AI 실전 9주차

LLM 모델을 저사양 컴퓨터에
구동하기 위한 양자화

모델 경량화 방법

- **pruning**
  - 0에 가까운 애들 다 지워버리자.
  - fine grained pruning -> structural pruning을 많이 사용함

- **quantization**
  - Float32 -> Int8 변환하는 과정
  - quantization on training : training에도 quantization 처리 하여 사용

- **Knowledge distillation**
  - 원본 DL 모델 (크고 학습많이한 model)이 teacher
  - 이 모델을 다 쓸필요없거나 device에 안맞을 때 사용할 기기 맞는 작은 student model 만듬.

- **low-rank factorization**
  - NxM이 너무 크니까 Nxk @ kxM 으로 줄인 것 (matrix factorization)

수 많은 경량화 기법

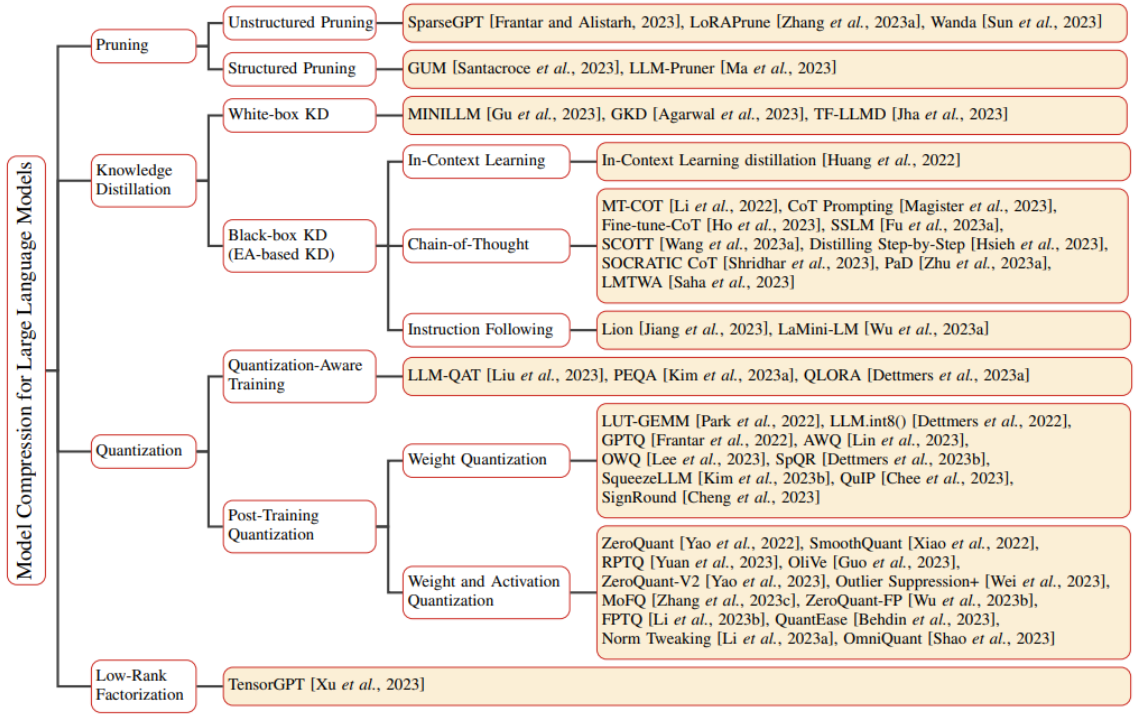
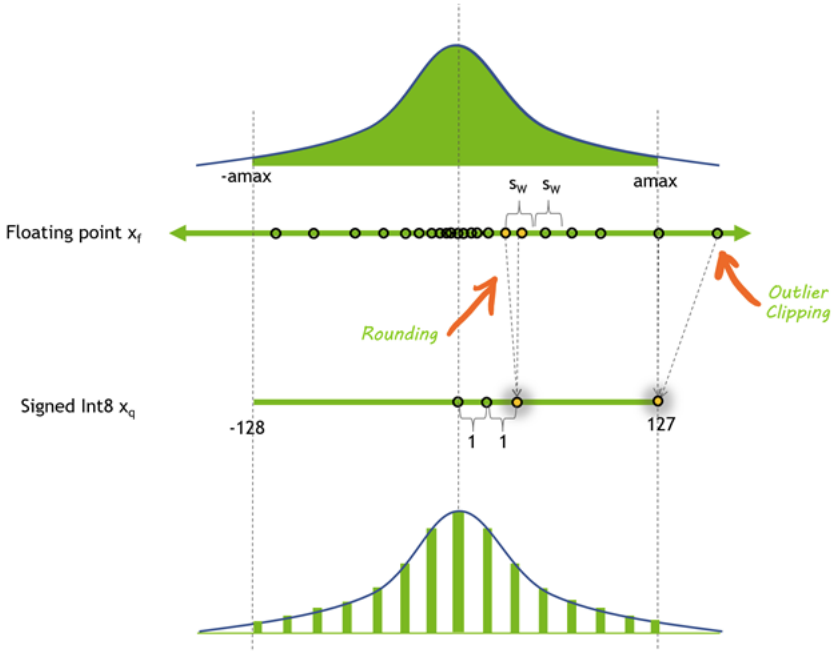


Figure 1: Taxonomy of Model Compression methods for Large Language Models.

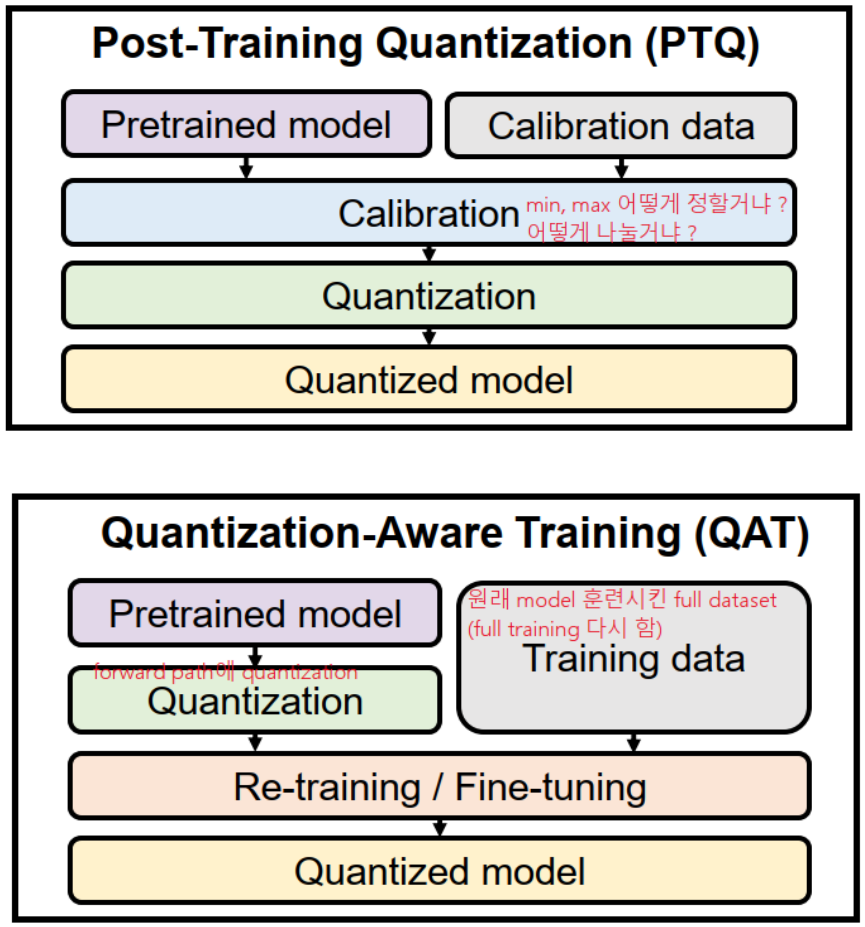
Precision	Methods
8-bit quantization	LUT-GEMM [Park *et al.*, 2022], LLM.int8() [Dettmers *et al.*, 2022], ZeroQuant [Yao *et al.*, 2022], SmoothQuant [Xiao *et al.*, 2022]
lower-bit quantization	LLM-QAT [Liu *et al.*, 2023], PEQA [Kim *et al.*, 2023a], QLORA [Dettmers *et al.*, 2023a], GPTQ [Frantar *et al.*, 2022], AWQ [Lin *et al.*, 2023], SpQR [Dettmers *et al.*, 2023b], RPTQ [Yuan *et al.*, 2023], OliVe [Guo *et al.*, 2023], Outlier Suppression+ [Wei *et al.*, 2023], OWQ [Lee *et al.*, 2023], ZeroQuant-FP [Wu *et al.*, 2023b], ZeroQuant-V2 [Yao *et al.*, 2023], SqueezeLLM [Kim *et al.*, 2023b], QuIP [Chee *et al.*, 2023], FPTQ [Li *et al.*, 2023b], QuantEase [Behdin *et al.*, 2023], Norm Tweaking [Li *et al.*, 2023a], SignRound [Cheng *et al.*, 2023], OmniQuant [Shao *et al.*, 2023]

양자화(quantization)란?



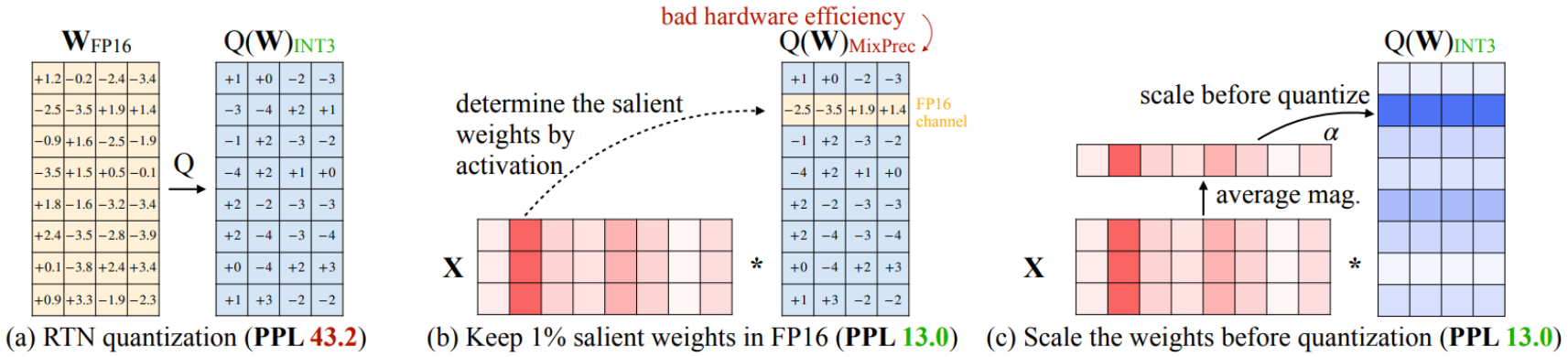
- 복잡한 정보를 작은 단위의 자료형으로 변환하여 저장
- 연속적인 값을 이산적 값으로 표현

양자화(quantization) 종류



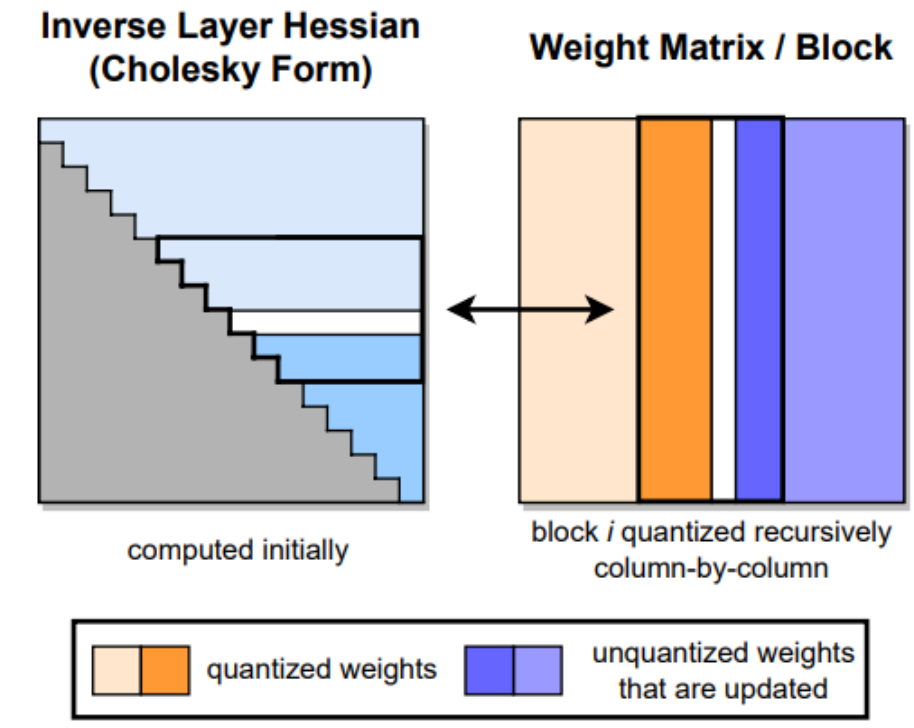
- 모델 학습 후 양자화 (PTQ)
  - 이미 훈련된 LLM을 양자화하는 기술을 의미
  - 모델 정확도가 감소할 수 있음
- 모델 학습 하면서 양자화(QAT)
  - 양자화를 고려하여 데이터를 사용하여 모델을 세밀하게 조정하는 방법
  - 우수한 모델 성능을 제공하지만, 더 많은 계산이 필요

AWQ



- 파라미터의 활성화 비율에 따라 자료형 변환
  - Float16 -> INT4, INT3
- 약 3~5배 가까이 용량 작게 만들 수 있음.

GPTQ(General Pre-Trained Transformer Quantization)



- GPTQ는 출력 오류를 최소화하는 양자화된 가중치(MSE(평균 제곱 오차)를 발견
- 한 번에 모델을 한 층씩 양자화하는 접근 방식인 레이어별 양자화

GGUF

- GPTQ (GGML) 를 기반하여 만든 모델 저장 구조로 기존의 GPU에서 구동만 가능하였으나, 그 외 다른 것도 구동이 가능함
- 여러 버전의 양자화 변환 가능 (2bit 그 마저 가능하다!)

모델 다운로드

```python
import os
from huggingface_hub import snapshot_download

MODEL_ID = "kakaocorp/kanana-nano-2.1b-instruct"
MODEL_NAME = MODEL_ID.split('/')[-1]

snapshot_download(repo_id=MODEL_ID,
                  local_dir=MODEL_NAME,
                  local_dir_use_symlinks=False,
                  revision="main")
```

모델 gguf 변환 후 양자화

```
git clone https://github.com/ggerganov/llama.cpp.git
pip install -r ./llama.cpp/requirements.txt

python llama.cpp/convert_hf_to_gguf.py ./kanana-nano-2.1b-instruct
--outfile ./kanana-nano-2.1b-instruct.gguf
./llama.cpp/quantize ./kanana-nano-2.1b-instruct.gguf ./kanana-nano-2.1b-
instruct.Q5_K_S.gguf q5_k_s
```