임베디드 시스템에서의 객체 탐지 네트워크 양자화

*신상준, 신동군
성균관대학교 컴퓨터공학과
e-mail : *sangjune97@g.skku.edu, dongkun@skku.edu*

Quantization of Object Detection Networks in Embedded Systems

*Sangjune Shin, Dongkun Shin
Computer Science and Engineering
Sungkyunkwan University

Abstract

This paper addresses the limitations of real-time application of object detection techniques in embedded computing environments. To overcome these limitations, compression methods such as pruning and quantization are being investigated. In this study, various quantization techniques, including post-training quantization, partial quantization, and quantization-aware training, are applied to the YOLO v3 model to evaluate and analyze their effects in embedded system environments. Experiments were conducted using the YOLO v3 model and the PASCAL VOC dataset. The experimental results show that the quantized models can reduce model size and inference time but result in accuracy loss.

Therefore, this research contributes to improving object detection performance in embedded systems by proposing and analyzing various quantization techniques for lightweighting object detection technology in embedded environments.

이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.IITP-2017-0-00914, 지능형 IoT 장치용 소프트웨어 프레임워크)

I. 서론

최근 객체 탐지 기술은 드론이나 자율주행 자동차 등 다양한 분야에 활용되며 많은 발전을 이루어 왔다. 그러나 제한적인 전력과 메모리, 연산능력 등 다양한 이유로 임베디드 컴퓨팅 환경에서 실시간으로 객체 탐지 기술을 적용하기에는 아직 한계가 있다. 이에 따라 가지치기나 양자화 등 다양한 경량화 기법이 연구되고 있다[1].

가지치기 기법은 네트워크를 구성하는 파라미터 중 중요하지 않은 파라미터를 제거하여 파라미터 수를 줄이는 경량화 기법이다. 각각의 파라미터의 중요도를 개별적으로 판단하여 희소 행렬을 가지게 되는 구조화되지 않은 가지치기(Unstructured pruning) 기법과 여러 파라미터의 묶음을 단위로 중요도를 판단하는 구조화된 가지치기(Structured pruning) 기법으로 나뉜다[2] [3].

양자화 기법은 네트워크의 구조는 유지하면서 32비트 부동 소수점으로 훈련된 파라미터를 낮은 비트의 고정 소수점 또는 정수로 변환하여 추론 연산을 수행하는 기법이다. 이를 통해 네트워크의 파라미터를 저장하기 위한 공간을 줄일 수 있고, 메모리 대역폭을 낮춰 빠른 추록 속도를 얻을 수 있다.

본 논문에서는 객체 감지 모델의 하나인 YOLO v3[4]를 대상으로 훈련 후 양자화(Post-training quantization), 부분 양자화(Partial quantization), 양자화 인식 훈련(Quantization-aware training) 등 다양한 양자화 기법을 적용하고 임베디드 시스템 환경에서 양자화 기법이 미치는 영향을 평가 및 분석하였다.

II. 본론

2.1 YOLO v3

기존 객체 탐지 알고리즘은 객체의 위치 제안과 객체 분류를 순차적으로 수행하는 2-stage detector 모델과 이를 동시에 수행하는 1-stage detector모델로 분류할 수 있다.

R-CNN[5], Faster R-CNN[6], Mask R-CNN[7] 등의 2-stage detector는 높은 정확성을 보장하지만, 느린 속도로 인해 임베디드 시스템에 분야에서 사용하기에는 너무 무거웠다. 이러한 문제를 해결하기 위해 제안된 YOLO[8], SSD[9], RetinaNet[10]등 1-stage detector는 정확도는 소폭 떨어지지만 훨씬 빠른 추론속도로 성능 개선을 이루었다.

대표적인 1-stage 객체 탐지기인 YOLO의 개선 모델 중 하나인 YOLOv3는 입력 이미지에서 객체 분류 및 위치 정보 획득을 위한 피처(Feature)를 추출하기 위한 백본(Backbone) 계층, 다양한 크기의 피처(Feature)를 가공하여 활용할 수 있도록 만드는 넥(Neck) 계층, 최종적으로 탐지한 물체의 분류와 경계 박스 위치 정보를 추론하는 헤드(Head) 계층으로 이루어져 있다.

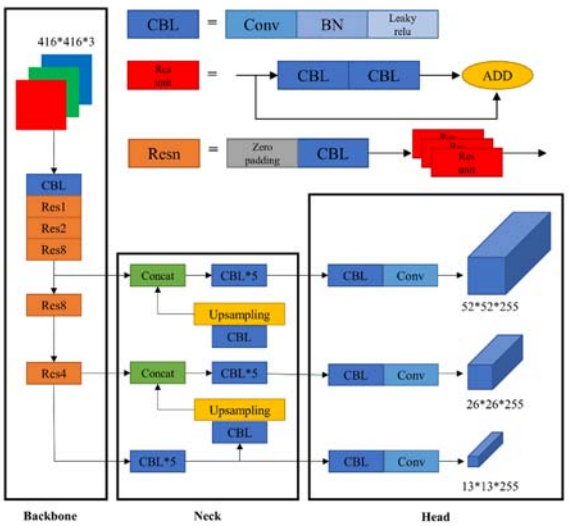

그림 1. YOLO v3의 구조

2.2 훈련 후 양자화 (PTQ)

훈련 후 양자화(Post-training quantization, PTQ)는 32비트 부동소수점으로 사전 훈련된 모델의 가중치와 활성화 값을 더 낮은 k(=1, 2, 4, 8, 16) 비트로 변환하는 기법이다[11]. 먼저 가중치와 활성화 값이 양자화되기 이전에 성능 감소를 최소화하기 위해 보정(Calibration) 단계를 거친다. 보정 단계에서는 일부 학습 데이터를 통해 각 계층에서 출력되는 값의 범위를 확인하여 각 계층마다 가중치 및 활성화 값을 정확하게 표현할 수 있도록 최적화된 스케일(Scale) 및 오프셋(Offset)을 결정한다. 이후 보정 단계에서 결정된 스케일 및 오프셋 값을 통해 가중치 및 활성화 값을 더 낮은 비트 수의 값으로 양자화를 진행하고 양자화된 모델을 생성한다.

본 논문에서는 32비트 부동소수점으로 사전 훈련된 YOLO v3 모델을 8비트 정수로 양자화를 진행하였다.

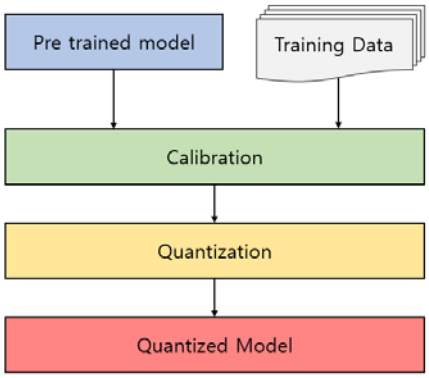

그림 2. 훈련 후 양자화 진행 과정

2.3 양자화 인식 훈련 (QAT)

양자화 인식 훈련(Quantization-aware training, QAT)은 32비트 부동소수점으로 사전 훈련된 모델에 양자화기(Quantizer)를 삽입하고 부동 소수점으로 다시 훈련하는 방법이다[12]. 양자화기는 재훈련 과정에서 순방향 경로에서 양자화 오류를 시뮬레이션하며, 재훈련 이후 양자화기에 저장된 정보를 통해 각 계층마다 이를 최소화하는 스케일 및 오프셋을 결정한다.

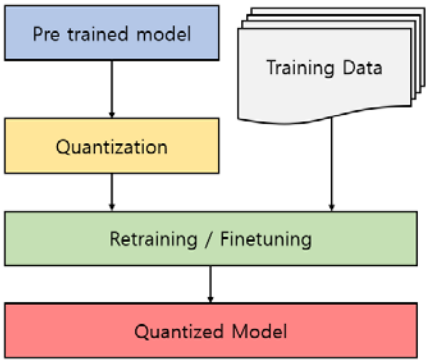

그림 3. 양자화 인식 훈련 진행 과정

2.4 부분 양자화 (Partial quantization)

네트워크 양자화 기법은 모델 크기와 추론 시간을 줄일 수 있지만, 추론 정확도에서 손실이 발생한다. 부분 양자화(Partial quantization)는 모델 압축과 추론 성능 유지라는 장점을 모두 활용하기 위해 제안된 기법이다[13]. 기존 네트워크 양자화 기법들은 전체 모델을 균일하게 양자화 했었다.[14] 그러나 정확도에 대한 양자화의 영향은 각 계층마다 다른 것으로 알려져 있다. 부분 양자화는 각 계층마다 비트 폭을 조정하여 효율적으로 모델을 양자화 한다. 본 논문에서는 YOLO v3 모델을 구성하는 백본(Backbone), 넥(Neck), 헤드(Head) 계층별로 다르게 부분 양자화를 진행하여 각 계층의 양자화가 모델 정확도에 미치는 영향을 살펴보려 한다.

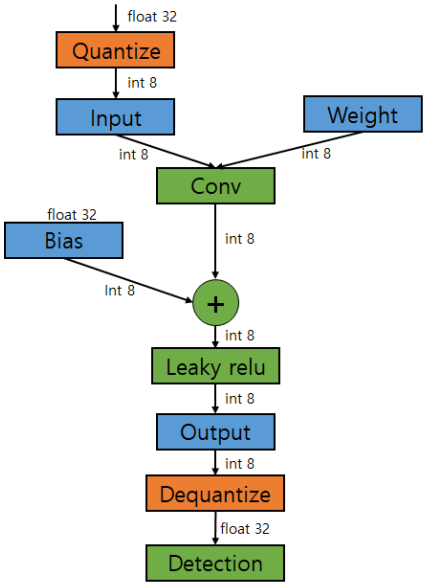

그림 4. 양자화된 네트워크의 추론 과정

Ⅲ. 실험

3.1 실험 환경 구성

신경망은 YOLO v3, 학습 데이터 셋은 PASCAL VOC 2007, 2012 데이터 셋을 이용하였다. VOC 데이터 셋은 탈 것, 가정용품, 동물 등을 포함한 20개 종류의 객체를 가지며, 16,551개의 학습 영상과 4,952개의 테스트 영상으로 구성된다. YOLO v3의 Pytorch 구현 모델을 이용하여 실험을 진행하였으며, 모델의 학습은 optimizer로 Adam을 사용하였고, 학습률은 초기값 1e-3에서 10 스텝마다 0.8을 곱해주도록 설정 후 200 epoch 동안 학습을 진행하였다. 모델의 입력 크기는 416 픽셀로 고정하였다.

이후 임베디드 시스템에서의 YOLO v3의 다양한 양자화 기법의 효과를 알아보기 위해서 Raspberry Pi 3B+ 에서 원본 모델, 훈련 후 양자화 적용 모델, 부분 양자화 적용 모델, 양자화 인식 훈련 적용 모델의 정확도 및 성능을 평가하였다. 실험 결과는 표 1,2 및 그림 5에 정리되어 있다.

3.2 훈련 후 양자화 (PTQ)

PTQ는 YOLO v3의 모든 계층을 INT8로 양자화 한 결과이다. 보정 데이터셋으로는 학습에 사용된 데이터셋에서 무작위로 2,000개의 이미지를 추출하여 사용하였다. 표1에서 보듯이 훈련 후 양자화에서 모델 크기는 약 74.8%가 줄어들었으며, 추론 시간이 약 48.4% 빨라졌다. 그러나 정확도가 69.33%에서 67.54%로 약 2.58%라는 매우 큰 정확도 손실을 보였다.

3.3 양자화 인식 훈련 (QAT)

QAT는 YOLO v3 원본 모델과 같은 환경에서 10 epoch동안 추가적으로 훈련을 진행하였다. 양자화 인식 훈련의 경우 훈련 후 양자화를 진행한 모델과 모델 크기 및 추론 시간에서는 거의 차이가 없었으나, 정확도가 69.29%로 원본 모델에 비해 0.06%라는 굉장히 적은 정확도 손실을 보였다.

Method	Precision	Model Size(MB)	mAP(%)	Latency(s)	FPS
Baseline	FP32	241.09	69.33	12.49	0.08
PTQ	INT8	60.84	67.54	6.51	0.15
QAT	INT8	60.84	69.29	6.51	0.15

표 1. QAT 및 PTQ를 적용한 모델의 성능

3.4 부분 양자화 (Partial quantization)

각 계층의 양자화가 모델 정확도에 미치는 영향을 살펴보기 위해서 백본(Backbone), 넥(Neck), 헤드(Head) 중 하나의 계층 또는 두개의 계층에 각각 훈련 후 양자화 기법을 통해 부분 양자화를 적용하였다.

표2에서 B, N, H는 각각 백본(Backbone), 넥(Neck), 헤드(Head)의 정밀도를 나타내며, 32와 8은 각각 FP32, INT8 정밀도를 적용하였음을 나타낸다.

그림5는 각각 원본 모델, 백본(Backbone) 양자화 모델, 넥(Neck) 양자화 모델, 헤드(Head) 양자화 모델과 PTQ, QAT를 적용한 모델의 정확도와 추론 시간, 그리고 모델 크기를 나타낸 그래프이다.

백본(Backbone) 계층은 양자화 시 추론 시간이 20.2% 빨라지고, 모델 크기가 50.5%로 크게 감소하여 양자화의 효과가 크게 나타났으나, 정확도 또한 0.87% 감소하였다.

넥(Neck) 계층은 추론시간이 1.4% 빨라졌으며 모델 크기는 17.98% 줄었고, 정확도는 0.32% 감소하여 양자화에 큰 영향을 받지 않았다.

반면 헤드(Head) 계층은 모델 크기 및 추론 시간은 각각 3.4%, 7.9%로 소폭 감소하였으나, 정확도가 1.15%나 감소하여 양자화가 효율적이지 않은 계층임을 보여준다.

Method	Precision (B,N,H)	Model Size(MB)	mAP(%)	Latency(s)	FPS
Baseline	32, 32, 32	241.09	69.33	12.49	0.08
Partial Quantization	8, 32, 32	121.95	68.73	9.96	0.10
	32, 8, 32	197.65	69.11	12.31	0.08
	32, 32, 8	222.15	68.53	12.06	0.08

표 2. 부분 양자화 기법 별 성능

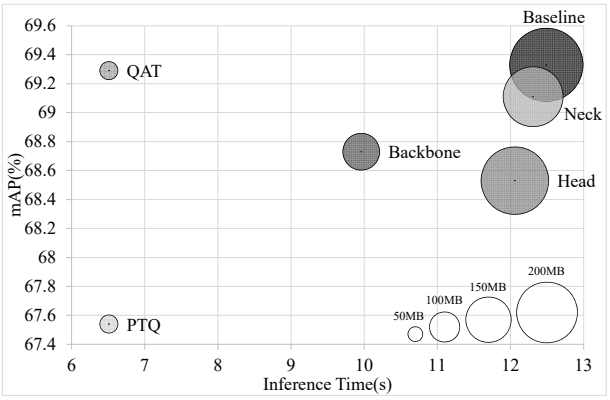

그림 5. 양자화 결과 분석

Ⅳ. 결론

객체 감지 모델의 하나인 YOLO v3 대상으로 훈련 후 양자화(Post-training quantization), 부분 양자화(Partial quantization), 양자화 인식 훈련(Quantization-aware training) 등 다양한 양자화 기법을 적용하고 임베디드 시스템 환경에서 양자화 기법이 미치는 영향을 평가 및 분석하였다.

실험을 통해 양자화 인식 훈련이 훈련 후 양자화보다 우수한 성능을 보이는 것을 확인하였으며, YOLO v3의 구성 요소 중 헤드(Head) 계층이 양자화에 적합하지 않음을 보였다.

참고문헌

[1] Han, Song, Huizi Mao, and William J. Dally. "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," arXiv preprint arXiv:1510.00149 (2015).

[2] Han, Song, et al. "Learning both weights and connections for efficient neural network," Advances in neural information processing systems 28 (2015).

[3] Li, Hao, et al. "Pruning filters for efficient convnets," arXiv preprint arXiv:1608.08710 (2016).

[4] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767 (2018).

[5] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks," Advances in neural information processing systems 28 (2015).

[6] He, Kaiming, et al. "Mask r-cnn," Proceedings of the IEEE international conference on computer vision, 2017.

[7] Cai, Zhaowei, and Nuno Vasconcelos. "Cascade r-cnn: Delving into high quality object detection," Proceedings of the IEEE conference on computer vision and pattern recognition, 2018.

[8] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection," Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

[9] Liu, Wei, et al. "Ssd: Single shot multibox

detector," Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016.

[10] Lin, Tsung-Yi, et al. "Focal loss for dense object detection," Proceedings of the IEEE international conference on computer vision, 2017.

[11] Choukroun, Yoni, et al. "Low-bit quantization of neural networks for efficient inference," 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), IEEE, 2019.

[12] Jacob, Benoit, et al. "Quantization and training of neural networks for efficient integer-arithmetic-only inference," Proceedings of the IEEE conference on computer vision and pattern recognition, 2018.

[13] Zhuang, Bohan, et al. "Towards effective low-bitwidth convolutional neural networks," Proceedings of the IEEE conference on computer vision and pattern recognition, 2018.

[14] Zhou, Shuchang, et al. "Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients," arXiv preprint arXiv:1606.06160 (2016).