포트하는 저장 교환 방안도 많이 떠오르는 유스케이스(use case)였다.

또한 네트워크의 토폴로지를 배치할 수 있도록 지

원하는 것도 중요한 부분이었다. 연구 분야에서 새로운 네트워크의 타입이 등장하면 그것을 포함시킬 수 있도록 NNEF 표준도 빠르게 진화할 수 있어야 했다.

2.2 NNEF – 신경망 파편화 문제

합성곱 신경망(CNN, Convolutional Neural Networks)은 CPU 사용 측면에서 비싸기 때문에 많은 회사들은 적극적으로 모바일과 임베디드 프로세서 구조를 개발하여 고속 및 저전력으로 신경망 기반의 추론 속도를 높이려 하고 있다. 그러한 빠른 발전의 결과로 임베디드 신경망 처리에 관한 시장은 파편화의 위험을 가지게 되었고 여러 개의 플랫폼에 대한 추론 엔진의 환경 변수를 제대로 설정하고 속도를 빠르게 하려는 개발자들에게 방해 요소가 되었다.

오늘날 대부분의 신경망 툴킷들과 추론 엔진들은 자체적인 포맷을 사용하여 신경망 패러미터들을 표현하고 있다. 따라서 [그림 1]과 같이, 자체적인 포맷

1) https://www.khronos.org/nnef#nnef_provisional_to_one_changes

을 가진 신경망을 여러 개의 서로 다른 추론 엔진에서 실행하게 할 수 있도록 임포트(import)하거나 익스포트(export)하기 위한 프로그램들을 필요로 하게 된다.

NNEF는 [그림 2]와 같이 Torch, Caffe, TensorFlow, Theano, Chainer, Caffe2, PyTorch, MXNet과 같은 다양한 신경망 저작 도구들과 추론 엔진들 사이에 경제적이면서도 신뢰성 있는 임포트(import)와 익스포트(export)를 수행할 수 있도록 설계되었다.

NNEF 1.0의 스펙(specification)을 보면 다양한 범위의 유스케이스와 많은 오퍼레이션들을 가지는 네트워크 타입들 그리고 엄밀한 정확성을 보장하기 위하여 이미 널리 사용되고 있는 기존의 컴퓨터 언어들의 문법들을 사용한 확장 가능한 설계 기능까지 커버하고 있다. NNEF는 사용자가 직접 만들 수 있는 복합적인 오퍼레이션들을 정의하는 방법까지 포함하고 있는데 이는 복잡한 신경망 최적화를 수행할 수 있는 여지까지 허용하고 있다는 것이다. 향후 연구에서 NNEF에서는 자체적인 구조를 좀 더 사용자들에게 친숙하고 예측가능하게 구성하려고 하여 NNEF가 배치를 위한 안정적인 플랫폼을 지속적으로 제시하면서도 빠르게 변화하는 머신 러닝 분야를 충분히 잘 따라갈 수 있도록 할 예정이다.

2.3 NNEF 워킹 그룹 참여자들

NNEF에 참여하고 있는 산업계의 참여 주체들 주목할만한 회사들로는 AMD, ARM, cadence, ETRI, 화웨이, 인텔, 엘지, 로스 알라모스 국립 연구소, 퀄컴, 삼성, 소니, 텍사스 인스트루먼트 등을 볼 수 있다.

Khronos는 여러 개의 오픈 소스 프로젝트²⁾를 시작하였는데 이를 살펴보면 NNEF 문법의 파서

(parser)와 유효성 검사기(calidator), 그리고 사례로 넣은 텐서플로와 같은 프레임워크들 중 선택한 프레임워크에 따른 익스포터(explorer) 등이 필요할 수 있다. Khronos는 또한 기존의 머신러닝 커뮤니티의 회사들이 NNEF가 자신들의 워크플로에도 유용할 수 있도록 하기 위해 참여를 환영하고 있다. 더불어 NNEF는 NNEF 파일들을 파싱하고 해석하는 것을 담당하는 Khronos OpenVX™ 워킹그룹과 긴밀하게 작업하고 있다. OpenVX 신경망 확장을 통해 OpenVX 1.2는 컴퓨터 비전 오퍼레이션과 딥 러닝 오퍼레이션을 하나의 그래프에 통합한 크로스 플랫폼 추론 엔진으로 작동할 수 있다.

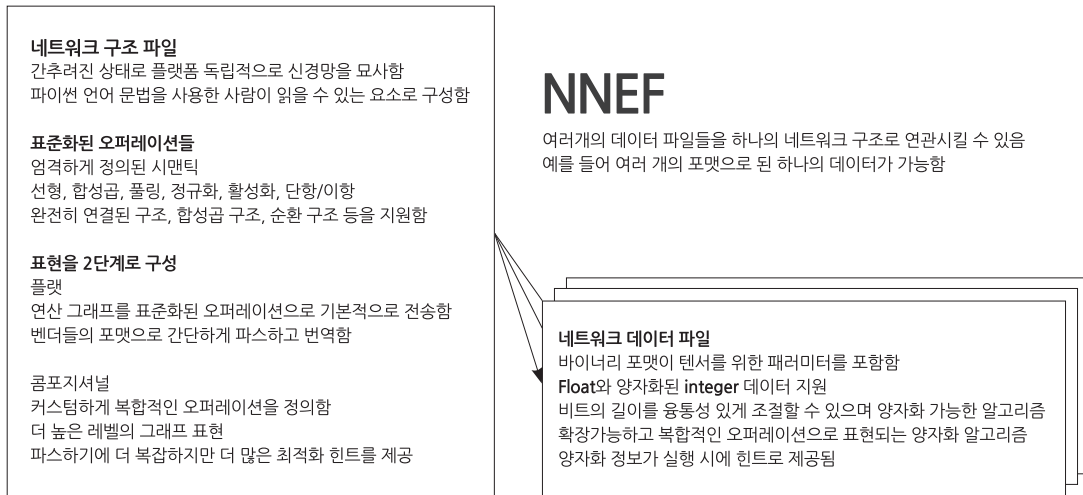
2.4 NNEF 구현(implementation)과 로드맵(roadmap)

현재 활발하게 진행되는 NNEF 로드맵을 보면 다음과 같은 작업들을 고려하고 있다.

- 업체들의 새로운 신경망 타입의 개발을 추적하여 이를 반영함
- 학습된 신경망의 교환 기능을 명확하게 지원함
- 서드 파티 NNEF 도구들을 통해 재교육을 지원함
- 더 많은 종류의 응용 프로그램들을 커버할 수 있도록 함
- NNEF 포맷의 표현력을 높이는 방안을 연구함
- NNEF 오픈 소스 프로젝트 지원

NNEF 관련 소프트웨어들은 Apache 2.0 라이선스로 Khronos의 NNEF GitHub 레포지토리에서 호스트되고 있으며 머신러닝 업체들을 위한 NNEF 컨버터(converter) 개발과 일치 테스트(conformance test)를 위한 관련 RFQ(Request for Quotations) 프로젝트들이 계속 제안되고 있다.

2) <https://github.com/KhronosGroup/NNEF-Tools>



[그림 3] NNEF 포맷

2.5 NNEF 파일 구조

NNEF 포맷을 보면 [그림 3]에서 보듯이 구조와 데이터 파일을 분리하고 있다³⁾.

이러한 포맷은 네트워크 구조 및 각각의 패러미터 데이터에 쉽고 독립적으로 접근하는 방향으로 설계 되었으며 파일들의 집합들은 tar나 zip과 같은 컨테이너 소프트웨어에 압축되거나 암호화되어 저장될 수도 있다.

3. ONNX(오픈 신경망 교환 포맷)

오픈 신경망 교환 포맷(ONNX, Open Neural Network Exchange)⁴⁾은 AI 개발자들이 그들의 프로젝트가 진화하는 것에 대응하기 위한 올바른 도구들을 선택할 수 있도록 하여 AI 개발자들의 능력을 향상시키게 해주는 오픈 생태계 시스템을 구성하기 위한 최초의 시도이다. ONNX는 AI 모델들을 위한 오픈 소스 포맷을 제공한다. ONNX는 확장가능한

컴퓨테이션 그래프 모델(신경망도 이러한 모델임)과 내장된 오퍼레이터들과 표준적인 데이터 타입들의 정의들을 가지고 있다.

ONNX에서는 초창기에는 추론에 필요한 기능들에 초점을 맞춰왔다. Caffe2, PyTorch, Microsoft Cognitive Toolkit, Apache MXNet 등의 소프트웨어 도구들은 ONNX를 지원하고 있다. ONNX는 자신들의 작업을 통해 서로 다른 프레임워크들 간의 상호운용성을 지원하고 연구로부터 생산까지의 개발 프로세스를 지원하여 AI 공동체의 혁신의 스피드를 높이는데 이바지하고자 한다.

3.1 ONNX 스펙(specification)

신경망을 이용한 딥 러닝은 데이터플로우 그래프 상의 연산으로 수행되어진다. 일부 프레임워크들(CNTK, Caffe2, Theano, TensorFlow)은 정적인 그래프를 사용하는 반면 PyTorch나 Chainer와 같은 도구들은 동적인 그래프를 사용한다. 그러나 이러한

3) <https://www.khronos.org/nnef>

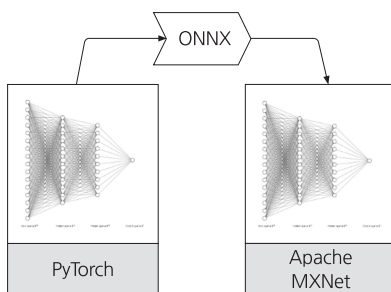
4) <https://github.com/onnx/onnx/blob/master/docs/Overview.md>

도구들은 개발자들이 컴퓨테이션 그래프 및 실행 시스템의 작성과 이들의 최적화된 처리를 용이하게 하도록 하는 인터페이스를 제공한다.

이러한 그래프들은 중간 표현(IR, Intermediate Representation)을 제공하는 데, 이 중간 표현을 이용하여 개발자의 소스 코드에서 개발자의 의도를 파악하여 최적화와 다른 특정 디바이스들(CPU, GPU, FPGA)에서 실행하기 위한 번역(또는 이식)을 수행할 수 있는 것이다.

3.2 공통 IR(common IR)

공통 IR 구성을 한 이유를 알아보면 다음과 같다. 현재 각각의 프레임워크는 자신들만의 그래프 표현을 가지고 있는데 실제로 보면 각각의 프레임워크들은 비슷한 기능들을 가지고 있다. 각 프레임워크들은 API, 그래프, 그리고 실행 시스템들을 가지고 있으며 빠른 학습이나 복잡한 네트워크 구조, 모바일 디바이스에서의 추론 등에 대해 최적화되어 있다. 개발자들은 여러 프레임워크들 중 하나만을 선택하게 된다. 이러한 경향으로 연구 단계와 실제 서비스 단계에서 변환이 필요한 경우가 많고 그로 인해 시간 낭비를 하게 되는 것이다.



[그림 4] ONNX를 통한 변환

AI 환경에 민주적인 경향을 불어넣을 수 있도록 개발자들이 어떠한 작업 단계(개발 단계나 아

니면 배치 단계)에서도 자신들의 프로젝트에 가장 적합한 프레임워크를 선택할 수 있도록 하는 것이 ONNX가 추구하는 것이다. 이렇게 막강한 생태계를 구성하기 위해서 ONNX는 공통의 IR 포맷을 제공한다.


[그림 4]와 같이 컴퓨테이션 그래프에 대한 공통의 표현을 제공함으로써 ONNX는 개발자들로 하여금 자신들의 작업에 가장 올바른 프레임워크를 선택하게 도와주고 혁신적인 향상에만 집중할 수 있게 해준다. 또한 하드웨어를 파는 벤더들에게는 자신들의 플랫폼을 전체 파이프라이닝의 최적화된 일부로 편입시키게 해준다.

아마존 웹 서비스, 페이스북, 마이크로소프트, AMD, 휴렛 팩커드, 텐센트, 인텔, IBM, 화웨이, 맵트랩으로 유명한 매스웍스(MathWorks), 쉘컴, 엔비디아, 유니티, 바이두 등의 기업들이 참여하고 있는 ONNX는 자신들의 제품을 오픈 포맷으로 설계하고 있다. AI 공동체의 누구든 기여할 수 있고 ONNX를 자신들의 생태계에 추가할 수 있는 것이다.

3.3 두 가지 변형(variants)

ONNX는 두 가지 변형을 가지고 있는데 하나는 ONNX이고 다른 하나는 ONNX-ML(machine learning)이다. ONNX에서 기본적으로 정의된 것들은 신경망 기술에 근거한 머신러닝 알고리즘에 대해 필요한 부분들을 지원하는 것이다. ONNX-ML은 전통적인 머신 러닝 알고리즘들에서 사용되는 부가적인 타입들과 표준적인 오퍼레이터들을 가지고 있다. 이 두 가지 변형이 만들어진 이유는 어떤 프레임워크들에서는 ONNX-ML을 통해 표준적인 방식의 신경망 알고리즘 이상의 수준을 원하는 반면 다른 프레임워크들에서는 ONNX를 통해 신경망 수준만 구현하면 충분하기 때문이다.

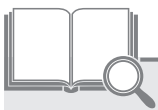
4. 맺음말

본고에서는 대표적인 인공지능망 표준 포맷인 NNEF(Neural Network Exchange Format)와 ONNX(Open Neural Network Exchange)에 대해서 간단히 알아보았다. 향후 딥 러닝 기술이 더욱 활발해지고 개발 공동체의 저변에 퍼진다면 더 세밀한 부분들에 대한 논의가 이루어질 것이고 신경망 외의 다른 머신러닝 기술들도 포용하는 방안들이 더 활발해질 것으로 기대한다. 

※ 이 논문은 2018년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2018R1D1A1A02050166).

[참고문헌]

- [1] NNEF Overview <https://www.khronos.org/nnef>
- [2] ONNX Tutorials <https://github.com/onnx/tutorials>



정보통신 용어 사전

<http://terms.tta.or.kr>

✓ 튜링 시험 Turing test

컴퓨터가 지능이 있는지를 판별할 수 있는 시험.

조사관이 블라인드 상태에서 컴퓨터와 글로 대화를 나눈 후, 대화 상대가 사람인지 컴퓨터인지 판단할 수 없게 되면 컴퓨터는 지능이 있는 것으로 튜링 시험에 합격 판정을 받는다.

앨런 튜링(Alan Turing)이 1950년 컴퓨터 지능의 운영 정의(operational definition)를 위해 튜링 시험을 고안하여 논문에서 제안하였다(※논문: Turing, A.M., Computing machinery and intelligence, Mind, 59(236):433-460, 1950).

튜링은 컴퓨터가 사람처럼 신체 동작하는 것은 지능과 관련이 없다는 전제하에, 사람이 사용하는 언어, 즉 자연어를 이용한 시험을 택하였다. 최초의 튜링 시험에서는 사람인 조사관이 블라인드 상태에서 컴퓨터와 5분간 채팅하여 상대가 컴퓨터인지 사람인지 판단한다. 여러 조사관들 중 상대가 사람일 것이라고 판단하는 조사관이 30% 이상이면 컴퓨터는 튜링 시험 합격이다. 기본 과정에 영상과 작은 연결 통로로 물건을 주고받는 동작 시험을 포함시킨 전체 튜링 시험(total Turing test)도 있다.

튜링은 시험을 통과하는 컴퓨터가 2000년이 되기 전에 개발될 것이라고 예상하였으나 아직 튜링 시험을 완전히 통과한 컴퓨터는 개발되지 못하였다. 튜링 시험이 제안된 이래 컴퓨터가 실제로 지능을 가질 수 있는지에 대해서는 다양한 논란이 있다.