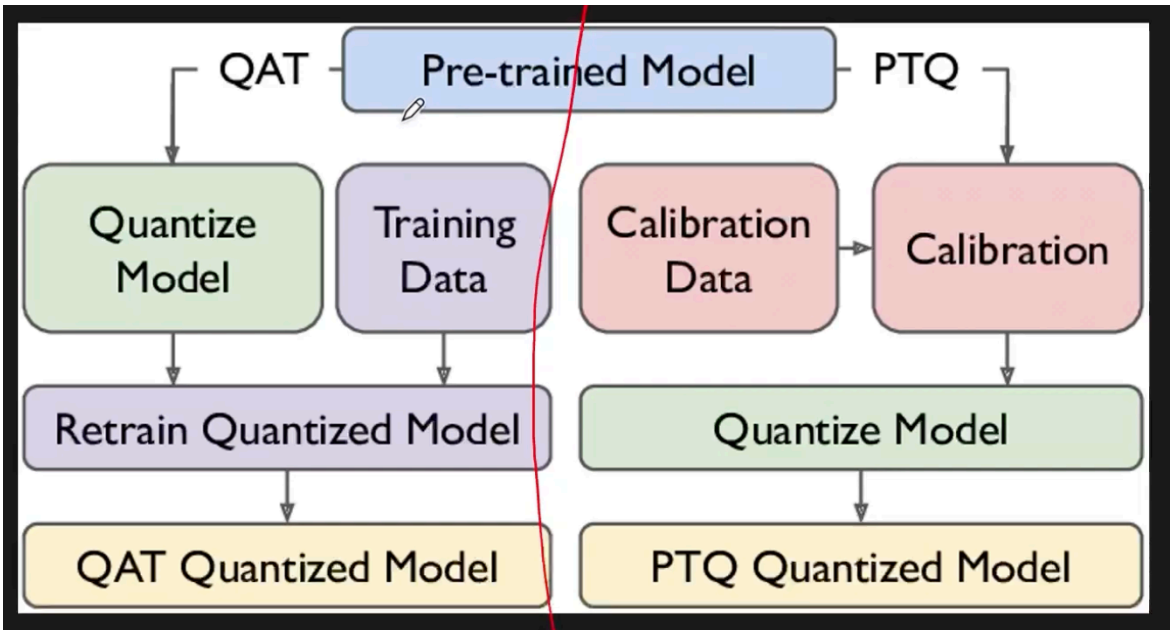
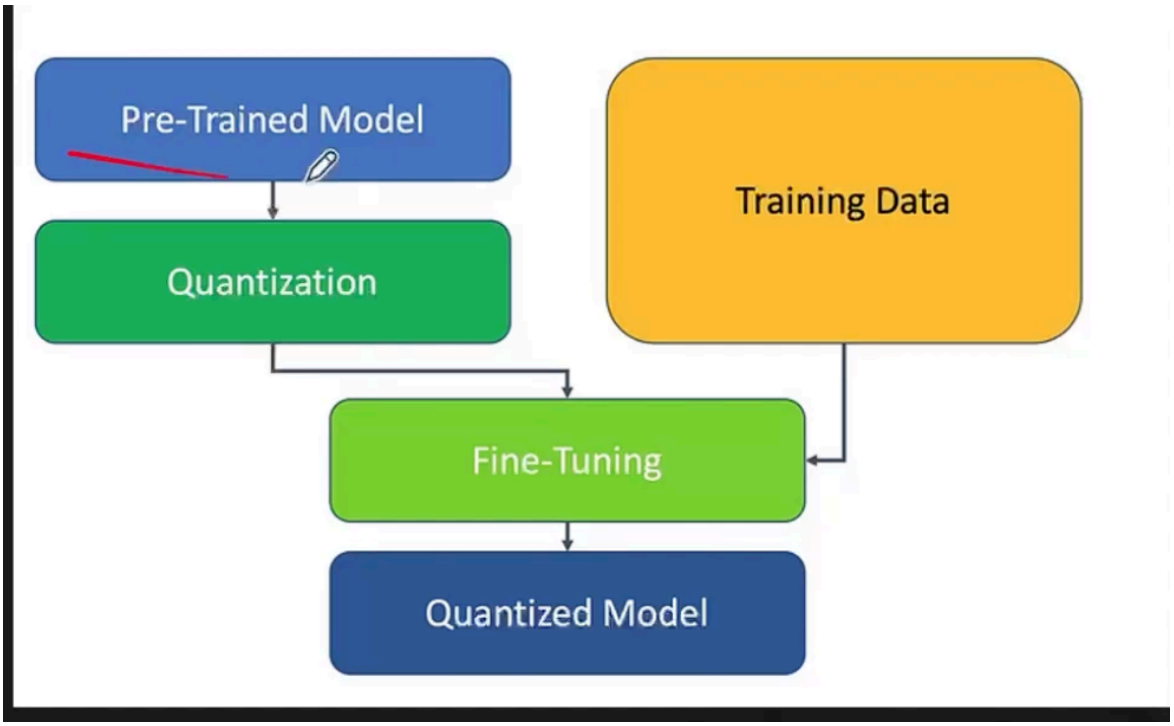
양자화의 접근방법

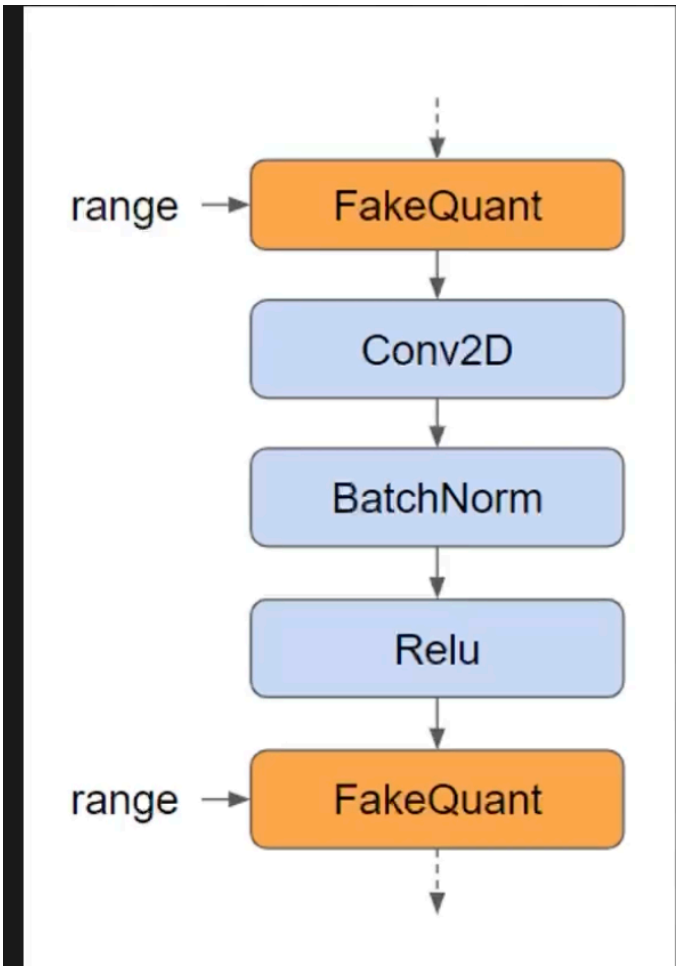


QAT (Quantization Aware Training)



- Pre-trained 모델로 시작하며, 모델의 가중치와 activation을 학습시키면서 양자화하는 것을 의미함.
    - 양자화를 반영하면서 훈련시킴
- 만약 양자화된 모델로 시작하는 경우, 양자화에서 발생한 정밀도 손실을 복구하고, 양자화 환경에서 더 fit하게 모델을 만들어줌
- 즉, QAT는 양자화 과정에서의 오류를 학습할 수 있도록 (오류를 미리 학습하여 반영) 양자화된 환경에서의 forward pass에서 발생하는 오류를 backward pass에서 최적화하여 보정

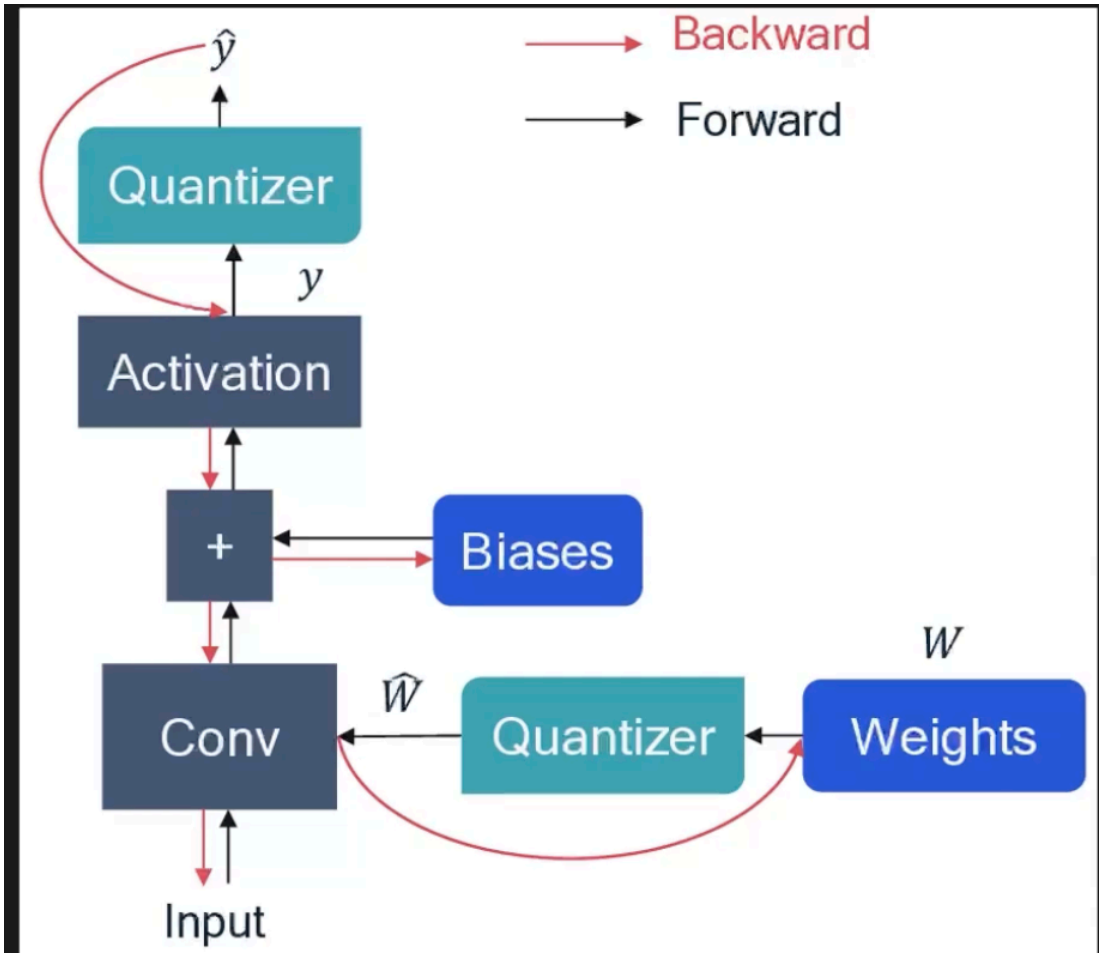
QAT의 동작과정: Forward Pass



- QAT는 양자화의 효과를 학습 단계에서 반영,
  이를 통해 모델이 양자화된 상태에서도 높은 성능을 발휘하도록
- Forward Pass

- FakeQuant Node가 삽입되며, FakeQuant Node는 실제 양자화처럼 동작하지만, 실질적으로 정수형 양자화 대신 실수형 값으로 계산하는 역할을 함
  - 모든 레이어의 값들은 아직 양자화가 되지 않은 상태
  - FakeQuant 노드를 통해 레이어들의 값들의 범위를 제한
- 이 과정에서 입력과 가중치를 양자화하고, 필요한 경우 출력을 다시 높은 정밀도로 변환
  - 실수값으로는 남겨두지만, 범위 제한으로 양자화의 학습 효과를 지님
- Fake Quant는 실제 양자화가 아니지만, 양자화로 인한 오차를 forward pass에서 모델이 경험하도록 함

Backward Pass

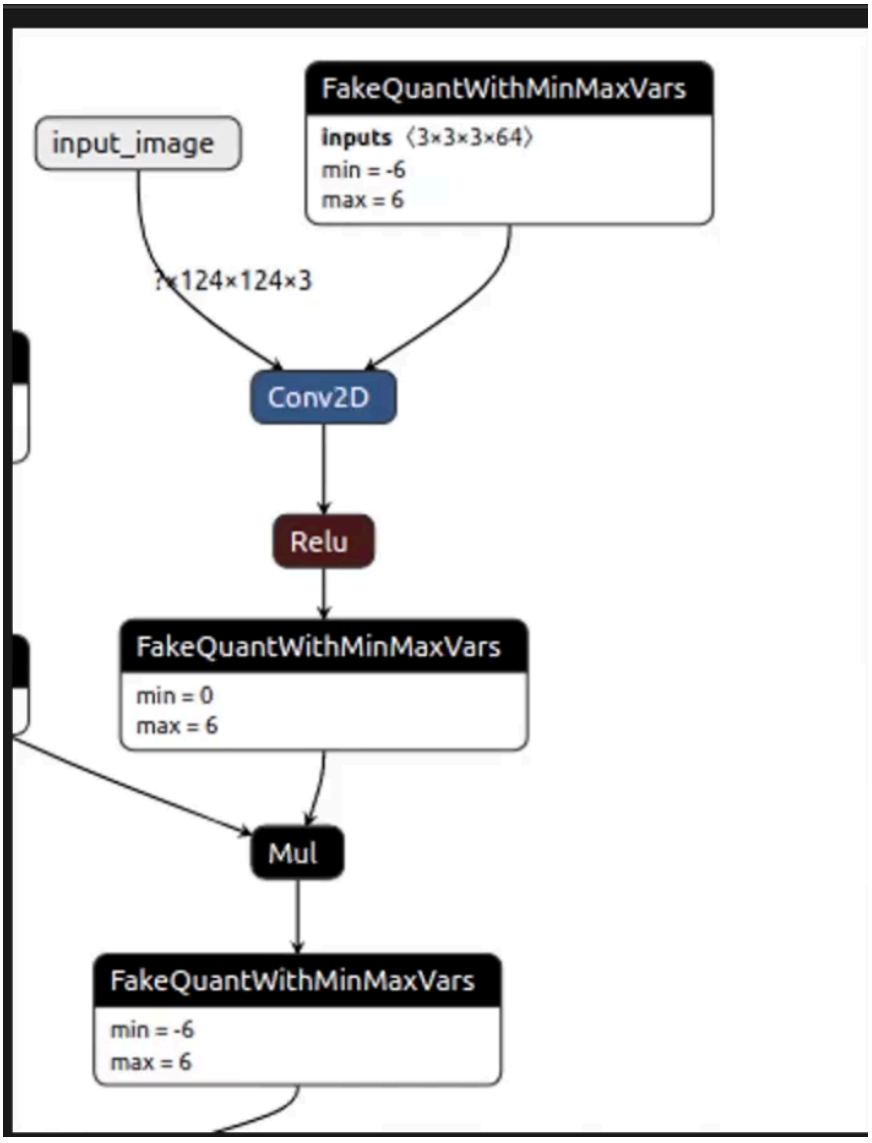


- 실제 양자회된 값이 아닌 원래의 실수형 값을 사용하여 그래디언트를 계산

- 양자화 과정에서 발생하는 오차는 그래디언트 계산에는 영향을 주지 않으며, 대신 이 오차를 최적화기를 통해 보정
- 결과적으로 모델은 양자화 과정에서 발생하는 오차를 줄이는 방향으로 학습

QAT: FakeQuant Node Insertion

- 동작방식

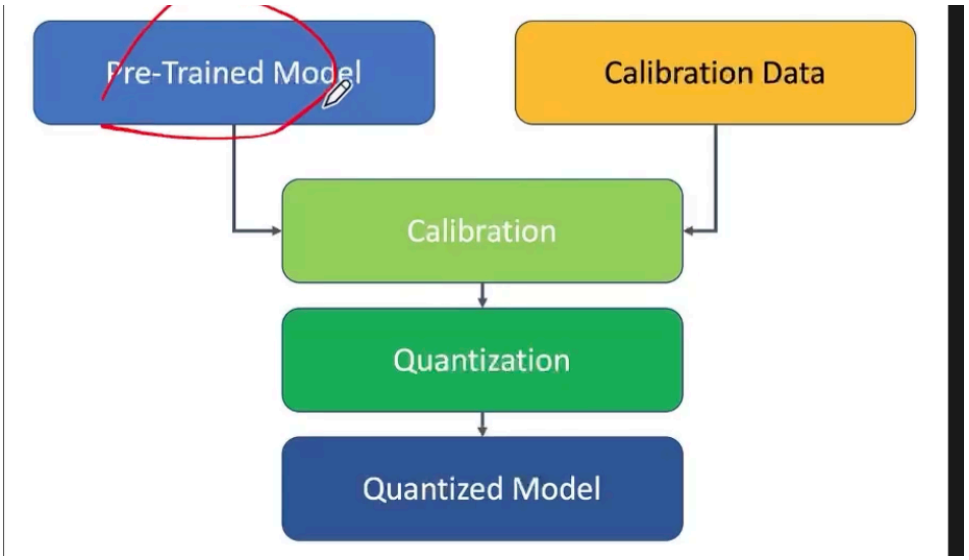


  - 입력 데이터(Input): 입력 데이터를 양자화된 값처럼 제한(clamp)
  - 가중치(Weights): 모델의 가중치도 양자화된 값처럼 제한
  - 출력(Output): 출력 값도 양자화된 값처럼 흉내 내도록 제한

- 이를 통해, FakeQuant Node Insertion은 입력 데이터, 가중치, 출력 데이터에 대해 양자화와 역양자화를 진행

- 너무 복잡하기에 산업에서 잘 활용하지 않음
    - 학습을 하되 보정을 잘하자는 방향으로 → 학습이란 과정을 좋아하지 않음

PTQ (Post Training Quantization)



- QAT가 훈련이라는 파이프라인이 필요하였던 것과는 반대로,
  훈련이 아닌 기존의 사전학습된 모델을 기반으로 양자화를 진행하는 것을 의미함
- QAT와 다르게 Training Data는 필요없으나,
  양자화 과정에서 calibration이 필요하며, calibration data가 필요함
  (보통 1,000 ~ 3,000개의 데이터만 필요함)
- 훈련비용이 들지 않는다는 강점이 있지만
  양자화 후 정교성이 조금 떨어지는 단점이 있음

PTQ vs QAT

비교 항목	PTQ (Post-Training Quantization)	QAT (Quantization Aware Training)
적용 시점	모델이 **훈련된 후** 양자화 수행	모델이 **훈련 중**에 양자화 적용
훈련 데이터 사용	일부 **검증 데이터**만 사용	**전체 훈련 데이터** 사용
학습 방식	추가적인 학습 단계 없이 양자화	**양자화 오차를 학습**하면서 보정
정밀도 손실	상대적으로 **정밀도 손실이 큼**	**정밀도 손실이 적음** (오차 보정)
복잡도	구현이 상대적으로 **간단**	**훈련 시간이 증가**하고 복잡함
실제 양자화 여부	모델의 가중치와 활성화 값을 정수로 변환	**FakeQuant 노드**로 양자화 흉내
적용 대상 모델	주로 **간단한 모델** 또는 사전 훈련된 모델	**모든 모델**에 적용 가능 (복잡한 모델 포함)
하드웨어 친화성	제한적인 최적화 (특정 하드웨어에서만 효과적)	**양자화된 하드웨어**에서도 높은 성능 유지
장점	- 간단하고 빠르게 적용 가능 - 추가적인 학습 필요 없음	- 양자화로 인한 정확도 손실 최소화 - 양자화된 환경에서 높은 성능 유지
단점	- 정밀도 손실이 큼 - 일부 레이어에서는 양자화가 어려움	- 학습 시간이 길어짐 - 복잡도가 증가

- 방법에 따른 성능 비교

※ unit : accuracy

Model	Original	Quantization Aware Training	Post Training Quantization
MobileNet v2 1.4	0.71	0.71	0.66
Inception v3	0.78	0.78	0.772