온디바이스에서의 딥러닝 모델 최적화 기법 동향

윤세현 [1], 최상현 [1], 오현영 [2*]
[1] 가천대학교 AI 소프트웨어학부 학부생
[2] 가천대학교 AI 소프트웨어학부 교수

wjwjwj2002@gachon.ac.kr, shchoi00@gachon.ac.kr, hyoh@gachon.ac.kr

A Survey on On-Device DL-Model Optimization Techniques

Se-Hyeon Yoon, Sang-Hyun Choi, Hyunyoung Oh
Dept. of AI · Software, Gachon University

요 약

온디바이스 환경에서 딥러닝 모델 최적화는 필수적이지만, 제한된 자원으로 고성능 모델을 직접 적용하는 데에는 한계가 있다. 본 논문에서는 이를 극복하기 위한 주요 기법인 가지치기, 양자화, 지식 증류, 신경망 아키텍처 탐색 및 이들의 결합 기법을 소개하고 분석한다. 각 기법의 정의와 특징, 적용 사례를 통해 성능 향상과 자원 효율성을 극대화하는 방법을 제시하며, 이를 바탕으로 최근 연구 동향을 소개한다.

1. 서론

최근 임베디드 시스템과 SDV (Software Defined Vehicle)와 같은 분야에서 온디바이스(On-device) 컴퓨팅의 중요성이 점점 더 커지고 있다[1]. 온디바이스 컴퓨팅은 지연 시간 감소, 데이터 전송 의존도 감소, 프라이버시 보호 강화 등의 장점을 제공한다[2]. 이에 따라 딥러닝 네트워크를 온디바이스 환경에 통합하려는 시도가 활발히 이루어지고 있으며, 특히 객체 탐지 (Object Detection), 이미지 분할 (Image Segmentation)과 같은 복잡한 작업에서 그 필요성이 두드러지고 있다[3-5]. 그러나 온디바이스 환경에서 딥러닝 모델을 최적화하는 일은 여전히 과제가 남아 있다. 제한된 전력과 메모리와 같은 자원 제약 속에서 네트워크의 성능을 극대화하는 것은 어려운 일이다[6].

본 논문에서는 온디바이스 환경에서 딥러닝 모델을 효율적으로 운영하기 위한 최신 최적화 기법들을 종합적으로 검토하고, 기법들의 적용 사례와 성과를 분석하고자 한다. 본 연구의 목적은 온디바이스 딥러닝 최적화 기법의 동향을 정리함으로써, 제한된 컴퓨팅 자원으로도 높은 성능을 발휘할 수 있는 전략을 제시하는 것이다.

2. 모델 최적화 기법

본 논문에서 소개할 모델 최적화 기법은 가지치기, 양자화, 지식 증류, 신경망 아키텍처 탐색, 결합 기법으로 총 다섯 가지이다,

- 가지치기 (Pruning)

가지치기[7]는 모델에서 중요도가 낮은 가중치나 불필요한 뉴런을 제거하는 기법이다. 이는 불필요한 연결을 제거하여 네트워크의 복잡성을 줄여 모델을 일반화시킬 수 있고, 연산 자원과 메모리 사용을 최소화할 수 있다. 온디바이스 딥러닝에서 가지치기는 메모리 자원과 계산 효율성을 최적화할 수 있는 중요한 방법이다[8]. DeepIoT[9]는 딥러닝 모델의 크기를 98.9%까지 줄이면서도, 정확도 손실 없이 에너지 소비를 최대 95.7%까지 감소시켰다.

- 양자화 (Quantization)

양자화는 딥러닝 모델의 가중치와 활성화 값을 낮은 비트로 나타내어 모델의 연산을 줄이는 기법이다. 일반적으로 32 비트 부동소수로 표현되는 모델의 매개 변수를 8 비트나 16 비트로 줄여 연산을 가볍게 할 수 있다. 이를 통해 모델의 크기를 줄이고 연산 속도를 증가시킬 수 있다. 특히 양자화는 메모리와 계산 자원이 제한된 온디바이스 환경에서 실시간 처리와 에너지 효율성을 높이는 데 효과적이다[10]. Wu et al.[11]은 8 비트 정수 양자화를 사용하여 메모리 사용량을 최대 4 배까지 줄일 수 있었으며, 특히 합성곱 및 행렬 곱셈 연산에서 최대 16 배의 속도 향상을 달성했다.

* 교신저자

- 지식 증류 (Knowledge Distillation)

지식 증류[12]는 큰 교사 모델 (Teacher Model)의 지식을 작은 학생 모델 (Student Model)에 전달하여, 성능 손실을 최소화하면서 모델의 크기를 경량화하는 기법이다. 교사 모델이 복잡한 계산을 통해 학습한 높은 수준의 정보를 학생 모델이 학습하여, 작은 모델이더라도 원래 교사 모델과 유사한 성능을 달성할 수 있다. EdgeSAM[13]은 지식 증류를 통해 최적화를 성공한 사례이다. 이는 기존 SAM 에 비해 속도에서 37 배 빠른 성능을 보이며, iPhone 14 와 같은 엣지 디바이스에서 30 FPS 이상으로 실행된다. 성능 면에서도 COCO 데이터셋에서 mIoU 2.3% 향상, LVIS 데이터셋에서 mIoU 3.1% 개선을 달성하였다.

- 신경망 아키텍처 탐색 (NAS)

신경망 아키텍처 탐색 (Neural Architecture Search)[14]은 딥러닝 모델의 아키텍처를 자동으로 설계하는 기법이다. 신경망 아키텍쳐 탐색을 통해 사람이 직접 모델을 설계할 필요 없이 강화 학습, 진화 알고리즘, 또는 차등 가능한 최적화 기법을 통해 다양한 아키텍처를 탐색하고, 주어진 데이터와 하드웨어 환경에 최적화된 아키텍처를 자동으로 생성할 수 있다. DNAS[15] 에서는 경량화된 Differentiable NAS 를 사용해 네트워크 가중치와 아키텍처를 동시에 최적화했다. 또한, 하드웨어 제약을 반영한 비용 정규화를 추가해 IoT 디바이스에서의 성능을 극대화했다. 이를 통해 메모리 사용량은 최대 30%, 추론 시간은 25% 단축되었다.

- 결합 기법 (Combined Methods)

가지치기, 양자화, 지식 증류, 그리고 신경망 아키텍처 탐색과 같은 최적화 기법들은 각각 모델의 크기와 연산 복잡도를 줄이는 데 효과적이다. 그러나, 각 기법을 단독으로 사용할 때는 성능 저하를 초래하거나 특정 작업에 최적화되지 않을 수 있다. 이에 따라 여러 기법을 결합하여 사용하는 조합 기법이 최근 사용되고 있다. PQK[16]는 가지치기, 양자화, 그리고 지식 증류를 결합하여 딥러닝 모델을 압축하면서도 성능을 유지했다. ResNet-32 모델을 CIFAR-100 데이터셋에서 실험한 결과, 32 비트에서 8 비트 양자화만 적용했을 때, 정확도는 67.4%였다. 양자화와 가지치기를 결합했을 때의 정확도는 69.8%로 성능이 회복되었고 모델 크기는 90% 절감되었다. 양자화, 가지치기, 지식 증류를 모두 결합했을 때, 성능은 69.8%, 모델 크기는 최대 90%까지 줄어드는 결과를 보였다.

3. 결론 및 향후 계획

본 논문에서는 딥러닝 최적화 기법인 가지치기, 양자화, 지식 증류, 그리고 신경망 아키텍처 탐색에 대해 분석하였다. 이 기법들은 자원이 제한된 온디바이스 환경에서 딥러닝 모델을 경량화하면서도 높은 성능을 유지할 수 있도록 돕는다. 특히 메모리 사용량과 연산 비용을 줄이면서도, 실시간 응답성을 높이는 데 기여한다. 향후 연구에서는 이러한 최적화 기법들을 더욱 통합하고, 하드웨어와 소프트웨어의 협력을 통해 더욱 효율적이고 성능이 뛰어난 온디바이스 딥러닝 솔루션을 연구하고자 한다. 또한 다양한 응용 분야에서의 실제 적용 사례를 확대하여 온디바이스 딥러닝의 활용 가능성을 높일 예정이다.

사사문구

이 논문은 2024 년도 정부(산업통상자원부)의 재원으로 한국산업기술기획평가원의 지원(No. RS-2024-00406121, 자동차보안취약점기반위협분석시스템개발(R&D))과 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원(No. RS-2022-00166529)을 받고 과기정통부 정보통신기획평가원의 정보보호핵심원천기술개발사업(No. RS-2024-00337414)으로 수행한 결과임.

참고문헌

[1] T. Springer et al., "On-Device Deep Learning Inference for SoC Architectures," Electronics, 2021.
[2] T. Zhao et al., "A Survey of Deep Learning on Mobile Devices: Applications, Optimizations, Challenges," Proceedings of the IEEE, 2022.
[3] J. Yang et al., "On-Device Unsupervised Image Segmentation," ACM/IEEE DAC, 2023.
[4] M. Hanyao et al., "Edge-assisted On-device Object Detection for Real-time Video Analytics," IEEE INFOCOM, 2021.
[5] Z. Qin et al., "ThunderNet: Real-time Object Detection on Mobile Devices," ICCV, 2019.
[6] Y. Cheng et al., "A Survey of Model Compression and Acceleration for Deep Neural Networks," arXiv, 2020.
[7] Z. Liu et al., "Rethinking the Value of Network Pruning," arXiv, 2019.
[8] R. Reed, "Pruning algorithms-a survey," IEEE Transactions on Neural Networks, 1993.
[9] S. Yao et al., "DeepIoT: Compressing Deep Neural Network Structures for Sensing Systems with a Compressor-Critic Framework," arXiv, 2017.
[10] B. Rokh et al., "A Comprehensive Survey on Model Quantization for Deep Neural Networks in Image Classification," ACM Transactions on Intelligent Systems and Technology, 2023.
[11] H. Wu et al., "Integer Quantization for Deep Learning Inference: Principles and Empirical Evaluation," arXiv, 2020.
[12] N. Kim, J. An, "Knowledge Distillation for Traversable Region Detection of LiDAR Scan in Off-Road Environments," Sensors, 2024.
[13] C. Zhou et al., "EdgeSAM: Prompt-In-the-Loop Distillation for On-Device Deployment of SAM," arXiv, 2024.
[14] B. Zoph, Q. V. Le, "Neural Architecture Search with Reinforcement Learning," arXiv, 2017.
[15] A. Burrello et al., "Enhancing Neural Architecture Search With Multiple Hardware Constraints for Deep Learning Model Deployment on Tiny Devices," IEEE Transactions on Emerging Topics in Computing, 2024
[16] J. Kim et al., "PQK: Model Compression via Pruning, Quantization, and Knowledge Distillation," arXiv, 2021.