

# 분산 딥러닝 모델 개발을 위한 고수준 분석 플랫폼

박경석\*, \*\*, 유찬희\* \*\*, Komal Sarda\* \*\*, 엄정호\*  
 \*한국과학기술정보연구원  
 \*\*UST 빅데이터과학과  
 gspark@kisti.re.kr

## High-level Analytics Platform for Development of Distributed Deep Learning Model

Kyongseok Park\*, \*\*, Chan Hee Yu\* \*\*, Komal Sarda\* \*\*, Jung-Ho Um\*  
 \*Korea Institute of Science and Technology Information  
 \*\*Department of Big Data Science, UST

### 요 약

딥러닝(deep learning)은 기계학습 알고리즘 중 가장 널리 활용되고 있는 알고리즘이다. 딥러닝 기술은 산업, 과학, 국방 및 공공 부문을 비롯하여 거의 모든 분야에서 폭넓게 확산되고 있다. 그러나 기계학습 기술에 대한 이해와 프로그래밍 지식이 부족할 경우 자유롭게 활용하는 데는 제약이 따르고 있으며 빅데이터를 활용하여 일반 이용자들이 직접 분산 학습 모델을 개발하고 배포하는 데 어려움이 발생하고 있다. 이러한 요구를 충족시키기 위해 딥러닝 프레임워크의 저수준 API를 추상화하여 고수준 분석과 분산 딥러닝을 지원하고 일반 이용자들이 실무적으로 복잡한 딥러닝 기술을 활용할 수 있는 기술을 개발하였다. 플랫폼 개발과 함께 중요하게 고려해야 하는 요소 중 하나로 플랫폼의 배포와 확장성 역시 고려되어야 한다. 본 플랫폼은 조직 내 계산 자원을 이용하여 플랫폼을 배포할 수 있으며 상용 클라우드 서비스와 연동하여 배포할 수 있도록 설계됨에 따라 환경의 제약 없이 유연한 서비스 제공이 가능하다.

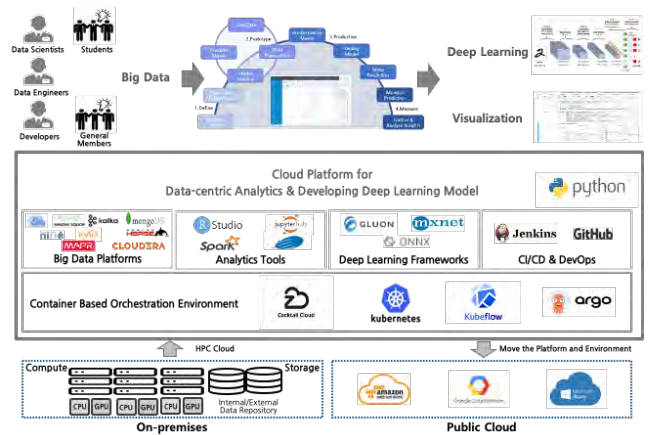
### 1. 서론

심층신경망(deep learning)은 높은 정확도와 실용성으로 다양한 분야에서 널리 활용되고 있다. 현재 심층신경망 모델을 개발하기 위해 수많은 프레임워크가 제공되고 있으며 개발자들은 대부분 이러한 프레임워크를 이용하여 모델을 개발한다. 개발자들 사이에서 널리 사용되고 있는 프레임워크는 TensorFlow, MXNet, Caffe, Theano, PyTorch, DL4J가 있으며 이러한 프레임워크를 기반으로 R, Matlab, SAS 등의 전통적인 분석 시스템과 융합한 기술도 활용되고 있다. 이러한 기술은 주로 Python, R, C, Java와 같은 언어를 활용하여 모델을 개발하기 때문에 프로그래밍 기술이 부족한 이용자들은 모델 개발과 활용에 어려움을 느끼고 있다. KNIME을 비롯하여 몇몇 시스템과 AWS, Azure 등의 상용 클라우드 서비스에서도 GUI 기반으로 딥러닝 모델 개발을 지원하지만 상업적 시스템으로 접근과 활용에 제약이 따른다.

빅데이터를 활용하여 모델을 구축하기 위해서는 다수의 계산 노드와 GPU를 활용하여 분산 학습을 수행해야 한다. 그러나 일반 이용자들이 공개된 프레임워

크를 이용하여 복잡한 분산 학습을 수행하기에는 어려움이 있다. 이 연구에서는 복잡한 심층신경망 모델 개발 시 저수준 API(low-level API)를 계산 블록 단위로 추상화하고 웹에서 블록 기반의 고수준(high-level API) 분석과 분산 학습을 지원하는 기술을 통해 복잡한 딥러닝 모델을 보다 간결하고 빠르게 개발하고 배포할 수 있도록 하였다.

### 2. 플랫폼 아키텍처 및 기술 구성



(그림 1) 플랫폼 아키텍처

개발한 기술은 빅데이터 분석과 딥러닝 모형 구현에 활용하기 위한 플랫폼이다. 빅데이터를 처리하고 분석하기 위해서는 Hadoop이나 Spark 등의 대용량 데이터 처리 기술이 필요하며, 실시간 빅데이터 처리를 위해서는 Kafka와 같은 대용량 메시지를 처리하기 위한 기술이 필요하다. 유연한 자원 활용과 확장성을 위해서는 가상화 기술과 컨테이너 기술이 요구되며 Kubernetes나 Kubeflow 등을 기반으로 하는 빅데이터 및 기계학습 모형 개발이 가능해야 한다.

딥러닝 모형 개발은 분산 학습을 위한 확장성(scale-out)과 코드 블록(code block)을 재활용하기 쉬워야 하기 때문에 MXNet과 GLUON을 적용하였다. 또한 다양한 딥러닝 모형을 상호 교환하기 위해 ONNX(open neural network exchange) 기반의 모형 교환 기술을 적용하였다. ONNX를 통해 이용자들은 이질적 프레임워크로 모형을 배포하거나 본 플랫폼으로 모형을 호출하여 학습과 추론(inference)를 수행할 수 있다.

플랫폼을 구축하고 운영하기 위해서는 이용자들이 조직 내에서 자체적으로 구축한 자원과 AWS, Azure, Google Cloud Platform 등의 클라우드 환경을 동시에 활용하고 연동할 수 있어야 한다. 따라서 모든 기술을 컨테이너나 가상머신에 구축하고 이를 외부 클라우드 서비스와 연동할 수 있도록 설계하였다.

### 3. 분산 딥러닝 모형 개발 환경



(그림 2) 모형 개발 환경

딥러닝 모형은 각각의 계산 단위로 구성된 블록을 연결하여 개발할 수 있다. 각 계산 블록에서는 저수준 API에서 요구하는 인자(argument)를 입력할 수 있으며 다수의 저수준 API를 하나의 블록에 통합하였다. 모형 개발은 데이터 처리부터 모형 검증 및 배포에 이르기까지 블록의 연결과 실행을 통해 이루어진

다. 개발한 플랫폼에서는 단순한 딥러닝 모형부터 복잡하고 전문적 수준의 모형까지 직관적으로 개발할 수 있다. 이용자들은 ResNet, DenseNet, VGG 등의 사전 학습된 모델을 이용하여 자신의 데이터에 적합하게 전이학습(transfer learning)을 수행할 수 있다 [1]. 플랫폼의 기본 프레임워크는 MXNet과 GLUON으로 이루어졌기 때문에 이용자들이 다른 프레임워크에서 개발한 모형을 본 플랫폼에 적용하기 위해서는 이용자 코드를 적용하고 실행할 수 있어야 한다. 이를 위해 사용자 정의 모델(user defined model)을 호출하고 실행할 수 있도록 하였다.



(그림 3) 딥러닝 모형 개발 흐름

빅데이터를 기반으로 모형을 개발하기 위해서는 분산 학습 기술이 필요하다. 분산 학습은 크게 데이터 병렬화(data parallelism)와 모형 병렬화(model parallelism)로 구분한다[2, 3, 4]. 일반적으로 빅데이터를 이용한 학습은 데이터 병렬화에 해당한다[4]. 데이터 병렬화를 이용한 딥러닝 모형 구축은 다수의 계산 자원에 데이터를 분할하여 할당하고 각 자원 별로 단계 별 학습을 수행한 후 파라미터를 공유하거나 갱신하는 방식으로 이루어진다. 파라미터 또는 가중치 갱신은 일반적으로 동기식 업데이트(synchronous update) 방식을 사용하지만 경우에 따라 비동기식 업데이트(asynchronous update) 방식을 사용하기도 한다. 때로는 이 두 가지 방식을 혼용(hybrid update)하여 사용할 수도 있다. 파라미터 갱신은 일반적으로 단일 또는 다수의 파라미터 서버를 설정하고 다수의 계산 노드(worker node)와 통신하며 파라미터를 서로 공유하고 갱신하게 된다. 이 과정에서 필연적으로 상당한 통신비용이 발생하게 된다.

본 플랫폼에서는 MXNet을 기본 프레임워크로 사용하고 있지만 MXNet에 구현된 자체 분산 학습 기술인 파라미터 서버 방식의 가중치 갱신 방식을 사용하지 않고 Horovod 기반의 분산 학습 기술을 적용

하였다. 이미 언급한 바와 같이 파라미터 서버 방식의 기술은 업데이트해야 하는 파라미터가 많을 경우 네트워크 병목으로 인한 성능 저하가 발생한다. Horovod의 경우 대규모 클러스터에서 파라미터 업데이트에 적합한 기술로 데이터의 증가와 계산 자원의 증가(scale-out)로 인한 성능 저하를 줄일 수 있다[5, 6]. 업데이트에 필요한 파라미터의 규모는 데이터의 양과 모형의 복잡도에 비례해서 증가하기 때문에 빅데이터를 활용하여 모형을 개발할 경우 분산 학습 기술의 필요성이 더욱 높아지게 된다. 그러나 분산 학습을 위해서는 분산 컴퓨팅이나 병렬 컴퓨팅에 대한 지식과 경험이 필요하며 이러한 이유로 대부분의 이용자들은 분산 학습에 어려움을 느끼고 있다. 본 플랫폼을 사용할 경우 이용자들은 분산 학습에 대한 전문적 지식을 필요로 하지 않으며 단지 원하는 계산 자원의 유형과 규모를 선택하여 학습을 수행하면 되기 때문에 모형 개발에 집중할 수 있다.

이 연구에서 제안하는 플랫폼에서는 이용자의 요청에 따라 가용한 자원을 할당하고 학습 과정을 수행하며 진행 상황을 모니터링 하고 관리할 수 있는 환경을 제공한다. 다수의 사용자가 단일 시스템을 동시에 이용하기 때문에 가용한 자원을 확인하고 자원 할당 정보를 공유할 수 있도록 관리 기능을 구현하였다. 다른 사용자의 자원 점유로 요청한 자원에 여유가 없을 경우 스케줄러(scheduler)의 큐(queue)에 작업을 할당하고 가용 자원이 존재할 때 이용자가 요청한 자원과 작업을 처리하게 된다.

#### 4. 결론

본 연구에서 개발한 플랫폼은 기계학습 전문가부터 모형 개발에 필요한 기초 개념을 이해한 일반 사용자까지 빠르고 직관적으로 복잡한 딥러닝 모형을 개발할 수 있도록 지원한다. 또한 분산 컴퓨팅에 대한 전문 지식과 경험 없이 빅데이터를 활용하여 고성능의 분산 학습을 수행할 수 있는 기술을 제공하고 있다. 이에 따라 이용자들은 모형 개발에만 집중할 수 있고 복잡한 분산 컴퓨팅 기술을 이해하는 데 필요한 시간을 절약할 수 있다. 플랫폼 배포와 관리를 클라우드 환경에 적용할 수 있기 때문에 보다 폭넓은 활용과 유연한 서비스 제공이 가능하다. 아울러 작업 관리와 자원 할당 기능을 제공하기 때문에 다수의 사용자가 동시에 플랫폼을 사용할 경우 효율적 자원 분배와 공유가 가능하다.

#### 사사

이 논문은 산업통상자원부의 재원으로 한국에너지기술평가원(KETEP)의 지원을 받아 수행한 연구입니다. (No. 20181110100420)

#### 참고문헌

- [1] Pratt, L. Y., "Discriminability-based transfer between neural networks", NIPS, Denver, 1993, pp. 204-211
- [2] Dean, J., Corrado, G. S., Monga, R., Chen, K., Devin, M., Le, Q. V., Mao, M. Z., Ranzato, M., Senior, A., Tucker, P., Yang, K., and Ng, A. Y., "Large scale distributed deep networks", NIPS, Nevada, 2012, pp. 1223-1231
- [3] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., and Zheng, X., "Tensorflow: A system for large-scale machine learning", OSDI, Savannah, 2016, pp. 265-283
- [4] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "ImageNet classification with deep convolutional neural networks", NIPS, Nevada, 2012, pp. 1097-1105
- [5] Alexander Sergeev and Mike Del Balso, "Horovod: Fast and Easy Distributed Deep Learning in TensorFlow", arXiv preprint arXiv:1802.05799, 2018
- [6] Yanghua Peng, Yibo Zhu, Yangrui Chen, Yixin Bao, Bairen Yi, Chang Lan, Chuan Wu, and Chuanxiong Guo, "A Generic Communication Scheduler for Distributed DNN Training Acceleration", SOAP, Ontario, 2019, pp. 16-29