사물인터넷융복합논문지 Vol. 9, No. 1, pp. 9-17, 2023 https://doi.org/10.20465/KIOTS.2023.9.1.009

초소형 IoT 장치에 구현 가능한 딥러닝 양자화 기술 분석

김영민[1], 한경현[2], 황성운[3*]
[1]가천대학교 IT융합공학과 학생, [2]홍익대학교 전자전산공학과 학생, [3]가천대학교 컴퓨터공학과 교수

Analysis of Deep learning Quantization Technology for Micro-sized IoT devices

YoungMin KIM[1], KyungHyun Han[2], Seong Oun Hwang[3*]
[1]Student, Department of IT Convergence Engineering, Gachon University
[2]Student, Department of Electronics and Computer Engineering, Hongik University
[3]Professor, Department of Computer Engineering, Gachon University

요 약 많은 연산량을 가진 딥러닝은 초소형 IoT 장치나 모바일 장치에 구현하기가 어렵다. 최근에는 이러한 장치에서도 딥러닝을 구현할 수 있도록 모델의 연산량을 줄이는 딥러닝 경량화 기술이 소개되었다. 양자화는 연속적인 분포를 가지는 파라미터 값들을 고정된 비트의 이산 값으로 표현하여 모델의 메모리 및 크기 등을 줄여 효율적으로 사용할 수 있는 경량화 기법이다. 그러나 양자화로 인한 이산 값 표현으로 인해 모델의 정확도가 낮아지게 된다. 본 논문에서는 정확도를 개선할 수 있는 다양한 양자화 기술을 소개한다. 먼저 기존 양자화 기술 중 APoT와 EWGS를 선택하여 동일한 환경에서 실험을 통해 결과를 비교 분석하였다. 선택된 기술은 ResNet모델에서 CIFAR-10 또는 CIFAR-100 데이터 세트로 훈련되고 테스트 되었다. 실험 결과 분석을 통해 기존 양자화 기술의 문제점을 파악하고 향후 연구에 대한 방향성을 제시하였다.

주제어 : 사물인터넷, 딥러닝, 양자화, 모델 훈련, 실험 구성

Abstract Deep learning with large amount of computations is difficult to implement on micro-sized IoT devices or moblie devices. Recently, lightweight deep learning technologies have been introduced to make sure that deep learning can be implemented even on small devices by reducing the amount of computation of the model. Quantization is one of lightweight techniques that can be efficiently used to reduce the memory and size of the model by expressing parameter values with continuous distribution as discrete values of fixed bits. However, the accuracy of the model is reduced due to discrete value representation in quantization. In this paper, we introduce various quantization techniques to correct the accuracy. We selected APoT and EWGS from existing quantization techniques, and comparatively analyzed the results through experimentations The selected techniques were trained and tested with CIFAR-10 or CIFAR-100 datasets in the ResNet model. We found out problems with them through experimental results analysis and presented directions for future research.

Key Words : Internet of Things, Deep Learning, Quantization, Model Training, Experimental Configuration

이 논문은 2021년도 가천대학교 교내연구비 지원에 의한 결과임.(GCU-202104500001)
*교신저자 : 황성운(sohwang@gachon.ac.kr)
접수일 2022년 11월 3일 수정일 2022년 12월 12일 심사완료일 2022년 12월 14일

1. 소개

최근 IoT 장치나 모바일 장치와 같은 소형 장치에 딥러닝을 구현하여 얼굴 인식, 사물 인식과 같은 다양한 응용에서 사용되고 있다. 특히 MobileNet과 같은 딥러닝 모델은 이러한 작은 장치에서도 최적의 성능을 달성할 수 있도록 구현되었다[1]. 하지만 여전히 딥러닝 모델은 많은 파라미터와 그에 따른 많은 메모리 소비량과 연산량을 동반하기 때문에 이러한 모델은 아직 초소형 장치에 직접 구현하기에는 어려움이 있다. 따라서 이러한 문제를 해결하기 위해 많은 경량화 방법이 소개되었으며 초소형 장치에서도 모델을 사용할 수 있도록 훈련 시키고 생성할 수 있다[2, 3, 4]. 특히 양자화[4]는 경량화 방법의 하나로 이는 연속적인 분포를 가지는 파라미터의 값들을 고정 비트의 이산 값으로 표현하는 방법을 말한다. 예를 들어 32-bit의 부동 소수점 파라미터를 8-bit 이산 값으로 표현하게 된다면 이에 대한 파라미터와 모델 크기를 줄일 수 있으며 메모리 사용 또한 4배 줄일 수 있게 된다. 하지만 bit 수가 낮아질수록 모델이 표현할 수 있는 정보도 잃기 때문에 성능이 제한된다. 따라서 최근에는 정확도 손실을 개선하는 다양한 양자화 연구가 진행되었다. 따라서 본 논문에서는 현재 사용되고 있는 양자화 기술을 조사하고 분석하며 실험에서는 소개된 양자화 기술 중 하나를 선택하여 실험 분석을 진행한다.

본 논문의 구성은 다음과 같다. 2장에서는 양자화 기술의 종류를 소개하며, 3장에서는 각 기술에 대한 분석을 설명하며, 4장에서는 양자화 기술을 선택하여 실험을 진행하고, 5장에서 결론으로 논문을 마무리한다.

2. 양자화 기술 소개

2.1 균일 양자화

균일 양자화[5, 6, 7, 17]는 딥러닝 양자화에서 가장 많이 사용되는 기술 중 하나이다. 훈련된 모델의 가중치와 활성화 분포는 대부분 평균이 0인 정규분포를 따르는데[8], 균일 양자화는 이러한 정규분포에서 특정 범위를 지정하여 양자화 값을 모두 동등한 간격으로 투영한다. 균일 양자화는 반올림 함수와 같은 간단한 식으로 구현을 할 수 있으므로 훈련과 추론의 계산 효율성 측면에서도 우위를 가진다. 평균이 0인 정규분포에서 대부분의 파라미터 값들은 평균 영역에 많이 분포해 있으나, 균일 양자화는 평균 영역에 많은 양자화 값을 투영하지 못하므로 훈련 중 정확도가 떨어지는 단점이 있다.

2.2 불균일 양자화

불균일 양자화[9, 10, 11, 15, 16]는 균일 양자화가 평균 영역의 양자화 값을 제대로 투영하지 못한다는 문제점을 해결한다. 불균일 양자화는 양자화 값의 간격이 일정하지 않으며 그로 인해 평균 영역에 더 많은 양자화 값을 투영할 수 있다. 이로 인해 기존의 정규분포와 더욱 유사하게 표현할 수 있으며 균일 양자화보다 더 나은 정확도를 가져올 수 있다. 하지만 불균일 양자화는 균일 양자화와 달리 평균 영역에 양자화 값을 집중적으로 투영을 해준다. 이 과정에서 추가적인 계산이 포함되며 훈련과 추론 중 많은 계산량으로 인한 계산 오버헤드가 발생하게 된다.

2.3 반올림 근사함수

일반적으로 양자화는 반올림 함수를 사용하여 이산 값을 투영한다. 하지만 반올림 함수는 훈련 중 역전파(Backward)에서 기울기가 0이기 때문에 기울기 소실 문제가 발생하게 된다. 따라서 이러한 문제를 해결하기 위해 훈련 중 반올림 함수와 비슷한 새로운 함수로 훈련하여 기울기가 0이 되는 문제를 방지할 수 있다[12, 13, 14]. 반올림 근사함수는 양자화 값을 사용하여 추론할 때 정확도를 보정하면 되기 때문에 훈련 중 반올림 근사함수를 사용하여 반올림 함수와 유사하게 만들어 훈련하면 양자화 모델을 최적화할 수 있다는 장점이 있다. 따라서 반올림 근사함수는 오직 훈련 중에 사용되며 추론에는 기울기 계산을 통한 파라미터 업데이트를 하지 않기 때문에 반올림 함수를 사용하여 양자화 값을 투영하고 추론한다. 하지만 균일 양자화와 유사한 기술이며 불균일 양자화의 성능을 뛰어넘지 못한다.

3. 양자화 기술 분석

앞서 소개된 각 양자화 기술은 양자화 인식 훈련과 훈련 후 양자화를 기본 베이스로 사용한다. 특히 두 방법은 양자화 적용을 위해 자유롭게 선택을 할 수 있으며 각 방법에 따라 장단점이 있다.

3.1 양자화 인식 훈련

양자화 인식 훈련[13, 14, 15, 16, 17]은 딥러닝 모델

이 훈련 과정에서 양자화를 함께 적용하면서 훈련하는 방법이다. 양자화 인식 훈련은 모델의 파라미터가 양자화된 상태로 정확히 추론할 수 있게 학습되기 때문에 비교적 높은 정확도를 가져올 수 있다. 양자화를 모델에 적용하여 훈련하는 방법은 기존의 훈련 방법과 같으며 특히 순전파(Forward)에서 양자화 프로세스가 추가된다. 훈련마다 모델의 가중치와 활성화 값은 양자화 프로세스를 통해 양자화 값으로 투영되며 투영된 양자화 값으로 모델을 계산하고 업데이트한다. 하지만 역전파 과정에서 모델 업데이트를 위한 각각의 값에 대한 기울기를 계산할 때 양자화 프로세스 (e.g. 반올림 함수)에 대한 기울기는 0이 되기 때문에 기울기 소실 현상이 발생하여 훈련이 진행되지 않는 문제가 발생할 수 있다. 따라서 대부분의 양자화 인식 훈련 연구에서는 양자화 프로세스에 대한 기울기를 무시하여 전파하는 STE (Straight Through Estimator) 기법[18]을 사용한다. STE는 양자화 프로세스에 대한 기울기를 항등 함수(e.g. 1)로 전파하여 기울기가 0이 되는 문제를 해결할 수 있다. STE 사용으로 양자화 인식 훈련은 대부분 높은 정확도 성능을 달성한다.

3.1.1 EWGS

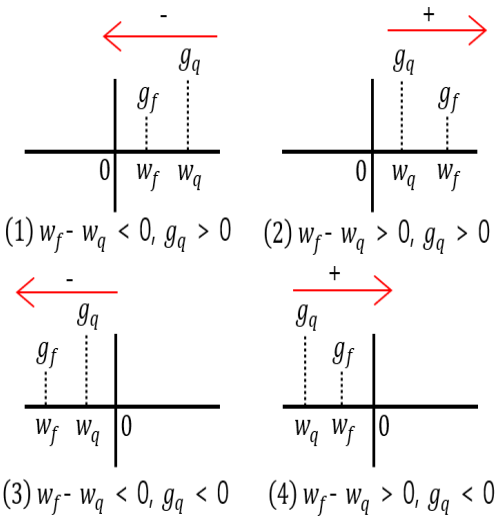
EWGS(Element-wise Gradient Scaling)[17]는 기본적으로 균일 양자화에 맞추어져 있으며 역전파의 기울기를 Scaling 하여 최적화를 하는 방법을 제안했다. 양자화 인식 훈련에서 STE는 기울기를 무시하여 전파해 기울기가 모두 0이 되는 문제를 해결할 수 있지만 이는 올바른 기울기가 아니라고 한다. 양자화 전과 후의 차이로 인해 기울기 차이 또한 발생할 수 있으며 만일 역전파에서 양자화 후의 기울기를 양자화 전 기울기로 Scaling 할 수 있으면 모델이 올바른 방향으로 훈련될 수 있다고 한다.

$$g_{new} = g_q * (1 + \lambda sign(g_q)) * (w_f - w_q) \quad (1)$$

여기서 g_q는 역전파에서 STE로 인해 계산되는 기울기이며 λ는 Scaling Factor 값이다. EWGS는 기존 연속 값과 이산 값의 차($w_f - w_q$)를 사용하여 각 기울기가 얼마나 떨어져 있는지를 확인한다. 그리고 부호 함수 $(sign)$를 통해 g_q의 부호를 결정할 수 있으며 $(1 + \lambda sign(g_q)) * (w_f - w_q)$로 인해 기울기가 Scaling 되는 방향을 정할 수 있다. [Fig. 1]은 EWGS에서의 기울기 Scaling 방법을 조금 더 자세하게 설명하였다. 그리고 Scaling Factor(λ)는 매 훈련 업데이트되는데 이는 모델이 훈련할 때 생성되는 기울기를 통해 계산된다. Scaling Factor(λ)는 다음과 같이 계산된다.

$$\lambda = \frac{Tr(H)/N}{G} \quad (2)$$

식 2에서 G는 각 훈련에서 계산되는 기울기(g_q)이며 Tr(H)은 기울기의 2차 도함수로 이루어진 Hessian Matrix에서 허친슨 방법(Hutchinson's method)[19]을 사용하여 값을 추적할 수 있다. N은 Hessian Matrix에서의 대각선의 개수이다. 따라서 EWGS는 역전파에서 매 훈련 Scaling Factor(λ)를 구할 수 있으며 이를 통해 기울기가 Scaling 되어 전파되며 모델은 올바른 기울기를 사용하여 업데이트를 진행할 수 있다.



[Fig. 1] Gradient scaling scheme for EWGS. g_f is the gradient to the existing weight value and is equal to the final $g_{\neq w}$ when the quantization gradient g_q is scaled. In particular, in the case of $w_f - w_q > 0$, it can be seen that g_q is scaled in the +(plus) direction, and vice versa, it is scaled in the -(minus) direction.

3.1.2 PoT

$$Q^w(\alpha, b) = \alpha \times g$$
$$\text{where } g \in \{0, 2^{-i}, ..., 1\}, 0 \le i < 2^b - 1 \quad (3)$$

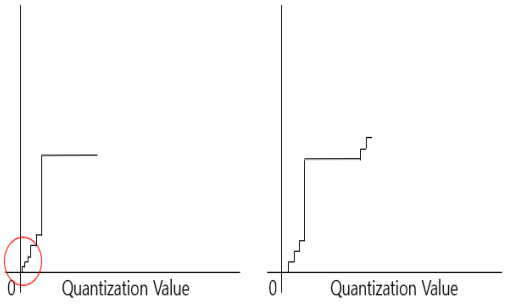
PoT(Powers-of-Two)[15]는 불균일 양자화 기법이다. 불균일 양자화는 평균 영역에 양자화 값을 투영시키기 위해 추가로 계산이 필요하므로 계산 오버헤드가 발생한다. 하지만 PoT[15]는 이러한 문제를 해결하기 위해 모든 양자화 값을 2의 거듭제곱으로 표현하였으며 평균 영역에 쉽게 투영할 수 있다. 그리고 2의 거듭제곱으로 표현된 양자화 값은 모델 훈련 또는 추론 시 $shift$ 연산로 대체되며 이를 통해 계산 효율성 또한 좋아진다는 장점이 있다. 하지만 PoT[15]에서는 bit 폭이 커질수록 평균 영역에 더 작은 값이 투영될 수 있다. 예를 들어, 3-bit에서 계산된 양자화 값은 $\{0, 2^{-1}, 2^{-2}, 1\}$로 표현되지만 4-bit 이상에서는 $\{0, 2^{-1}, 2^{-2}, 2^{-3}, ..., 1\}$로 양자화 값이 평균에 점차 가까워진다는 것을 알 수 있다. 그리고 양자화 값에 대한 간격은 해상도를 따르며, PoT[15]의 높은 bit에서는 매우 작은 양자화 간격으로 인해 [Fig. 2]의 (1)과 같이 양자화 값이 평균값과 가장 가깝게 형성되는 경직된 해상도 문제(Rigid resolution problem)가 발생할 수 있다.

3.1.3 APoT

$$Q^w(\alpha, kn) = \alpha \times \left\{ \sum_{i=0}^{n-1} g_i \right\}$$
$$\text{where } g_i \in \left\{ 0, \frac{1}{2^i}, \frac{1}{2^{i+n}}, ..., \frac{1}{2^{i+(2^k-2)n}} \right\} \quad (4)$$

APoT(Additive Powers-of-Two)[16]는 PoT[15]에서 발생하는 경직된 해상도 문제를 해결하기 위해 나온 불균일 양자화 방법이다. 기존의 2의 거듭제곱만을 사용하는 PoT[15]방법과 달리, APoT[16]에서는 2의 거듭제곱을 합으로 표현한다. 따라서 평균 영역에 양자화 값이 몰리지 않고 범위 내에서 골고루 투영된다. APoT[16]는 PoT[15]와 마찬가지로 2의 거듭 제곱을 사용하기 때문에 $shift$ 연산으로 대체되기 때문에 계산 효율성이 유지된다.

APoT[16]와 PoT[15]의 비교는 [Fig. 2]에 설명되어 있다.

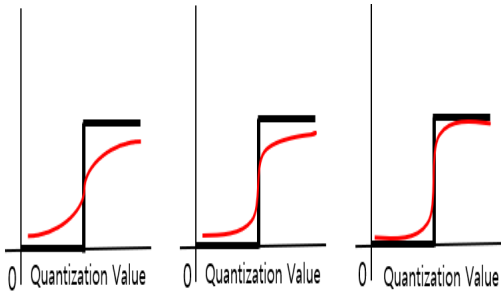


(1) PoT (2) APoT

[Fig. 2] Comparison of PoT and APoT. In PoT, the quantization values are distributed around the mean area (Refer to the red circle), and small quantization intervals can cause rigid resolution problems. On the other hand, in APoT, quantization values are uniformly distributed in the mean and tail.

3.1.4 DSQ

DSQ(Differentiable Soft Quantization)[13]는 반올림 근사 방법을 사용하여 양자화를 진행하는 기법이다. DSQ는 훈련 중 모든 실숫값을 Tanh 함수에 투영한다. 그리고 Tanh 함수는 훈련이 진행될수록 점차 반올림 함수에 근접하며 매 훈련 서로 다른 기울기를 전파할 수 있다. 추론 시 기울기를 계산하지 않기 때문에 반올림 근사함수로 훈련된 모델은 양자화 프로세스에서 반올림 함수를 사용하였을 때 정확도를 보정할 수 있다.



(1) Epoch 1 (2) Epoch 2 (3) Epoch n

[Fig. 3] The DSQ approaches the Tanh function as a rounding function for each epoch, where we can see what is the most approximate to the rounding function.

하지만 역전파에서 기울기를 전파할 때 Tanh 함수에 대한 기울기를 전파하게 되는데 훈련이 진행될수록 반올

림 함수에 근접할수록 기울기가 작아져 기울기 소실 문제가 발생할 수 있다는 단점이 있다.

3.1.5 DAQ

DAQ(Distance-Aware Quantization)[14]는 비선형함수를 사용하지 않고 양자화 값과 실숫값의 거리를 사용하여 반올림 근사함수를 만들어 양자화 훈련을 하는 기법이다. 양자화 값과 실숫값의 거리를 사용하기 위해 거리점수를 사용하며 식은 다음과 같다.

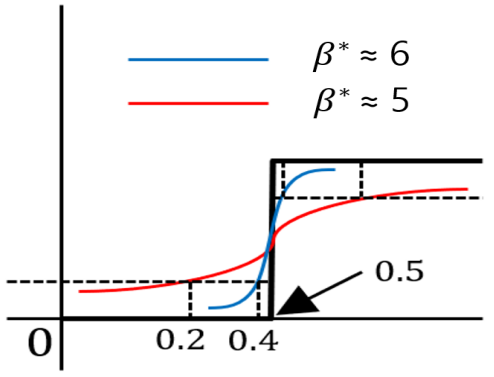
$$d_x(q) = \exp(-|x - q|) \quad (5)$$

식 (5)에서 x는 기존의 실숫값이며 q는 양자화 값이다. 예를 들어 실숫값 0.4과 0.2는 모두 0으로 양자화가 되며 그에 따른 거리점수를 계산할 때 $d_{0.4}(q) \approx 0.67$, $d_{0.2}(q) \approx 0.81$로 계산됨을 알 수 있다. 따라서 두 값의 거리가 가까울수록 거리점수는 높아지며 다음 식을 통해 반올림 근사함수를 새로 생성할 수 있다.

$$\beta^* = \frac{\gamma}{|s_x(q_f) - s_x(q_c)|} \quad (6)$$

DAQ에서는 온도(β^*) 개념을 도입하였다. 이는 모델이 훈련될 때마다 실숫값에 따른 점수($s_x(q_f)$)와 양자화 값에 따른 점수($s_x(q_c)$)를 각각 계산하여 유동적으로 반올림 함수와 근사하는 그래프를 새로 생성할 수 있다. $s_x(q_f)$와 $s_x(q_c)$는 식 (5)에서 계산된 값을 따르며 $s_x(q_i) = k_x(q_i)d_x(q_i)$로 계산된다. 여기서 $k_x(q_i)$는 $d_x(q_x)$만으로 생성된 함수는 미분할 수 없으므로 Kernel soft argmax[25] 함수를 추가함으로써 미분 함수를 생성할 수 있다. 예를 들어, [Fig. 4]에서 실숫값 0.4과 0.2에 대해 $k_x(q_i)$가 0.5로 계산될 때 식 (6)에 따라 ($|s_x(q_f) - s_x(q_c)|$)는 각각 0.34와 0.41로 계산된다. γ는 고정된 정수이며 최종적으로 γ = 2로 온도가 계산될 때 0.4 ($\beta^* \approx 6$)은 0.2 ($\beta^* \approx 5$)보다 온도가 높은 것을 알 수 있다. 온도가 낮은 경우에는 반올림 근사함수가 완만하게 표현되며 그 반대의 경우에는 반올림 함수와 가장 근접하는 그래프가 생성된다.

따라서 DAQ의 반올림 근사함수는 역전파 동안에 서로 다른 기울기를 생성할 수 있으며 각각의 양자화 값에 대한 최적의 기울기를 찾을 수 있다.



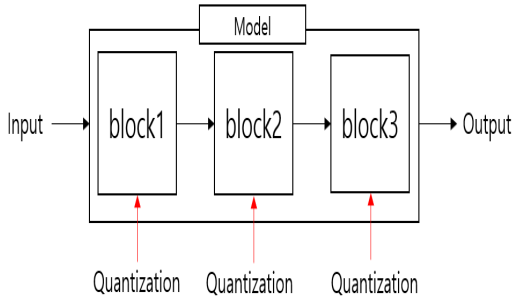
[Fig. 4] Expression of the rounding approximation function at each temperature(β^*) when γ=2. The further the distance from the quantization value, the higher the temperature.

3.2 훈련 후 양자화

훈련 후 양자화[20, 21, 22]는 모든 훈련이 끝난 추론 모델을 소량의 데이터 세트만을 사용하여 모델을 양자화하는 방법이다. 처음부터 훈련을 진행하는 양자화 인식 훈련과 달리 훈련이 완료된 모델을 가지고 미세조정만을 진행하여 최적화를 진행하기 때문에 빠르게 양자화 모델을 만들 수 있다는 장점이 있다. 하지만 양자화 인식 훈련보다 정확도가 떨어지는 단점이 있다.

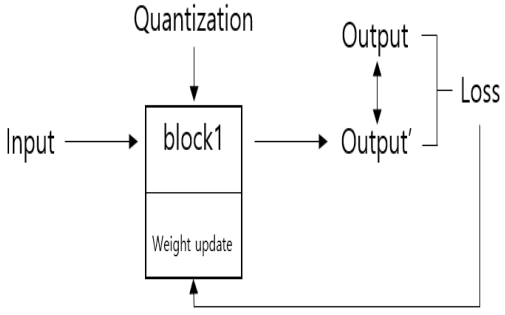
3.2.1 BRECQ

BRECQ(Block Reconstruction Quantization)[22]는 기존의 양자화 인식 훈련과 달리 훈련 후 양자화 방법에 속한다. BRECQ는 훈련된 모델에 소량의 데이터를 사용하여 모델 계층의 집합인 블록마다 미세조정을 하여 양자화를 진행한다.



[Fig. 5] Structure of BRECQ quantization

BRECQ에서 훈련이 완료된 모델은 블록마다 최적의 값을 가지고 있으며 소량의 데이터를 통해 각 블록에 대한 출력값과 양자화 후 출력값을 계산할 수 있다. 그 후 두 값을 통해 Loss 값을 계산하며 각 블록에 대해 업데이트를 진행하고 최적화를 할 수 있다. 예를 들어, [Fig. 6]과 같이 첫 번째 블록(block 1)에 대해 양자화를 진행할 때 Input 데이터를 받아 양자화를 한 후 첫 번째 블록의 출력값(Output')을 계산한다. 이때 출력값은 양자화 값과 Input으로 인해 계산된 값이다. 그리고 해당 모델은 이미 훈련이 완료된 모델이기 때문에 기존의 첫 번째 블록(block 1)에 대한 출력값(Output)을 알 수 있다. 이를 통해 양자화 출력값과 비교하여 Loss를 구한다. 그 후 첫 번째 블록(block 1)에 대해 역전파를 진행할 수 있으며 첫 번째 블록을 업데이트한다. 이와 같은 방법으로 모든 블록에 대해 양자화 최적화를 할 수 있다.



[Fig. 6] First block quantization process

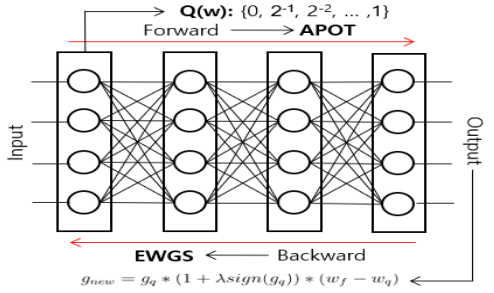
4. 양자화 기술 실험

4.1 실험 방법

우리는 양자화 기술에 대해 실험을 위해 소개된 양자화 기술을 선택하여 실험하였다. 특히 계산 효율성이 좋으면서 그와 동시 정확도를 보정 할 수 있는 불균일 양자화 기법인 APoT[16]를 사용하였다. 또한 APoT[16]는 기본적으로 역전파에서 STE[18] 기법을 사용하기 때문에 우리는 EWGS[17] 기법을 추가로 사용하여 정확도 성능을 분석하였다. 따라서 최종적으로 APoT+EWGS를 구현하고 기존의 APoT와 정확도를 비교하였다.

4.1.1 APoT + EWGS 구현

우리는 [Fig. 7]와 같이 APoT+EWGS의 프로세스를 표현하였다. 먼저 순전파에서는 APoT[16] 기법을 사용하여 가중치와 활성화 함수에 대해 모두 불균일 양자화를 적용하였다. 이로 인해 모든 실수는 이산 값으로 표현되어 계산된다. 그 후 역전파에서는 STE[18] 기법 대신 EWGS[17] 기법을 사용하였다. EWGS[17]는 STE[18]에 대한 기울기 불일치 문제를 해결하여 성능향상을 달성할 수 있으므로 APoT[16] 기법에서도 좋은 효과를 가져올 수 있다.



[Fig. 7] Architecture of APoT+EWGS

4.1.2 실험 세팅

우리는 양자화를 적용하기 위해 우선 32-bit로 훈련이 진행된 사전 훈련된 모델을 불러와 양자화를 적용하였다. 사용된 모델은 ResNet-20과 ResNet-32[23]이며 데이터 세트는 CIFAR-10과 CIFAR-100[24]을 사용하였다. ResNet-20에 대해 CIFAR-10을 사용하며 ResNet-32에 대해서는 CIFAR-100을 사용하였다. 순전파에서 APoT에 대한 양자화 bit는 가중치와 활성화에 대해 2-bit만을 사용하였다. 역전파에서 EWGS에 대한 Scaling factor(λ)값은 초기에 0으로 설정이 되며 1 epoch 동안에는 STE로 훈련이 되며 매 훈련 Scaling factor(λ)가 업데이트되어 기울기가 Scaling 된다. 두 모델에 대해 모두 300번 테스트를 진행하였다. 또한 batch_size를 128, learning_late는 4e-2, weight decay는 1e-4를 32-bit와 2-bit에 모두 적용하였다.

4.1.3 실험 결과

〈Table 1〉 CIFAR10 on ResNet20

Method	Precision (W/A)	Accuracy(%) Top-1
FP.(ResNet-20)	32/32	91.80
APoT	2/2	90.24
APoT + EWGS	2/2	89.88

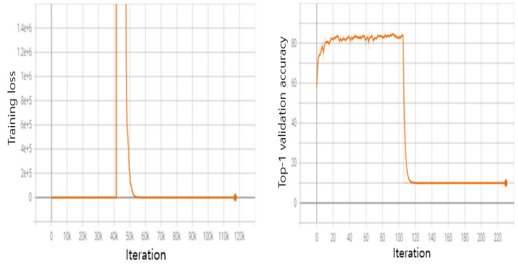
〈Table 2〉 CIFAR100 on ResNet32

Method	Precision (W/A)	Accuracy(%) Top-1
FP.(ResNet-32)	32/32	70.03
APoT	2/2	67.41
APoT + EWGS	2/2	67.33

〈Table 3〉 Repeat training on ResNet-20

Method	Precision (W/A)	Accuracy(%) Trial 1	Trial 2	Trial 3
FP.(ResNet-20)	32/32	91.80	-	-
APoT	2/2	90.24	-	-
APoT + EWGS	2/2	89.88	89.60	85.58

〈Table 1〉과 〈Table 2〉는 각각 ResNet-20과 ResNet-32 결과를 나타낸 것이다. 실험 결과에서 ResNet-20 모델에 APoT는 2-bit에서 32-bit에 비해 정확도가 약 1.57% 차이가 나는 것을 확인할 수 있었다. 하지만 APoT[16]와 EWGS[17]를 함께 적용하면 정확도 성능이 감소하는 것을 확인할 수 있었다. 이는 ResNet-32에서도 같은 결과를 보이는데 EWGS[17]를 적용한 APoT[16]는 기존의 APoT[16]에 비해 정확도가 약 0.08% 감소하였다. 특히 〈Table 3〉의 ResNet-20에서 실험을 여러 번 반복하였을 때 대부분 정확도가 기존보다 매우 감소하는 현상을 보였다.



(1) Training loss (2) Validation accuracy

[Fig. 8] Training loss and validation accuracy for APoT+EWGS. The validation accuracy of APoT+EWGS significantly decreases as the training loss explodes after a few epoch.

4.1.4 한계 및 토론

이번 절에서는 APoT[16]에 EWGS[17]를 적용하였을 때 개선되지 않는 이유를 분석한다. 이 문제는 EWGS[17]의 Scaling Factor(λ)를 구할 때 발생하는 문제로 파악된다. 우선 식 3.1.3에서 APoT는 양자화 값으로 0을 사용하는 것을 알 수 있다. 하지만 기울기를 계산할 때 양자화 값에 0이 존재할 때 3.1.1의 EWGS[17] 식에서 원치 않는 기울기를 얻을 수 있다. 그리고 식 2에서 모델에 대한 기울기들의 2차 도함수 집합인 Hessian Matrix를 사용하여 Scaling Factor(λ)를 계산할 때 0 값으로 인해 Scaling Factor(λ) 값이 매우 커지게 되며 그로 인해 scaling 되는 기울기 값 또한 매우 커지게 된다. 그리고 기울기뿐만 아니라 Loss 값에 대해서도 문제가 드러난다. [Fig. 8]에서 APoT+EWGS에 대해 Loss 값을 분석하였으며 APoT+EWGS는 훈련 중 Loss 값이 매우 커져 정확도가 떨어진다는 것을 알 수 있었다. 따라서 EWGS[17]는 기울기를 Scaling 함으로써 좋은 성능을 가져올 수 있지만 APoT[16]와 같이 양자화 값이 0을 포함하게 되면 심각한 정확도 저하를 초래할 수 있다고 분석된다.

5. 결론

본 논문은 딥러닝 모델의 경량화를 위한 양자화의 개념과 다양한 기술을 소개하였다. 그리고 우리는 소개된 양자화 기술 중 불균일 양자화인 APoT[16]를 선택하여 실험을 진행하였으며 결과에 대해 분석을 진행하였다. APoT[16]는 역전파 중 STE[18]를 사용하기 때문에 양자화에 대한 기울기 불일치 문제가 발생할 수 있으며 우리는 EWGS[17]로 대신하여 사용하였다. 실험 결과 APoT[16]에 EWGS[17]를 적용하는 것은 기존의 APoT[16]보다 정확도가 낮은 것을 확인할 수 있었다. 이는 Scaling Factor 계산 식에서 2차 도함수인 Hessian Matrix를 사용할 때 APoT[16]의 양자화 값인 0으로 인해 성능에 크게 영향을 미칠 수 있다는 것을 확인했다. 따라서 본 실험을 바탕으로 향후 양자화 체계를 적용할 때 역전파의 기울기 계산을 참고하여 최적의 양자화 값을 분석하고 선택하는 것이 중요함을 알 수 있다.

REFERENCES

[1] Howard, Andrew G., et al. “Mobilenets: Efficient convolutional neural networks for mobile vision applications.” arXiv preprint arXiv:1704.04861, 2017.

[2] Blalock, Davis, et al. “What is the state of neural network pruning?.” Proceedings of machine learning and systems 2, pp.129-146, 2020.

[3] Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." arXiv preprint arXiv:1503.02531, 2015.

[4] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. "Binarized neural networks." Advances in neural information processing systems 29, 2016.

[5] Raghuraman Krishnamoorthi. "Quantizing deep convolutional networks for efficient inference: A whitepaper." arXiv preprint arXiv:1806.08342, 2018.

[6] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. "Quantization and training of neural networks for efficient integer-arithmetic-only inference." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp.2704-2713, 2018.

[7] Hao Wu, Patrick Judd, Xiaojie Zhang, Mikhail Isaev, and Paulius Micikevicius. "Integer quantization for deep learning inference: Principles and empirical evaluation." arXiv preprint arXiv:2004.09602, 2020.

[8] Song Han, 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.

[9] Zhaohui Yang, Yunhe Wang, Kai Han, Chunjing Xu, Chao Xu, Dacheng Tao, and Chang Xu. "Searching for low-bit weights in quantized neural networks." Advances in neural information processing systems 33, pp.4091-4102, 2020.

[10] Kohei Yamamoto. "Learnable companding quantization for accurate low-bit neural networks." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.5029-5038, 2021.

[11] Yunchao Gong, Liu Liu, Ming Yang, and Lubomir Bourdev. "Compressing deep convolutional networks using vector quantization." arXiv preprint arXiv:1412.6115, 2014.

[12] Yang, Jiwei, et al. "Quantization networks." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.7308-7316, 2019.

[13] Gong, Ruihao, et al. "Differentiable soft quantization: Bridging full-precision and low-bit neural networks." Proceedings of the IEEE/CVF International Conference on Computer Vision, pp.4852-4861, 2019.

[14] Kim, Dohyung, Junghyup Lee, and Bumsub Ham. "Distance-aware quantization." Proceedings of the IEEE/CVF International Conference on Computer Vision, pp.5271-5280, 2021.

[15] Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, and Yurong Chen. "Incremental network quantization: Towards lossless cnns with low-precision weights." arXiv preprint arXiv:1702.03044, 2017.

[16] Yuhang Li, Xin Dong, and Wei Wang. "Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks." In International Conference on Learning Representations, 2020.

[17] Lee, Junghyup, Dohyung Kim, and Bumsub Ham. "Network quantization with element-wise gradient scaling." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp.6448-6457, 2021.

[18] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. "Estimating or propagating gradients through stochastic neurons for conditional computation." arXiv preprint arXiv:1308.3432, 2013.

[19] Avron, Haim, and Sivan Toledo. "Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix." Journal of the ACM (JACM), Vol.58, No.2, pp.1-34, 2011.

[20] Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. "Improving post training neural quantization: Layer-wise calibration and integer programming." arXiv preprint arXiv:2006.10518, 2020.

[21] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. "Data-free quantization through weight equalization and bias correction." In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp.1325-1334, 2019.

[22] Li, Yuhang, et al. "Brecq: Pushing the limit of post-training quantization by block reconstruction." arXiv preprint arXiv:2102.05426, 2021.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp.770-778, 2016.

[24] Alex Krizhevsky, Geoffrey Hinton, et al. "Learning multiple layers of features from tiny images." 2009.

[25] Lee, Junghyup, et al. "Sfnet: Learning object-aware semantic correspondence." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.2278-2287, 2019.

김 영 민(YoungMin KIM) [준회원]

■ 2021년 2월 : 홍익대학교 컴퓨터정보통신공학과(공학학사)
■ 2021년 3월 ~ 현재 : 가천대학교 일반대학원 IT융합공학과 (공학석사과정)

〈관심분야〉
딥러닝, 모델 경량화

한 경 현(Kyung Hyun Han) [정회원]

- 2015년 2월 : 홍익대학교 컴퓨터정보통신학과(공학학사)
- 2017년 2월 : 홍익대학교 일반대학원 전자전산공학과 (공학석사)
- 2017년 3월 ~ 현재 : 홍익대학교 일반대학원 전자전산공학과 (공학박사과정)

〈관심분야〉
머신러닝, 사이버보안

황 성 운(Seong Oun Hwang) [정회원]

- 1993년 8월 : 서울대학교 수학과 (이학사)
- 1998년 2월 : 포항공과대학교대학원 정보통신학과 (공학석사)
- 2004년 8월 : 한국과학기술원 전자전산학과 (공학박사)
- 2006년 1월 ~ 2006년 12월 : University of Michigan 박사 후 연구원
- 2008년 3월 ~ 2020년 2월 : 홍익대학교 컴퓨터공학과 교수
- 2020년 3월 ~ 현재 : 가천대학교 컴퓨터공학과 교수

〈관심분야〉
정보보호, 사이버보안, 기계학습