Proposal on QAT LLM quantization

# **Introduction**

From Information Theory LLM is a discrete object. So considering quantization noise as channel error as long as LLM capacity as channel is higher than data rate it can be quantized almost without degradation. The problem is the weights optimization for the discrete object has exponential complexity while SGD training methods which consider LLM as continuous have polynomial complexity. But continuity means an infinite accuracy and redundant bits for LLM FP weights and activations. One approach is just adding some quantization noise (PTQ), but this always distorts activation distributions and reduces quality. PTQ directly operates on LLM weights and activations without or with very small amount of training data, so it has low complexity.

Discrete-continuous contradiction can be solved by considering discrete (quantized) LLM as continuous object with noise. Similarly to Shannon’s theorem on AWGN[[1]](#footnote-0) channel capacity in this case SNR defines the channel bit-width. SGD can still be applied to continuous part of LLM giving reasoning to QAT methods.

QAT can be fixed-width and mixed-width. In the latter case the optimal bit-width is selected for each channel based on quality. Mixed-width QAT also covers pruning because it corresponds to zero bit quantization of pruned channels.

Unfortunately, T in QAT means training. LLM training is prohibitive both due to computations and amount and availability of training data. So for practical purposes it is necessary to develop data-free QAT with similar complexity to PTQ.

# **Problem statement**

## **What we have**

### **LLM PTQ**

Pro: PTQ does not need training and training data.

Contra: PTQ reduces NN quality as it distorts activation distributions due to quantization error.

### **Bit-width differentiable CNN QAT with random noise injection**

Pro: QAT can recover quality if NN capacity is enough. Mixed-precision quantization.

Contra: QAT is the same speed or slower than FP training which is prohibitable for LLM. Training data is needed.

## **What we do not have**

Data-free fast QAT.

# 

# 

# 

# **Proposal**

## **Stage 0. Prerequisites**

1. Identify SLM (Small Language Model) which can be trained for a reasonable amount of time

Training should be reproducible!

* Several transformer-based models can be chosen
  + MobileBERT (25.3M)
  + MobileBERT\_tiny (15.1M)
  + ALBERT\_base (12M)
  + MobiLlama[[2]](#footnote-1)(0.5B)
* Although most of the papers provide benchmarks for fairly large models, such as LLAMA 7B, etc.

1. Setup inference and training pipeline
2. Repeat SOTA training quality, that presented in papers or HF

## **Stage 1. Adapt QAT for SLM**

1. Integrate per-channel mixed-precision QAT into SLM training
2. Different bit-widths may be considered for different parts of the transformer
3. Embedding layers
4. FFN (perceptron) layers
5. Multi-head Attention layers
6. Quantize SLM with various bit-width targets
7. Compare quality with PTQ

## **Stage 2. Data-free QAT**

1. Implement random text generation using pretraining FP SLM by random sampling from generator input distribution with empty prompt
2. Implement SLM data-free distillation using quantized network as student.
3. Integrate distillation and QAT
4. Compare results with stage 1 QAT

## **Stage 3. Forward tiled QAT**

1. Quantize SLM by 3 layer tiles one-by-one starting from input. Use FP SLM tile output as reference output with L1/L2 distance (Hinton-like forward learning)[[3]](#footnote-2).
2. Integrate forward tiled QAT with stage 2 distillation
3. Compare results with stage 1,2 QAT

## 

## 

## **Stage 4. Evaluation**

1. Evaluate on LLMs

# **Quantization Granularity**

To achieve better quantization quality different approaches can be applied to quantization scheme.There are several types of layers with weights that transformer consist of. Basically most of the layers are Linear aka Fully-connected layers that composed in a different ways.

* Embedding
  + Introduced PEG[[4]](#footnote-3) - per-embedding group quantization for embedding activations
* FFN, Multi-Head attention
  + For these types of layers several techniques are exist.
  + per-row quantization - basically each row has it’s own scale factor
  + group-wise quantization - approach used in LUT-GEMM[[5]](#footnote-4), varying amount of scale-factors per-group in specific layer channel.
  + block-wise quantization - approach used in GGUF/GGML, which is similar to group-wise quantization, but with no respect to channels.

**Open Questions**

1. How to generate output embedding for the decoder part of the transformer without input data?
2. How to determine the level of overparametrization for the model? Importance matrix?
3. How do the quantization times of Forward Tiled QAT and PTQ compare?
4. Is finetuning on smaller possibly not well representable dataset is enough for QAT?
5. Differences of language models quantization vs quantization of computer vision models.

1. https://en.wikipedia.org/wiki/Shannon%E2%80%93Hartley\_theorem [↑](#footnote-ref-0)
2. https://arxiv.org/abs/2402.16840 [↑](#footnote-ref-1)
3. https://arxiv.org/abs/2212.13345 [↑](#footnote-ref-2)
4. https://arxiv.org/abs/2109.12948 [↑](#footnote-ref-3)
5. https://arxiv.org/abs/2206.09557 [↑](#footnote-ref-4)