**Quantization**

실수형 변수(floating-point type)를 정수형 변수(integer or fixed point)로 변환하는 과정

이 작업은 weight나 activation function의 값이 **어느 정도의 범위 안에 있다는 것을 가정**하여 이루어진다.

즉, 양자화는 모델 파라미터의 32비트 부동수를 8비트 정수로 변환하는 기술로

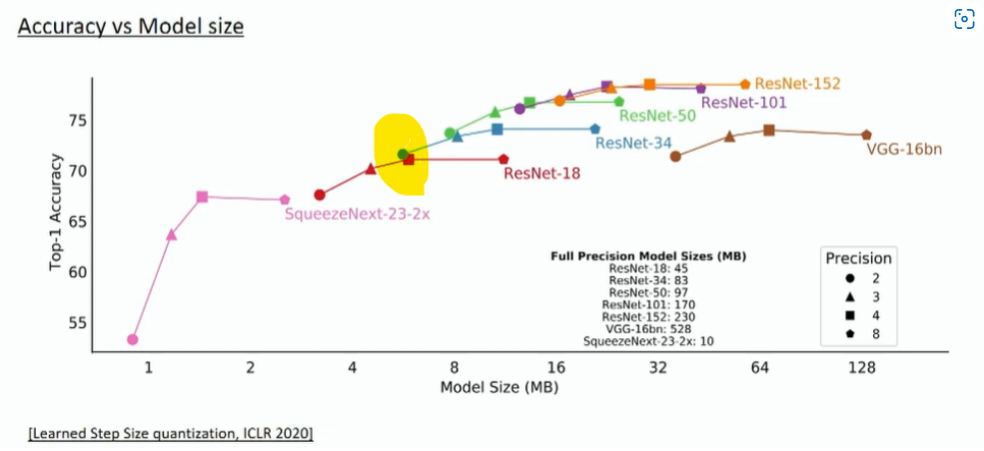
양자화를 통해 모델 크기와 메모리 공간을 원래 크기의 1/4로 줄일 수 있으며,

정확도는 거의 동일하게 유지하면서 추론을 약 2-4배 더 빠르게 할 수 있다.

+) **추론**(Inference)은 학습을 통해 만들어진 모델을 실제로 새로운 입력 데이터에 적용하여 결과를 내놓는 단계.

**따라서, Quantization의 주요 목적은 다음과 같다.**

1. 모델의 사이즈 축소  
2. 모델의 연산량 감소  
3. 효율적인 하드웨어 사용 (배포 중에 이점이 있음)



**Precision** = Quantization할 때 사용한 Bit

ResNet-34를 2 bit로 표현하였을 때의 Top-1 Accuracy가 ResNet-18을 4-bit로 표현하였을 때 보다 성능이 더 좋음.

이 때 모델 사이즈는 오히려 ResNet-34가 조금 더 가벼운 것도 확인가능!

이를 통해서, **양자화의 필요성**을 살펴볼 수 있음!

기억할 점은 FP32 → INT8로 변환 시 정보 손실이 발생하기 때문에

역으로 INT8 → FP32로 변환 시 잃어버린 정보로 인하여 그대로 변환되지 않는다.

이 때 발생하는 **Error를 Quantization Error**라고 하며 이 Error를 줄이는 것이 좋은 Quantization 알고리즘!

**양자화를 하는 방법**

1. 훈련 후 동적 양자화
2. 훈련 후 정적 양자화
3. 양자화 인식 훈련

그러나 사용하려는 모델에 이미 양자화된 버전이 있는 경우

위의 세 가지 워크플로우를 거치지 않고 직접 사용할 수 있음.

예를 들어, **토치비전 라이브러리**에는 이미

**모바일넷 v2, ResNet 18, ResNet 50, Inception v3, GoogleNet 등의 모델용 양자화된 버전이 포함**되어 있음.

**https://pytorch.org/tutorials/recipes/quantization.html**

**Quantization Aware Training**

훈련 후 양자화 method에 더 쉽게 접근할 수 있지만

**모델의 정확성**을 생각해보면, **양자화 인식 훈련**을 사용하는 것이 정확도에는 더 좋다.

***양자화 인식 훈련***

<https://www.tensorflow.org/model_optimization/guide/quantization/training>

***양자화 인식 훈련 종합 가이드***

https://www.tensorflow.org/model\_optimization/guide/quantization/training\_comprehensive\_guide

***Keras 예제의 양자화 인식 훈련***

<https://www.tensorflow.org/model_optimization/guide/quantization/training_example>

양자화 인식 훈련은 주로 CNN에 이용

모델을 양자화 인식 트레이닝에 사용할 수 있도록 하려면

모델 정의의 \_\_init\_ 메서드에서 QuantStub 및 DeQuantStub을 정의하여

텐서를 부동 소수점에서 양자화된 유형으로 변환하고 그 반대의 경우도 마찬가지로 진행한다.