2023 대한임베디드공학회 추계학술대회

"Embedded AI for Industrial Application"

일 자 : 2023년 11월 8일(수) ~ 11월 11일(토)

장 소 : 제주 그라벨 호텔

주최 | IEMEK (사)대한임베디드공학회 Institute of Embedded Engineering of Korea

주관 | ETRI 한국전자통신연구원 Electronics and Telecommunications Research Institute, DGIST 대구경북과학기술원 Daegu Gyeongbuk Institute of Science & Technology, GITC (재)경북IT융합산업기술원 Gyeongbuk Institute of IT Convergence Industry Technology, Yeungnam University 정보통신연구소, 대구대학교 DAEGU UNIVERSITY, 경북대학교 KYUNGPOOK NATIONAL UNIVERSITY, KIAPI 지능형자동차부품진흥원 Korea Intelligent Automotive Parts Promotion Institute, TTA 한국정보통신기술협회 Telecommunications Technology Association, SJTP (재)세종테크노파크, erae, MORAI

후원 | SK broadband, kt, HUCONN, JIN INFRA, (주)담비, Root lab, BOTO, BRIGHTEN, JMON, UENS, ㈜티에이씽크, 경북자동차임베디드연구원 GIAET Gyeongbuk Research Institute of Vehicle Embedded Technology, 사단법인 대경ICT산업협회 DGICT, emotion inspire emotion within

엣지 장치 기반 딥 러닝 분산 학습 성능 평가

(Performance Evaluation of Deep Learning Distributed Training on Edge Devices)

박 시 형[†*], 이 제 민[‡], 황 태 호[†]
[†]한국전자기술연구원, [‡]한국전자통신연구원
(Sihyeong Park, Jemin Lee, Taeho Hwang)
([†]Korea Electronics Technology Institute, [‡]Electronics and Telecommunications Research Institute)

Abstract : 엣지 장치에서도 딥 러닝 모델 학습의 필요성이 증가함에 따라 경량 모델과 다수의 엣지 장치를 사용한 분산 학습이 연구되고 있다. 본 논문에서는 NVIDIA Jetson AGX Orin 보드를 사용해 MobileNet v2를 CIFAR-100 데이터세트로 전이 학습 기반 분산 학습의 성능 평가를 하였다. 이를 위해 2개의 보드를 사용해 모델의 학습 과정에서의 정확도, 손실 및 시간을 분석하였다. 실험을 통해 분산 학습에 사용되는 장치 수에 따라 파라미터 동기화 등의 오버헤드로 학습 시간이 증가할 수 있음을 보였다.

Keywords : edge devices, deep learning, distributed training, transfer learning, performance

Ⅰ. 서 론

엣지 장치는 연산 성능과 배터리의 제한으로 인해 딥 러닝 연산을 서버로 오프로딩 (offloading) 하는 방식을 사용하였다 [1]. 하지만 서버 오프로딩은 장치에서 수집한 데이터를 서버로 보내야 하고 네트워크 상태에 따라 처리 시간이 지연될 수도 있다.

개인 정보 보호의 중요성이 높아짐에 따라 엣지 장치에서 온 디바이스 학습의 필요성이 증가하고 있다. 이를 위해 MobileNet [2] 등과 같은 모바일 환경을 위한 경량 딥 러닝 모델 구조와 엣지 장치에서 학습을 위한 방법 [3] 등이 연구되었다. 반면 서버 환경에서의 딥 러닝 모델 학습은 여러 서버 혹은 다수의 Graphics Processing Unit (GPU)을 이용한 분산 학습 [4, 5]을 통해 규모가 큰 모델을 빠르게 학습하는 방식을 사용한다.

본 논문에서는 엣지와 같은 모바일 환경을 위한 경량 모델을 사용해 분산 학습을 하였다. 이를 위해 NVIDIA Jetson AGX Orin으로 분산 학습 환경을 구축하고 CIFAR-100 데이터세트와 MobileNet v2 모델로 Tensorflow (Keras)를 통해 전이 학습 [6]으로 모델을 학습하고 성능을 측정하였다.

*Corresponding Author (sihyeong@keti.re.kr)
박시형, 황태호: 한국전자기술연구원
이제민: 한국전자통신연구원

※ 이 연구는 2023년도 산업통상자원부 및 산업기술평가관리원 (KEIT) 연구비 지원에 의한 연구임 ('20009972', 고효율 초저전력 경량 엣지 디바이스용 소자회로 및 SoC 개발)

Ⅱ. 배경지식

딥 러닝 모델 학습 및 검증에 널리 사용하는 Tensorflow와 PyTorch 등의 프레임워크들은 분산 학습을 위한 기능을 제공하고 있다. 분산 학습은 크게 모델을 여러 부분으로 나눠 다수의 장치 (혹은 서버)에서 학습하는 방식과 같은 모델을 다수의 장치에서 학습하는 방식이 있다.

PyTorch는 Distributed Data Parallel (DDP), RPC-Based Distributed Training (RPC), Collective Communication (c10d) 등의 분산 학습 방법을 제공한다 [4]. DDP는 단일 프로그램 멀티 데이터 학습 방법으로 모델을 모든 프로세스에 복제해 다른 데이터세트로 학습하는 방식이다. RPC는 분산 파이프라인 병렬화, 파라미터 서버 등을 지원하는 분산 학습 방식이며, c10d는 DDP와 RPC를 기반으로 그룹 내 프로세서 간 텐서 전송을 지원하는 방법이다.

Tensorflow는 Mirrored Strategy, TPU Strategy, Multi Worker Mirrored Strategy, Central Storage Strategy, Parameter Server Strategy 등의 분산 학습 방식을 제공한다 [7]. Mirrored Strategy는 하나의 장치에서 다수의 GPU로 동기식 학습 방식이며 TPU Strategy는 Google

의 Tensor Processing Units (TPU)를 지원하는 학습 방식이다. Multi Worker Mirrored Strategy는 Mirrored Strategy와 유사하지만 여러 GPU가 있는 다수의 장치에서 동기식 학습을 지원한다. 이를 위해 모든 장치에 같은 모델을 사용하고, 각 장치의 모든 변수의 사본을 생성한다. Parameter Server Strategy는 PyTorch의 RPC와 유사한 방식으로 여러 장치에서 모델 학습한 매개변수를 매개변수 서버로 전송해서 업데이트 하는 방식이다.

본 논문에서는 Keras로 Multi Worker Mirrored Strategy를 사용해 2개의 엣지 장치를 사용하여 모델을 분산 학습 하였다.

Ⅲ. 실험 환경 및 결과

1. 실험 환경

본 논문에서는 NVIDIA Jetson AGX Orin을 사용해 분산 학습을 수행하였다. 실험 환경은 표 1과 같다. 실험에서는 최대 2대의 장치를 사용하였으며 각 장치는 1Gbps 유선 네트워크 스위치를 통해 연결되어 있다.

분산 학습을 위해 CIFAR-100 데이터세트[1]를 사용한다. CIFAR-100은 100개의 클래스로 이루어지며 각 클래스당 600개의 32×32 이미지로 구성된다. 또한, Keras에서 제공하는 MobileNet v2 모델을 백본 (backbone)으로 사용하고 ImageNet 데이터세트로 사전 학습된 가중치를 바탕으로 전이 학습하였다. 클래스 분류를 위해 백본 네트워크에서 GlobalAveragePooling2D, Dense (512, relu), Dropout (0.3), Dense (100, softmax) 레이어를 추가하였다. 학습을 위해 손실 함수는 categorical cross entropy를 사용하였고, 10 에포크 (epoch)부터는 Exponential Decay로 학습률 (learning rate)을 조정하였다.

표 1. 실험 환경
Table 1. Experimental Environments

		Specification
HW	CPU	ARM Cortex-A78AE (12-core)
	Memory	32 GB (CPU, GPU shared)
	GPU	NVIDIA Ampere GPU (1792-core)
SW	OS	Ubuntu 20.04 (JetPack 5.1.1)
	CUDA	11.4 (pycuda 2022.2.2)
	Python	3.8.10

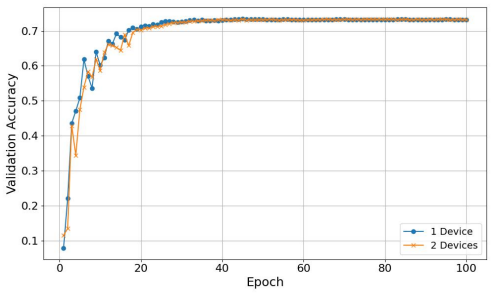

그림 1. 모델 학습 정확도
Fig. 1. Model Training Accuracy

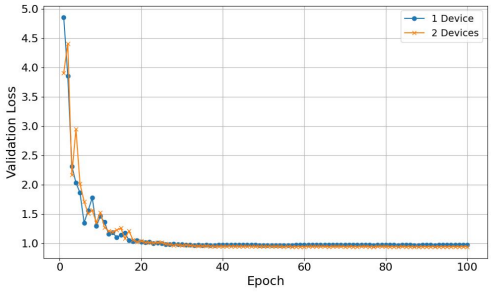

그림 2. 모델 학습 손실
Fig. 2. Model Training Loss

전이 학습을 위해 엣지 장치의 메모리를 고려해 배치는 64로 설정하고 정확도 향상을 위해 데이터세트의 이미지를 64×64로 변환해서 사용하였다. 학습은 100 에포크 동안 진행하였다. 분산 학습 과정에서 장치 간 통신은 NVIDIA NCCL 라이브러리를 사용하도록 설정하였다.

2 실험 결과

NVIDIA Jetson을 사용한 분산 학습의 성능을 비교하기 위해 1개의 장치를 사용하였을 때와 2개의 장치를 사용할 때의 학습 결과를 비교하였다.

그림 1은 학습 과정에서의 검증 (validation) 정확도를 나타낸다. 1 에포크 학습 이후 1개의 장치의 정확도는 0.08이었지만, 2개의 장치를 사용하면 0.12로 0.04의 차이가 발생하였다. 하지만 나머지 에포크에서는 대체로 1개의 장치만 사용하여 학습할 때 더 높은 정확도를 보였다. 학습 과정에서의 손실은 그림 2와 같다. 1개의 장치를 사용해서 학습하였을 때는 2개의 장치를 사용한 경우보다 빠르

1) https://www.cs.toronto.edu/~kriz/cifar.html

게 감소하였다. 특히, 학습을 위해서 1개의 장치만 사용하면 전체 학습 시간이 9,310초 소요되었지만 2개의 장치를 사용하면 28,248초로 약 3배 이상 증가하였다.

실험 결과를 바탕으로 프레임워크에서 기본적으로 제공하는 설정을 바탕으로 분산 학습을 수행하면 장치의 수가 증가하더라도 장치 간 통신, 동기화로 인해 학습 속도가 증가하였다. 특히, 실험을 위해 장치별로 같은 모델과 데이터세트를 사용하여서 정확도 향상에 대한 이점을 가지지 못하였다. 또한, 분산 학습 과정에서 최적화된 손실 함수에 대한 고려가 필요함을 보였다.

Ⅳ. 결론 및 향후 연구

본 논문에서는 엣지 환경에서 널리 사용되는 NVIDIA Jetson 보드를 사용해서 분산 학습을 하였다. 이를 위해 Tensorflow (Keras)에서 ImageNet 데이터세트를 사용해 사전 학습된 모델을 바탕으로 CIFAR-100 데이터세트로 전이 학습할 때의 성능을 측정하였다. 결과를 통해 분산 학습을 위해 통신 및 모델 구조 최적화가 이루어지지 않으면 장치의 수가 증가하면 동기화 등의 오버헤드로 인해 학습 시간의 증가가 발생함을 보였다.

향후 연구로 엣지 환경에서 적합한 경량 모델 구조와 이를 이용한 분산 학습 방법에 관한 연구를 수행할 계획이다. 이를 통해 분산 학습에 사용되는 장치의 수를 증가시켜 성능 분석을 수행할 계획이다. 특히, 엣지 장치의 하드웨어 자원 사용을 최적화하고 장치 간 전송 오버헤드를 줄일 방법을 연구한다.

References

[1] A. Alelaiwi, “An efficient method of computation offloading in an edge cloud platform”. Journal of Parallel and Distributed Computing, Vol. 127, pp.58-64, 2019.

[2] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks”. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.4510-4520, 2018.

[3] H. Cai, C. Gan, L. Zhu, S. Han, “TinyTL: Reduce Memory, Not Parameters for Efficient On-Device Learning”. Advances in Neural Information Processing Systems 33 (NeurIPS 2020), pp.11285-11297, 2020.

[4] S. Li, Y. Zhao, R. Varma, O. Salpekar, P. Noordhuis, T. Li, A. Paszke, J. Smith, B. Vaughan, P. Damania, and S. Chintala, “PyTorch Distributed: Experiences on Accelerating Data Parallel Training”. Proceedings of the VLDB Endowment, Vol. 13, No. 12, pp.3005-3018, 2020.

[5] A. Or, H. Zhang, and M.J. Freedman, “Resource elasticity in distributed deep learning”. Proceedings of Machine Learning and Systems 2, pp.400-411, 2020.

[6] J. Talukdar, S. Gupta, P.S. Rajpura, and R.S. Hegde, “Transfer Learning for Object Detection using State-of-the-Art Deep Neural Networks”. 2018 5th international conference on signal processing and integrated networks (SPIN), pp.78-83, 2018.

[7] G. Ponnuswami, S. Kailasam, and D.A. Dinesh, “valuating Data-Parallel Distributed Training Strategies”. 2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS), pp.759-763, 2022.