Chapter 01

AI 반도체를 위한 신경망 학습용 데이터 포맷기술 동향

김혜지_한국전자통신연구원 선임연구원

AI 반도체는 인공신경망 연산을 메모리 효율적이면서 고속으로 처리하는 하드웨어 처리장치로써, 전용의 최적화된 컴파일러 및 소프트웨어 라이브러리를 통해 AI 시스템으로 완성되는 핵심 기반 기술이다. 최근 인공신경망 학습 규모가 해마다 10배씩 상승함에 따라 학습을 위한 메모리 사용량과 AI 반도체의 개수도 점차 증가하고 있다. 신경망의 막대한 학습 비용을 현실화하기 위해 연산을 빠르고 효율적으로 처리하면서 인식 정확도의 손실을 최소화하는 방법으로 부동소수점 기반 데이터 양자화 포맷 기술이 발전하고 있다. 본 고는 AI 반도체 산업계를 중심으로 발전하는 저정밀도 데이터 포맷 기술의 표준화 및 상용화 현황을 알아보고 저정밀도 데이터 포맷의 상세 구조 및 관련 알고리즘의 최신 연구 흐름을 살펴본다.

I. 서론

2012년 ILSVRC[1] 객체인식대회에서 인공신경망을 사용한 AlexNet이 월등한 성능으로 우승하여 관련 연구자들의 이목을 집중시킨 지 10년을 지나고 있다. 지난 10년 동안 인공신경망 연구는 자율주행, 화질 개선, 추천시스템, 챗봇, 이미지 생성 등 다양한 산업 분야에 적용되어 우리 주변에 가깝게 다가왔다. 이러한 AI 연구와 산업의 비약적 발전 배경에 다양한 요소가 있겠지만 AI 반도체 성능의 발전이 그 중심에 있을 것이다. GPT와 같은 자연어처리 인공신경망의 규모는 해를 거듭하며 10배씩 거대해 짐에 따라 막대한 양의 학습 데이터를 저장하고 처리하기 위해 AI 반도체의 내장 메모리는 더 빠르고 고용량을 지원하며 연산 병렬성이 높아지도록 발전하고 있다[2]. 그러나 반도체 공정과 메모리 및 연산기의 성능 향상만으로 고속 성장 중인 인공지능 분야를 감당하기에는 막대한 자본이 요구된다. 따라서

* 본 내용은 김혜지 선임연구원(☎ 042-860-5379, hyejikim@etri.re.kr)에게 문의하시기 바랍니다.
** 본 내용은 필자의 주관적인 의견이며 IITP의 공식적인 입장이 아님을 밝힙니다.
** 이 기고문은 2022년도 과학기술정보통신부의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구입니다. (No.2022-0-00018, 거대 인공지능 학습을 위한 K-인공두뇌 반도체 개발)

AI 반도체의 혁신과 더불어 인공 신경망을 효율적으로 처리하기 위한 연산 최적화 및 신경망 경량화 소프트웨어 알고리즘의 연구가 필연적으로 발전하고 있다. 특히, 인공신경망 데이터를 함축적으로 표현하면서, AI 가속기에 친화적인 전용의 데이터 포맷을 통해 반도체의 집적도와 메모리 재사용을 높이는 SW-HW 통합 연구가 중요하다.

인공신경망의 학습에 필요한 파라미터의 양은 무수히 많고, 그 중에는 원하는 정확도 성능을 달성하는 데 거의 영향을 끼치지 않는 불필요한 학습 파라미터가 다수 포함되어 있다. 이러한 사실에 기반하여 양자화(quantization), 가지치기(pruning), 전이학습(knowledge distillation)과 같은 신경망의 효율적인 학습 및 추론을 위한 경량화 연구가 활발히 진행되고 있다. AI 반도체에 부동소수점 8bit 연산기가 추가되는 것은 신경망 양자화 연구 분야와 관련된다. 신경망 경량화를 위한 데이터 양자화 기술은 8bit 이하의 정수형 또는 부동소수점 데이터 포맷으로 신경망 학습 및 추론에 필요한 메모리 사용량을 줄이고 연산을 가속하는 데이터 표현 방법 및 알고리즘에 관한 분야이다. 신경망의 연산 과정, 특히 행렬연산에서 데이터를 32bit로 표현하지 않고 부동소수점 8bit로 표현해도 종래의 32bit와 유사한 정확도로 신경망을 학습할 수 있다[3],[4].

AI 반도체가 8bit 부동소수점 데이터 포맷을 지원하면서 관련 하드웨어 연산기능을 제공하는 경우, 행렬 연산기는 종래의 32bit 데이터 1개를 입력받을 수 있던 것에 비해 8bit 데이터 4개를 한 번에 입력받을 수 있으므로 구조에 따라 이론적으로 최대 16배까지 처리량을 상승시킬 수 있다. 다른 의미로, 8bit 양자화 기능은 같은 메모리 대역폭에 더 많은 행렬의 원소를 담을 수 있으므로 연산기 개수의 증가가 시스템의 최대 처리량에 직접적인 상승 효과를 가져 온다. 그에 따라 최근 NVIDIA H100 GPU와 Tesla Dojo D1 Chip은 부동소수점 기반 8bit 연산기를 포함하여 효율적인 신경망 학습 및 추론 기능을 지원하고 있다[2],[5]. 국내의 AI 반도체 기업 중 FuriosaAI의 WARBOY Chip은 추론 연산에서 높은 에너지 효율과 처리속도를 위해 정수형 8bit 데이터 타입을 지원하고 있다.

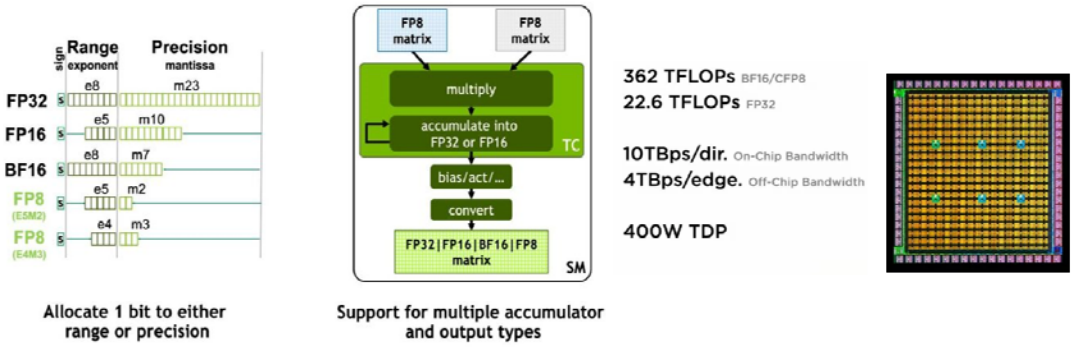
II. 저정밀도 부동소수점 데이터 포맷 기술의 상용화 동향

인공신경망 구조를 정의하고 학습 알고리즘을 개발하며 신경망을 학습하는 과정 모두에 AI 반도체가 필요하다. NVIDIA는 2017년 텐서 코어(Tensor Core)가 포함된 볼타(Volta)

아키텍처를 공개하면서 본격적으로 AI 반도체의 상용화 시대를 열었다. 과거의 NVIDIA GPU는 게이밍 그래픽스 연산에 최적화된 아키텍처로 구성되던 것에 비해 볼타는 텐서 코어를 탑재하여 인공신경망의 핵심인 행렬연산을 가속하는 데 집중했다[6]. 또한, CuDNN, TensorRT 등 인공신경망 연산에 특화된 연산 라이브러리를 Pytorch, Tensorflow와 같은 신경망 학습 프레임워크에 연동하여 AI 연구 및 개발에 편의적인 환경을 제공하고 있다. 텐서 코어는 인공신경망을 가속하는 전용 SW 라이브러리와 함께 지속적인 발전을 거듭하여 종래의 부동소수점 32bit(FP32) 데이터 포맷보다 간소화된 부동소수점 19bit(TF32)와 16bit (FP16, BF16) 데이터형을 처리하면서 더욱 빠른 신경망 학습 및 추론 속도를 달성하고자 하였다[7].

1. NVIDIA H100

2022년 NVIDIA는 저정밀도 부동소수점 8bit(FP8) 연산 기능을 지원하는 Hopper 아키텍처 기반의 H100 GPU를 공개했다([그림 1] 참조)[2]. 해당 칩은 범용 AI연구 및 신경망 학습에 사용되며 2020년에 공개된 Ampere 아키텍처의 A100 GPU와 비교하여 자연어처리 모델 GPT3에서 약 6배 빠른 학습 속도를 달성했다. FP8 데이터 포맷은 부동소수점의 지수(exponent)와 가수(mantissa)의 bitwidth에 따라 두 가지 버전의 포맷(E4M3, E5M2)으로 구성된다. 해당 포맷을 사용하면 전작의 A100 GPU의 FP16 연산 대비 약 6.4배 빠른 처리속도를 보인다. 특히, 트랜스포머 엔진을 통해 인공신경망의 레이어마다 FP8 포맷의



[FP8: NVIDIA H100 Tensor Core] [CFP8: Tesla Dojo D1]

[그림 1] FP8이 적용된 상용 AI 반도체

데이터 표현 범위를 동적으로 조절하도록 하여 실제 학습 과정에서 필요한 데이터 형 변환 및 동적 바이어스(bias) 튜닝 등에 대한 연산을 하드웨어 상에서 처리하여 부가적인 지연시간을 최소화하도록 구성했다.

2. Tesla Dojo D1

2021년 테슬라에서 자사의 자율주행자동차를 위한 AI 반도체 Dojo D1 칩을 발표했다[5]. 해당 칩은 자율주행에 필요한 인공신경망의 학습 및 비디오 처리시스템에 최적화된 하드웨어 구조를 가지고 있다. 특히, CFP8(Configurable Floating-Point 8bit) 저정밀도 데이터 포맷을 지원하여 FP32 포맷 대비 약 16배 빠른 연산이 가능하다. CFP8은 부동소수점에 기반하며, NVIDIA H100과 마찬가지로 Exponent(E)와 Mantissa(M)의 bitwidth에 따라 두 가지 종류의 포맷(E4M3, E5M2)을 동시 지원한다([그림 1] 참조). Exponent는 6bit의 바이어스 정보를 별도로 가지고 있어서 해당 값을 통해 데이터의 표현범위를 조절하여 신경망 연산 과정에서 8bit 연산에 의한 정보 손실을 최소화 하는데 사용된다.

3. 부동소수점 8bit 데이터 포맷의 표준화

산업계는 8bit 저정밀도 부동소수점 포맷의 개방형 라이선스를 획득하여 사용자들의 광범위한 활용을 권장하기 위해 IEEE 표준화를 위한 논의를 시작했다. NVIDIA, ARM, Intel은 복합 FP8(E4M3, E5M2) 타입으로 CNN, RNN, Transformer-based 모델을 학습하여 FP16 연산과 비슷한 학습 정확도를 보이는 내용의 논문을 2022년 IEEE 표준화 그룹에 제출하였다[3]. 같은 시기에 Graphcore, AMD, Qualcomm 또한 8bit 복합 부동소수점 데이터 타입의 실효성을 인정했다[8]. Graphcore는 Mantissa와 Exponent의 Bitwidth를 가변하여 8bit로 가능한 다양한 포맷들로 신경망을 학습한 내용의 논문을 출판했으며, (E4M3, E5M2) 복합 데이터 타입이 AI 전반적인 분야에 적합하다는 의견을 IEEE 표준화 그룹에 제출하였다[4]. IBM은 2019 NIPS 논문[9]을 통해 (E4M3, E5M2) 복합 부동소수점 데이터 타입을 가장 먼저 제안하였기 때문에 표준화에 긍정적이다. 하지만 Qualcomm의 경우 2022 NIPS 논문에서 E4M3보다 E3M4 또는 E2M5 타입의 넓은 Mantissa Bitwidth로 웨이트(weight)와 엑티베이션(activation)을 표현하는 것이 정확도 성능에 더 적합함을 주장하면서 FP8

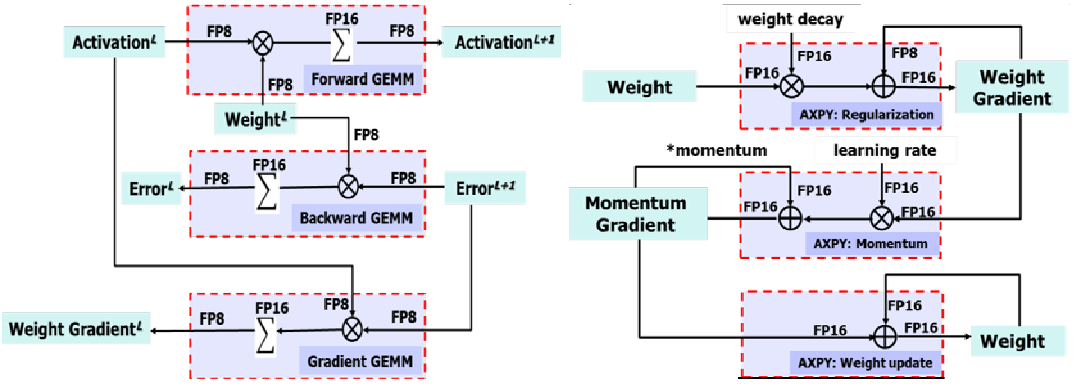
표준 포맷에 대한 논의를 이어나가고 있다[10].

III. 신경망 학습을 위한 저정밀도 부동소수점 포맷 연구 동향

신경망 학습과정에서 행렬곱(GEneral Matrix Multiplication: GEMM)이 사용되는 연산은 처리 시간이 오래 걸리고 메모리 사용량이 많다. 경량화 연구는 이러한 GEMM 함수 기반으로 구현된 신경망 레이어에 대해 학습 정확도 손실을 최소화하면서 효율적으로 처리하도록 신경망 구조 및 연산 데이터를 최적화하는 데 집중하고 있다. 따라서 신경망에서 저정밀도 부동소수점 포맷 FP8은 GEMM 함수로 구현되는 Convolution Layer와 Inner Product(i.e. Fully-Connected) Layer의 순방향과 역방향 연산에 적용되어 함수의 입력 연산자를 단정밀도 FP32대신 저정밀도 FP8로 변환하여 처리한다. Activation Layer 및 Batch-Normalization Layer와 같이 정밀도에 민감하면서도 상대적으로 연산복잡도가 높지 않은 부분은 저정밀도 FP8 사용에 따른 학습시간을 단축하기보다 단정밀도 FP32를 사용하여 학습 정확도가 손실되는 것을 방지하고 있다. 웨이트 업데이트를 위한 옵티마이저(Optimizer) 함수 연산 또한 일반적으로 단정밀도 FP32를 사용한다. 다만, 웨이트와 그레이디언트(gradient)에 저정밀도 FP8을 적용한 경우 FP32 형 변환 및 옵티마이저 함수 연산자를 저장하는 과정에서 불필요한 메모리 사용을 절감하기 위해 웨이트 업데이트에서 사용되는 GEMM 및 AXPY 연산은 부동소수점 기반 16bit를 사용하여 구현한다[9]. 이와 같은 저정밀도 신경망 학습기술에 대한 성능 평가는 종래의 FP32로 학습된 결과와 비교되며, 이미지 처리 및 언어모델을 포함한 전반적인 AI 분야에서 정확도 손실은 약 1% 미만으로 FP32와 유사한 성능을 보인다[3],[4],[9]. 본 챕터에서는 IBM이 주도하는 FP8 포맷 기술의 연구 흐름과 NVIDIA가 제시한 새로운 지수계열 데이터 타입 및 학습방법을 소개한다.

1. IBM FP8(E5M2)

본 기술은 2018 NeurIPS 학회에 발표된 연구내용으로, 신경망 학습 과정에서 GEMM 기반의 순방향, 역방향 연산에 사용되는 Weight, Activation, Error, Gradient 연산자에 대해 부동소수점 8bit의 E5M2 포맷을 적용했다([그림 2] 참조)[11]. GEMM 내부 연산에서



[FP8-GEMM Functions during Fwd/Bwd Pass] [FP8-AXPY Functions during Weight Update]

〈자료〉 Wang, Naigang, et al., "Training deep neural networks with 8-bit floating point numbers", Advances in neural information processing systems 31, 2018.

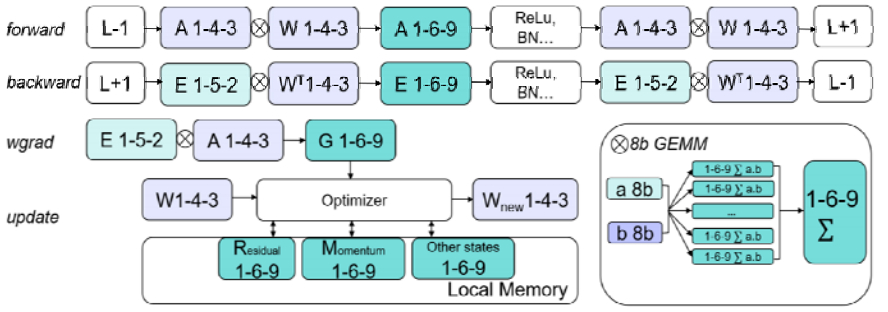
[그림 2] IBM FP8(E5M2) 포맷 기반의 신경망 학습

누적용 데이터 타입은 단정밀도 FP32 대신 E6M9 포맷의 부동소수점 16bit를 사용하여 연산시간을 단축하고 메모리 사용량을 줄였다. 웨이트 업데이트 과정에서 사용되는 Weight, Gradient, Momentum 정보는 E6M9 포맷의 16bit를 적용하여 GEMM, AXPY 함수에 사용함으로써 종래의 FP32보다 불필요한 메모리 사용을 절감했다.

행렬곱의 누적 과정에서 단정밀도 FP32의 23bit 지수보다 9bit의 간소화된 지수를 사용하면 부동소수점 덧셈에서 정보 손실이 심화되는 swamping 현상에 노출되기 쉬워진다[12]. 본 고는 누적 연산을 청크(chunk) 단위로 분리한 뒤 두 단계로 분할 수행하는 chunk-based accumulation 기법을 사용했다. 이 방법은 입력 연산자의 지수 크기 차이를 작게 만듦으로써 9bit 지수에 따른 정보 손실을 절감할 수 있다. 또한, 지수를 9bit로 라운딩하는 과정에서 리지듀얼(residual) 정보를 활용하여 적은 지수 정보에 의한 손실을 보상하기 위해 확률 기반으로 수를 표현하는 stochastic rounding 기법을 적용했다.

2. IBM Hybrid FP8(E4M3 & E5M2)

본 기술은 2019 NeurIPS 학회에 발표된 연구 내용으로, 웨이트와 액티베이션은 E4M3 포맷을 사용해서 2018년에 발표한 E5M2 포맷보다 지수 정보를 보강하고 역방향 연산의 에러와 그레이디언트는 E5M2 포맷을 통해 지수(exponent)의 범위를 크게 두어 지수적으



〈자료〉 Sun, Xiao, et al., Supplemental of "Hybrid 8-bit floating point(HFP8) training and inference for deep neural networks", Advances in neural information processing systems 32, 2019.

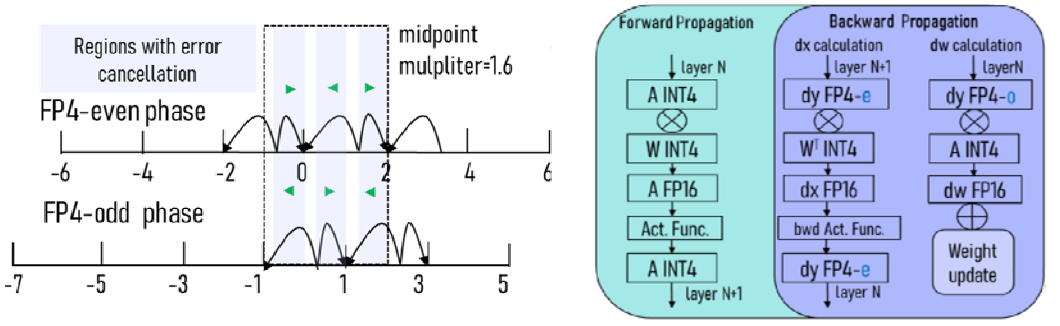
[그림 3] IBM HFP8(E4M3, E5M2) 포맷 기반의 신경망 학습

로 매우 작으면서 넓은 값을 표현할 수 있도록 설정했다([그림 3] 참조)[9]. GEMM 함수의 누적용 데이터 타입은 부동소수점 16bit의 E6M9 타입을 사용했다.

E4M3 포맷은 exponent bias를 별도로 두어 더 0에 가까운 값 위주로 표현하거나 더 큰 값 위주로 표현할 수 있도록 조절하게 하였다. 바이어스는 설정값에 따라 신경망 학습의 성공 여부를 결정할 정도로 매우 중요하다[4]. 본 고는 다양한 실험을 통해 전반적으로 학습이 성공하는 수준의 고정된 바이어스 값을 실험적으로 찾았다. 알고리즘의 복잡도를 고려하여 그 값을 16으로 고정하고 데이터를 2^{-4} 크기만큼 더 작은 값 위주로 표현하도록 설정했다. 이는 신경망 연산데이터가 1보다 큰 값보다 0에 가까운 값이 더 많이 존재함을 고려하였다. 또한, 웨이트 업데이트 연산을 통해 계산된 새로운 웨이트를 E4M3 타입으로 변환하는 과정에서 형 변환 전후 차이를 리지듀얼 값으로 별도 저장한 후, 다음 업데이트 과정에서 옵티마이저 함수에 해당 값을 반영하여 8bit 웨이트에 의한 정보 손실을 보상하는 기법을 도입했다.

3. IBM Radix4 FP4(Radix4-E3M0 & E0M3)

본 기술은 2020 NeurIPS 학회에 발표된 연구 내용으로, 4bit로 신경망을 학습하는 연구이다([그림 4] 참조)[13]. IBM은 이전 연구들을 통해 웨이트와 액티베이션을 표현하는데 지수 정보가 중요하고, 에러와 그레이디언트는 지수 정보가 성능에 주된 영향을 준다는 사실을 확인하였다. 그에 따라 순방향 연산은 지수로 구성된 INT4(i.e. E0M3) 포맷을 사용하고 역방향 연산의 에러, 그레이디언트는 지수로 구성된 E3M0 포맷을 사용했다. E3M0 포맷은



[Two-Phase Rounding] [Radix4 FP4 Pw/Bwd/Gradient Pass]

〈자료〉 Sun, Xiao, et al., "Ultra-low precision 4-bit training of deep neural networks", Advances in Neural Information Processing Systems 33, 2020, 1796-1807.

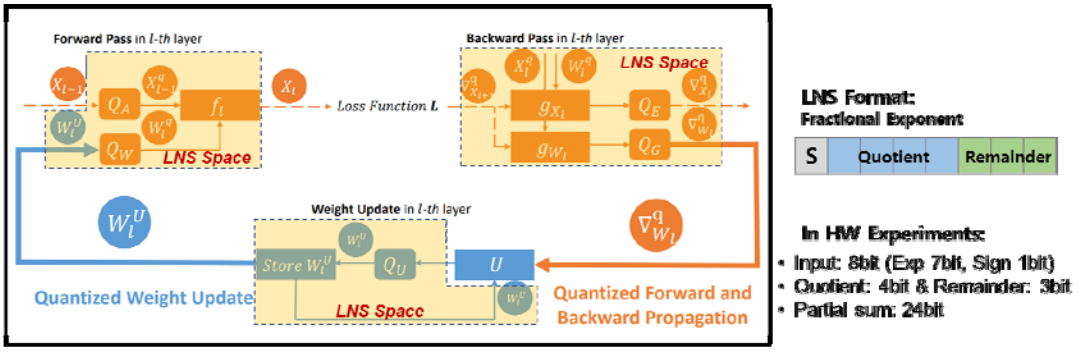
[그림 4] IBM Radix4 FP4(E3M0, E0M3) 포맷 기반의 신경망 학습

기존 부동소수점의 수체계가 2의 n승을 따르던 것과 달리 4의 n승을 따르는 Radix4 수체계를 사용하여 Radix2 Exponent보다 더 넓은 범위의 데이터를 표현하도록 하였다. 행렬곱의 누적 연산은 기존 연구와 동일하게 16bit의 E6M9 포맷을 사용한다.

종래의 FP32 포맷을 통해 32bit로 표현되던 신경망 데이터가 4bit로 변환되면 데이터는 24개의 경우의 수로 표현해야 한다. 적은 경우의 수로 연산 데이터를 정확하게 모델링 하려면 최적화된 exponent bias 값이 필요하다. 기존 논문은 전체 신경망에 대해 통일된 exponent bias를 사용했다면, 본 고는 레이어마다 학습을 통해 exponent bias 값을 설정하여 데이터의 표현범위를 세밀하게 조절했다. 또한, 역방향 연산의 에러와 그레이디언트를 계산하는 과정에서 서로 다른 라운딩 기법을 적용하여 Radix4 Exponent 사용에 따른 스파스(sparse) 표현 구간을 보상하는 기법을 도입했다.

4. NVIDIA LNS

본 기술은 2022 IEEE Transactions on Computers 저널에 발표된 연구내용으로, 부동소수점 데이터 포맷을 지수계열의 수체계(Logarithmic Number System: LNS)로 일반화하여 신경망을 학습하는 방법론 및 하드웨어 구조에 관한 것이다([그림 5] 참조)[14]. 종래의 부동소수점 포맷은 지수를 정수로 표현하고 있다. 본 고는 지수를 실수로 표현하여 지수 정보까지 통합한 표현방식을 따른다. 그 결과 행렬곱 과정에서 곱셈 연산은 지수 덧셈으로



LNS Fwd/Bwd/Gradient/Update of DNN Training

〈자료〉 Zhao, Jiawei, et al., "LNS-Madam: Low-Precision Training in Logarithmic Number System using Multiplicative Weight Update", IEEE Transactions on Computers, 2022.

[그림 5] NVIDIA LNS 포맷 기반의 신경망 학습

치환되고 누적 연산은 LUT(Look-Up Table) 기반의 정수형 덧셈 연산기로 구현할 수 있다. LNS 포맷은 Fixed Point의 7bit Exponent 정보와 1bit Sign 정보로 구성된다. 실수형 지수는 4bit의 정수와 3bit의 소수로 구분되며, 누적 연산은 정수형 24bit를 사용한다.

신경망 학습 과정에서 웨이트와 그레이디언트를 LNS 포맷으로 표현하는 경우, 기존의 학습방법을 따른다면 웨이트 업데이트 연산을 위해 다시 부동소수점 포맷으로 형 변환하는 과정이 필요하다. 논문은 LNS 포맷으로 웨이트 업데이트 연산이 가능하도록 형 변환이 필요 없는 옵티마이저 함수를 개발했다. LNS 포맷은 지수값으로 데이터를 표현하기 때문에 업데이트에 필요한 그레이디언트는 옵티마이저 수식에서 지수에 직접 반영되어야 양자화 손실을 최소화할 수 있다. 따라서 옵티마이저 함수는 multiplicative function으로 구성하여 2의 지수 부분을 통해 웨이트의 학습이 가능하도록 정의했다.

IV. 결론

인공신경망은 갈수록 거대해지며 학습을 위한 메모리 사용량, 지능형 반도체의 개수도 점차 증가하고 있다. 그에 따라 신경망을 낮은 정보량으로 빠르고 효율적으로 처리하는 학습 알고리즘은 필연적으로 발전하고 있다. 2020년까지 IBM의 연구를 통해 신경망에서 연산밀도가 높은 부분은 4bit의 정보량으로 빠르게 학습이 가능하고, 특히 지수계열의 데이터가

신경망 학습연산에 중요함이 확인되었다. 그러나 정확도 손실을 보상하기 위한 부수적인 알고리즘의 비중이 증가했다. 2022년 NVIDIA의 연구는 학습에 필요한 정보량이 낮아짐에 따라 부동소수점 연산기를 탈피하려는 노력에서 시작되었다. 그 결과 신경망 학습에서 지수 계열 데이터 포맷을 사용하되 부동소수점 연산기를 사용하지 않고 정수형 연산기만을 가지고 신경망을 학습하는 방법론과 하드웨어 구조를 고안하여 연산기 최적화된 AI전용 수체계를 정의했다고 할 수 있다. 하지만 별도의 전용 옵티마이저 함수를 구현해야 하므로 학습 알고리즘 활용에 제한이 있다.

현재 지능형 반도체 연구 분야는 산업계를 중심으로 움직이고 있다. 반도체 설계 및 공정을 위한 막대한 자본을 감당하기에는 산업계의 도움 없이 학계와 연구계만으로는 접근하기 힘든 분야가 되었다. 지능형 반도체와 시스템, 관련 알고리즘을 인공지능 서버 및 단말에 실증하여 실효성을 검증하기 위해서도 기술의 수요층이 확고한 산업계를 통해 진행될 필요가 있다. NVIDIA의 경우, NVResearch Center를 통해 AI 반도체 기술 개발을 위한 산·학·연 클러스터의 중심 역할을 해내고 있다. MIT, Stanford, Harvard, CMU 등 세계 상위권 대학 및 정부산하 기관들과 학술 교류를 진행하고 있으며 대학원생 인턴십 프로그램을 통해 SW와 HW의 경계 없이 다양한 분야의 최신 연구 동향을 파악하여 학술 저널에 논문을 발표하고 있다. 국내에도 지능형 반도체 산업계를 중심으로 산·학·연을 아울러 최신 기술을 교류하고 개발 시간을 단축하여 기술 도약의 선순환을 이루는 생태계가 구성되기를 희망한다.

참고문헌

[1] Russakovsky, Olga, et al., “Imagenet large scale visual recognition challenge”, International journal of computer vision 115.3, 2015, pp.211-252.

[2] NVIDIA, “NVIDIA H100 Tensor Core GPU Architecture”, NVIDIA, 2022.

[3] Micikevicius, Paulius, et al., “FP8 Formats for Deep Learning”, arXiv preprint arXiv:2209.05433 2022.

[4] Noune, Badreddine, et al., “8-bit Numerical Formats for Deep Neural Networks”, arXiv preprint arXiv:2206.02915, 2022.

[5] Tesla, “A Guide to Tesla’s Configurable Floating Point Formats & Arithmetic”, Tesla, 2021.

[6] NVIDIA, “NVIDIA Tesla V100 GPU Architecture”, NVIDIA, 2017.

[7] NVIDIA, “NVIDIA A100 Tensor Core GPU Architecture”, NVIDIA, 2020.

[8] Graphcore, “GRAPHCORE AND AMD PROPOSE FP8 AI STANDARD WITH QUALCOMM

SUPPORT", 2022/07/05.
[9] Sun, Xiao, et al., "Hybrid 8-bit floating point(HFP8) training and inference for deep neural networks", Advances in neural information processing systems 32, 2019.
[10] Kuzmin, Andrey, et al., "FP8 Quantization: The Power of the Exponent", arXiv preprint arXiv:2208.09225, 2022.
[11] Wang, Naigang, et al., "Training deep neural networks with 8-bit floating point numbers", Advances in neural information processing systems 31, 2018.
[12] Higham, Nicholas J., "The accuracy of floating point summation", SIAM Journal on Scientific Computing 14.4, 1993, pp.783-799.
[13] Sun, Xiao, et al., "Ultra-low precision 4-bit training of deep neural networks", Advances in Neural Information Processing Systems 33, 2020, pp.1796-1807.
[14] Zhao, Jiawei, et al., "LNS-Madam: Low-Precision Training in Logarithmic Number System using Multiplicative Weight Update", IEEE Transactions on Computers, 2022.