Summary of “Quantization of Generative Adversarial Networks for Efficient Inference"

Abstract

-GANs are great for generating digital content, but they take so much time/energy during inference that it "complicates [...] their deployment on edge devices" (edge devices like a computer)

-Quantization helps reduce that problem

    -What quantization is: "replacing floating-point computation with low-bit integer ones"

However, quantization techniques are more explored for discriminative models than for generative models. So this paper further explores quantization on GANs; tries quantization on GAN architectures like StyleGAN (other two are Self-Attention GAN, and CycleGAN)

Paper results: successfully quantized GAN models for inference "with 4/8-bit weights and 8-bit activations"

Introduction

-GANs especially take "tremendous computational and memory resources spent during the inference phase". Because of this, hard to deploy GANs where memory/latency matter

-Techniques for compression of deep neural networks:

1) pruning

2) efficient architecture design

3) knowledge distillation

4) quantization (particularly researched because it is "'orthogonal' to the majority of other compression techniques" meaning it can be used alongside them. Also because quantization reduces energy consumption.)

Introduction (cont.)

Most current work on quantization techniques has to do with image classification, not generative models. Quantizing generative models is more challenging and not as much researched.

This paper contributes to research on quantizing generative models by testing how effective post-training quantization techniques are, particularly on 3 GAN architectures: self-attention GAN, cycle-consistent GAN, and style-based GAN. This paper reports "first time report successful 4-bit quantization of these models."

Related Work

-GANS; mentions "three architecturally distinct variants of GANs, which are CycleGAN, SAGAN, and StyleGAN.

-For style-based GAN architecture, they state that the generator architecture of "providing latent codes inside adaptive instance normalization blocks instead of the first convolutional layer" improves generated image quality AND increases latent space disentanglement. And they study how quantization affects those style-based generator characteristics.

-Mention previous work on quantization of GANs

-one method “cannot directly be used to reduce latency” but could be used for weight compression

-A more relevant method was “uniform quantization to both weights and activations” where they combined quantization with pruning and knowledge distillation. That paper was not as specific about GANs, focusing more on overall compression techniques

Methodology

Design choices for quantization:

- -Uniform & non-uniform quantization
- -Static & dynamic quantization
- -Per-channel and per-tensor weight quantization
- -Symmetric and asymmetric quantization
- -Post-training quantization
- -Quantization-aware training

Uniform & non-uniform quantization

-Uniform quantization “maps values of weights and activations to a uniform grid of fixed-point representations” (in other words, maps the existing full-precision numbers to simpler, quantized versions of those numbers)

$$x_q = \Delta \cdot \text{clamp}\left(\text{round}\left(\frac{x}{\Delta}\right), t_{\min}, t_{\max}\right) = \Delta \cdot x_{int},$$

-x = floating point number, x_q = quantized value, x_{int} = integer representation

-Non-uniform quantization also maps values to quantized values, but the mapping is to a “non-equidistant grid of quantized values” which is harder to deploy because of hardware limitations

This paper only considers uniform quantization because of the hardware difficulty.

Static & dynamic quantization

-dynamic: computes during inference (recomputes quantization parameters for each batch of data)

-static: "uses precomputed quantization parameters for activations" --> more efficient

-paper focuses on static quantization for efficiency

Per-channel and per-tensor weight quantization

-per-channel quantization "assigns distinct parameters (quantization scales) to each convolutional kernel, enabling flexible quantization of weights"

-per-channel also doesn't slow down computations very much

-paper focuses on per-channel quantization for GANs

Symmetric and asymmetric quantization

Asymmetric quantization involves adding some integer to the uniform-mapping formula (insert formula here)

-allows "modeling non-zero-centered distributions of weights and activations more effectively" (since adding the integer can change the center of the distribution)

-however, adds computational cost (small cost for asymmetric quantization of activations, but possibly significant for weights)

Post-training quantization

Two approaches:

1) Vanilla PTQ

2) A technique similar to BRECQ technique; "block-wisely optimize weights and quantizes by matching activations' feature maps to the ones of the full-precision model"

-STE-BRECQ which uses "straight-through estimator (STE) and learned step size quantization (LSQ) for differentation through rounding operation"

-AR-BRECQ which uses adaptive rounding technique for weights quantization

Quantization-aware training

-results in higher quality results

Experimental Results

-Post-training quantization:

These results are for quantization of a style-based generator

compare between: vanilla PTQ, STE-based BRECQ, AR-BRECQ, and full-precision (full precision is the non-quantized case to compare against)

-best performance (lowest FID / qFID) = from STE-BRECQ for 8-bit weights

-for 4-bit weights, best performance was by AR-BRECQ

-paper explains this result due to adaptive rounding

-Overall results for post-training quantization is that it provides good quality already, but to go further, need quantization-aware training.

Experimental Results

Quantization-aware training:

-Compared between: MA-QAT, LSQ-QAT

-even lower (better) FID/qFID scores than the post-training quantization results

Discussion and Conclusion

-The 3 GAN architectures mentioned in the paper (StyleGAN, Self-Attention GAN, and CycleGAN) can be quantized for 4/8-bit inference without much quality degradation

Reference

Andreev, P., & Fritzler, A. (2022, August). Quantization of generative adversarial networks for efficient inference: A methodological study. In *2022 26th International Conference on Pattern Recognition* (ICPR) (pp. 2179-2185). IEEE.