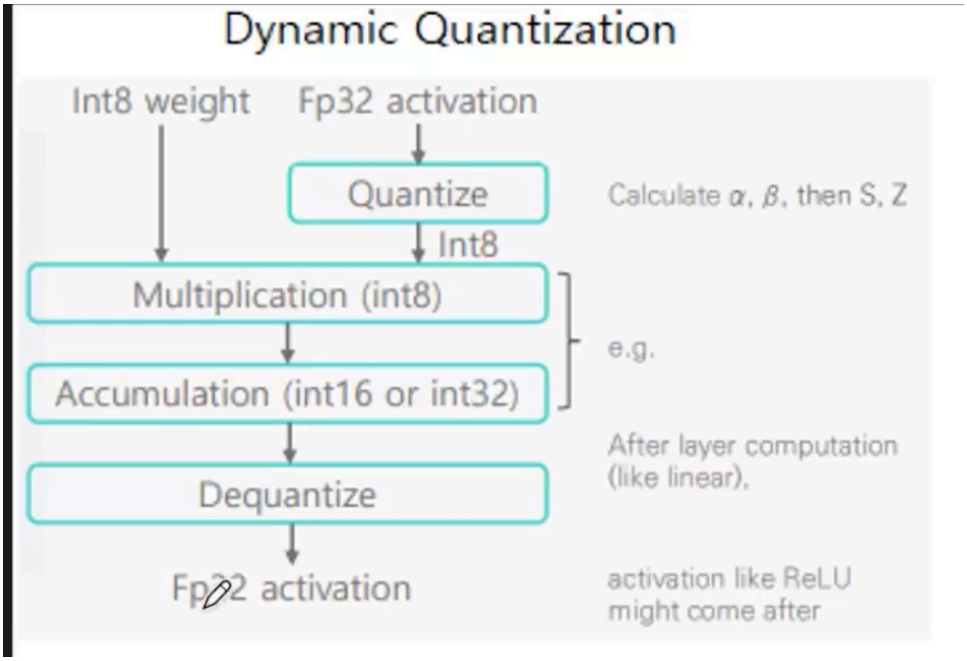
다양한 양자화 기법들 소개

Dynamic Quantization



- 모델의 가중치만을 양자화하고, 활성화 값은 추론 시에만 동적으로 양자화
  (추론 중에만 수행되며, 모델이 학습되지 않은 상태에서 추가적인 양자화 단계를 거치지 않고도 효율적으로 수행)
- 가중치의 양자화
  - Dynamic Quantization에서는 모델의 가중치를 정수(int8)로 변환하여 메모리 효율성 향상
- 활성화값은 부동소수점 형태로 그대로 유지되며, 추론 단계에서 연산 전후로만 동적으로 양자화
- 즉, 모델의 메모리 사용량을 줄이지만 연산 속도 향상에는 제한

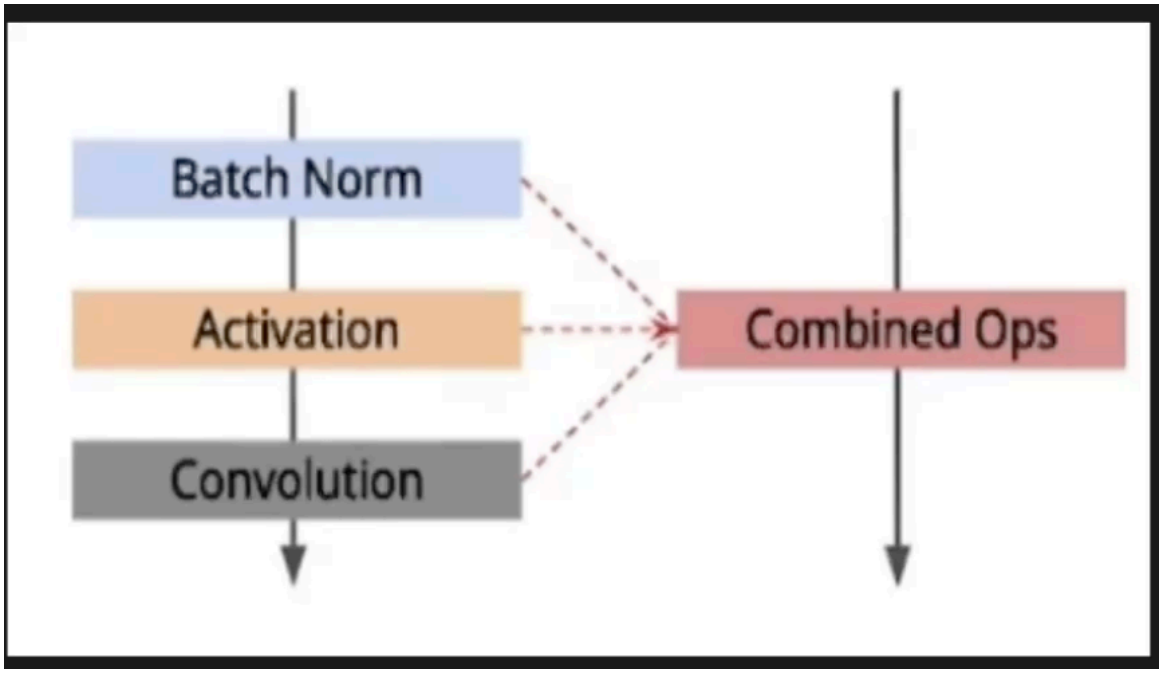
- 활성화 값은 추론 시, 메모리에 floating-point 형태로 read/write가 이루어지며, quantized kernels가 필요한 연산에서만, 처리 직전에 활성화를 8비트로 양자화한 후 연산을 진행하고, 연산이 끝난 후 dequantization을 통해 다시 부동소수점 형태로 변환

- 정확도를 유지하면서도 메모리 효율성을 일부 제공하지만,
  활성화가 항상 float 형태로 저장되므로 양자화된 가중치와 함께 사용하는 이점만 가짐

- 모델 로딩 시간 개선
  - 모델의 가중치를 메모리에 로드하는 과정에서 용량이 줄어들어 로딩 속도가 빨라짐
- 연산속도 향상 효과 제한
  - 연산 속도는 GPU 연산에 비해 CPU 연산에서는 효과적일 수 있으나,
    실제 연산의 kernel이 float으로 변환 후 계산되기 때문에 속도 향상이 미미

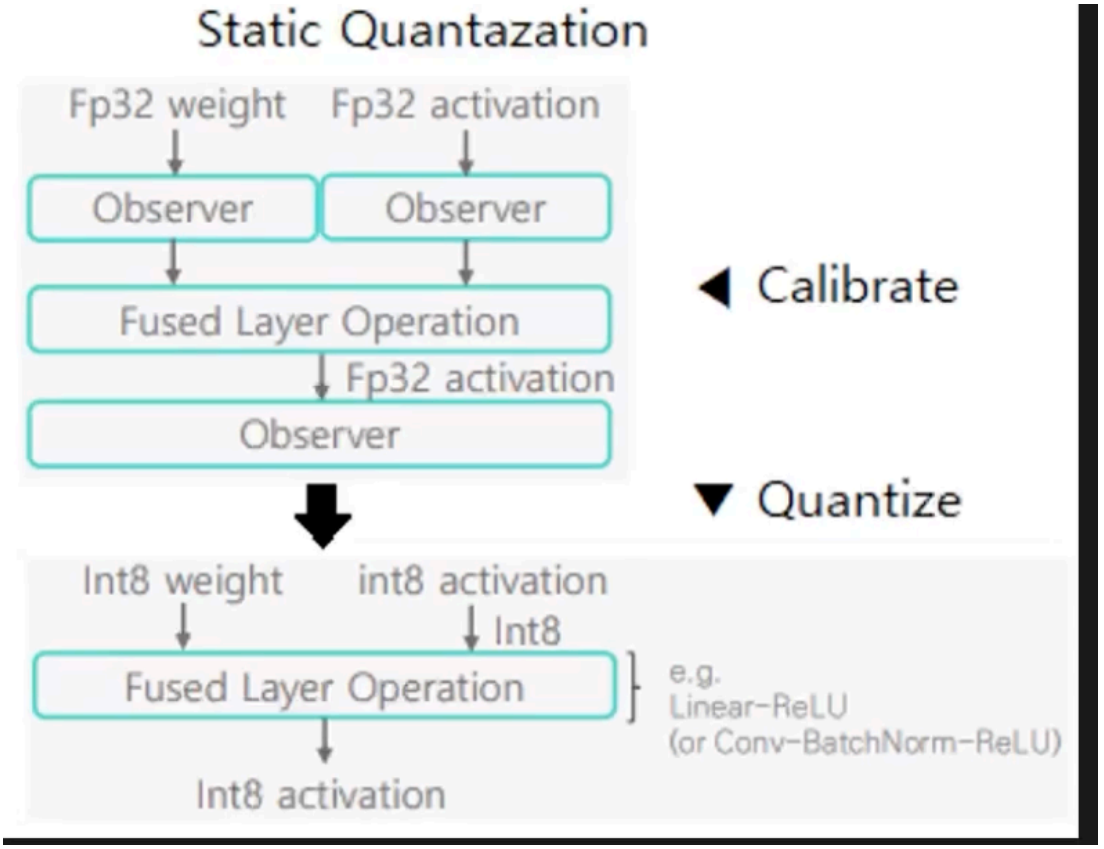
Static Quantization

- NPU 디바이스에서 사용 되는 경우(온디바이스)



- 모델의 가중치와 활성화 값이 모두 사전에 양자화
- 이 과정에서 가중치는 학습이 끝난 후에 양자화되고,
  활성화 값은 calibration을 통해 설정
- 이러한 사전 양자화를 통해 모델이 실행되기 전에 이미 양자화된 값들을 사용하게 되므로 추론시 빠른 연산이 가능
  - 병렬 처리 가능

- 가중치와 활성화의 Fusion
  - Static Qunatization에서는 활성화 양자화를 위해 preceding layer와의 fusion이 이루어짐
  - Fusion은 서로 다른 기능을 가진 레이어들(예: 활성호, 모듈)을 하나로 합치는 과정
  - 레이어를 합치면 레이어간 데이터 이동에 따른 context switching 오버헤드를 감소시켜 효율성 증대시키는 방식
- Fusion의 장점
  - Fusion을 통해 순차적으로 처리되던 연산을 병렬로 처리할 수 있어 모델의 처리 속도가 더욱 빨라짐
  - 레이어 간의 중간 데이터 이동이 줄어들면서 메모리 사용량을 최적화
  - Fusion은 CPU와 같은 환경에서 특히 효과적이며 GPU 내에서도 연산 효율성을 높이는 데 유리
- Statis Quantization의 장점



  - 가중치와 활성화 값을 통해 사전에 양자화하므로, Dynamic Quantization보다 연산 효율이 높음

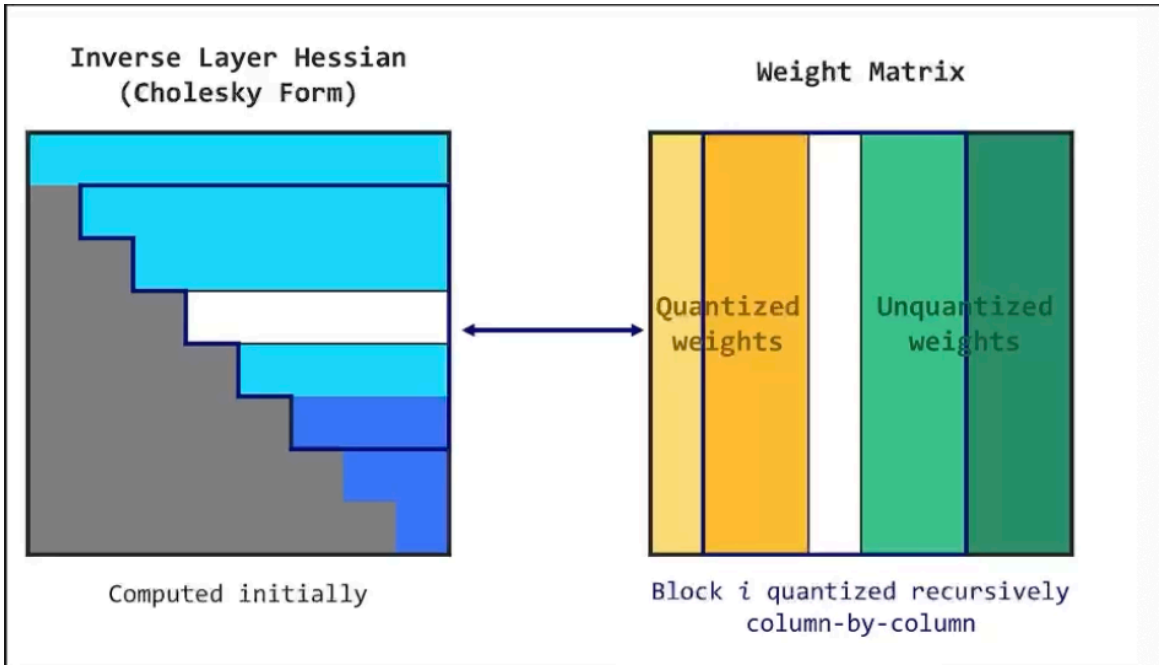
    - 추론 시, 양자화된 데이터만을 사용하므로 메모리와 계산 자원을 절약
    - 주요 이점
        - 연산 속도의 개선
        - 메모리 효율성 향상
        - context switching 오버헤드 감소

PTQ: GGUF

- GGML의 새로운 버전
    - GGML은 LLM 라이브러리의 C++ 복제본으로 LLaMA 시리즈 및 Falcon 등 여러 LLM을 지원
- 이 라이브러리에서 지원하는 모델은 Apple Silicon에서도 사용가능
- CPU가 충분하지 않을 경우 몇 개의 레이어를 GPU로 오프로드할 수 있으며,
  다양한 모델을 CPU기반으로 출력할 수 있음
- 2비트 ~ 8비트 정밀도까지 다양한 수준의 양자화를 제공
- 원본 모델을 가져와 GGUF 형식으로 변환한 후,
  마지막으로 GGUF 형식을 낮은 정밀도로 양자화하는 방식으로 적용됨

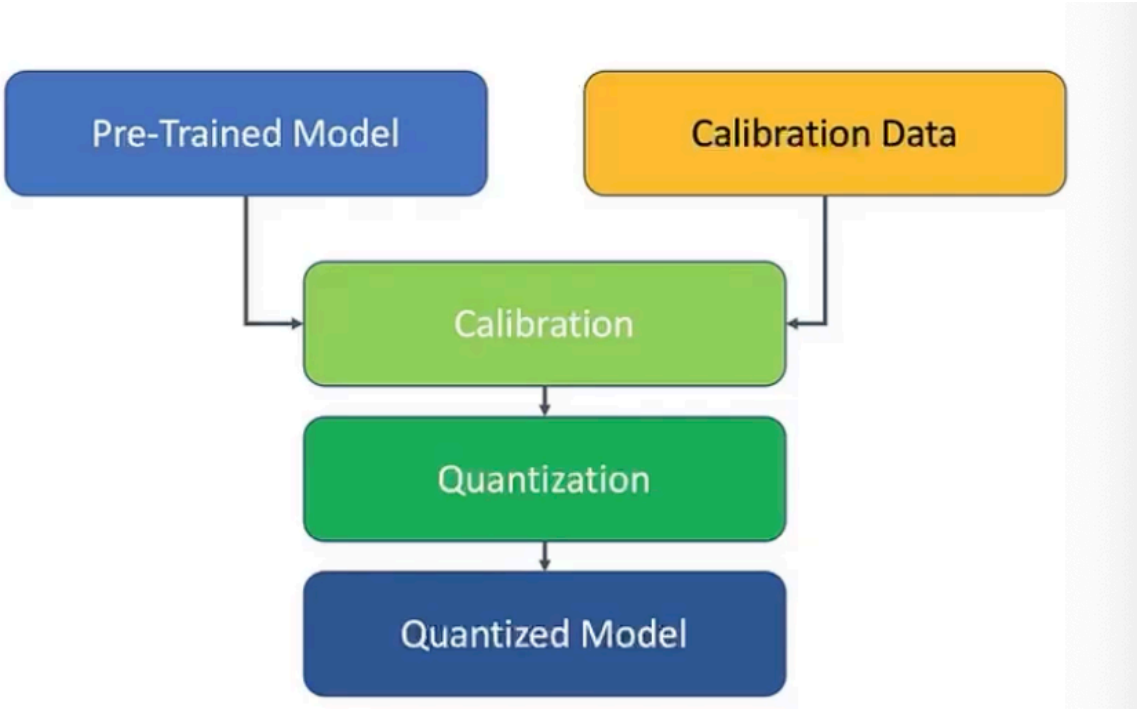
PTQ: GPT-Q

- 대표적인 표준화 방법



- 다양한 양자화 유형을 사용하여 모델의 각 요소에 최적화된 양자화를 수행하고, 성능 저하를 최소화
  - Static Range Quantization
    - 모델의 가중치를 고정된 범위로 양자화하는 방식
  - Dynamic Range Quantization
    - 모델의 가중치와 활성화 값을 동적으로 양자화하는 방식
  - Weight Quantization
    - 모델의 가중치만 양자화하는 방식으로, 메모리 절약과 연산 효율성
- 위와 같은 다양한 양자화 기법을 모델의 특성과 환경에 맞춰 적용할 수 있는 융통성을 제공하는 방법론
  - Calibrating도 가능

PTQ: AWQ



- Activation-Aware Quantization for LLMs,
  CPU와 GPU 모두 활용이 가능한 양자화 기법
- 모델의 모든 가중치를 양자화하지 않고, 모델의 성능을 유지하기 위해
  중요하지 않은 가중치만 선별하여 양자화하는 기법
- 기존의 양자화 방법론들이 가중치 값만을 이용하여 양자화하고,
  이 오차를 최소화하기 위해 다양한 방법들을 적용하는 것들과 다르게
  오차가 어디서 발생하는지 찾아나가는 것이 가장 큰 다른점
- 다양한 파이프라인과 맞물려서 동작할 수 있으며,
  Gradient를 활용하지 않고, Activation 값들을 활용하여 양자화 과정에서 발생하는 오차를 줄이는 방식으로, 다른 양자화 기법들과 함께 적용할 수 있다는 특징
- 적은 양의 calibrate 데이터를 활용하여 오차를 최소화 한다는 점에서 매우 효율적