**Quantization Introduction**

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

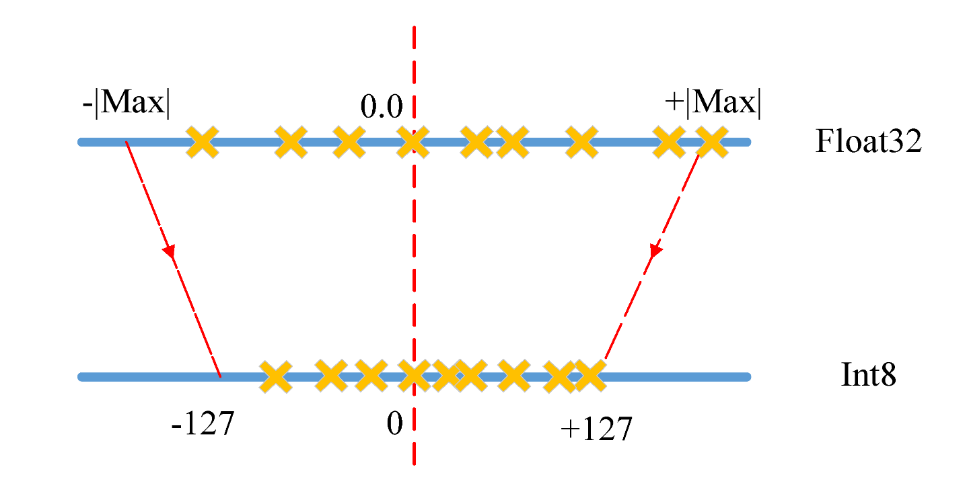
이 작업은 weight나 activation function의 값이 **어느 정도의 범위 안에 있다는 것을 가정**하여 이루어진다.

즉, 양자화는 **모델 파라미터의 32비트 부동수를 8비트 정수로 변환하는 기술**로

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

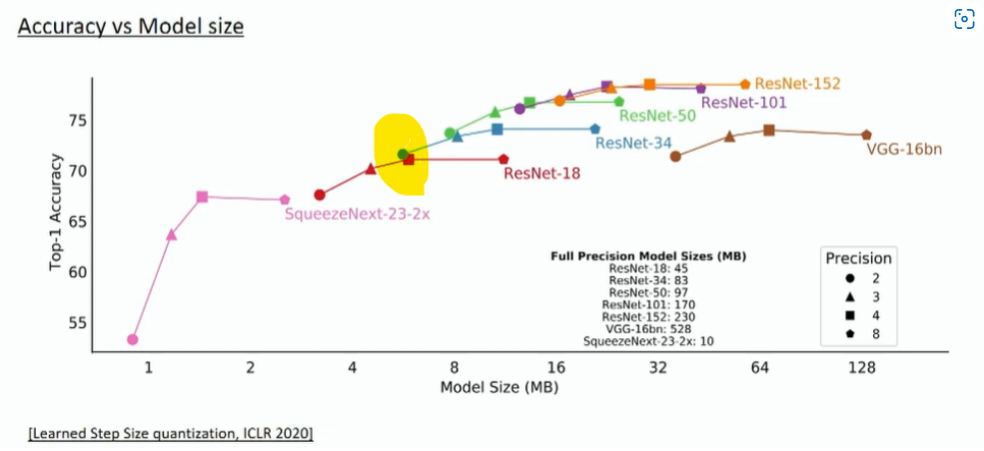
정확도는 거의 동일하게 유지하면서 추론을 약 2-4배 더 빠르게 할 수 있다.

+) **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 양자화** - 훈련 후 동적 양자화

- 훈련 후 정적 양자화

2. **양자화 인식 훈련**

**훈련 후 양자화**는 이미 Training된 모델에 Quantization을 적용하는 것으로,

Quantization이후에도 정확도의 하락 폭이 작지만, 소형 Model에서는 적합하지 않다.

**양자화 인식 훈련**은 학습을 통한 Quantization을 Simulate해보며 최적화하는 방식으로,

training과정 중에서 Quantize를 수행하는 것이며

Fake Quantization Node를 첨가하여 Quantize되었을 경우 어떻게 동작할지 Simulation한다.

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.

**Dynamic Quantization**

**동적 양자화**

**특징**

1. 가장 간단한 양자화 기법
2. 가중치(Weights)에 대해서만 양자화 진행하며, Training 이후에 진행
3. 활성화(Activations)는 추론(Inference)할 때 동적으로 양자화 진행
4. **보통 CPU 환경에서만 Inference 가능**

**With PyTorch 1.7.0, we could do dynamic quantization using x86-64 and aarch64 CPUs. However, NVIDIA GPUs have not been supported for PyTorch dynamic quantization yet.**

**Static Quantization**

**정적 양자화**

**특징**

1. 모델의 가중치(Weights)와 활성화(Activations) 모두 사전에 양자화를 진행
2. Activation, Convolution 등 각각의 기능을 수행하는 Layer들을 Fusion(합치기)하여

Layer간 데이터 이동으로 발생하는 Context Switching Overhead를 줄일 수 있다.

또한, Sequential하게 처리되던 연산을 병렬로 처리할 수 있다는 장점도 있다.

**Quantization Aware Training**

**양자화 인식 훈련**

난이도 자체는 **훈련 후 양자화** Method에 접근하는 것이 더 쉽지만,

모델의 성능과 정확성을 생각해보면, **양자화 인식 훈련**을 사용하는 것이 더 좋다.

양자화 인식 훈련은 주로 CNN에 이용

모델을 양자화 인식 트레이닝에 사용할 수 있도록 하려면

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

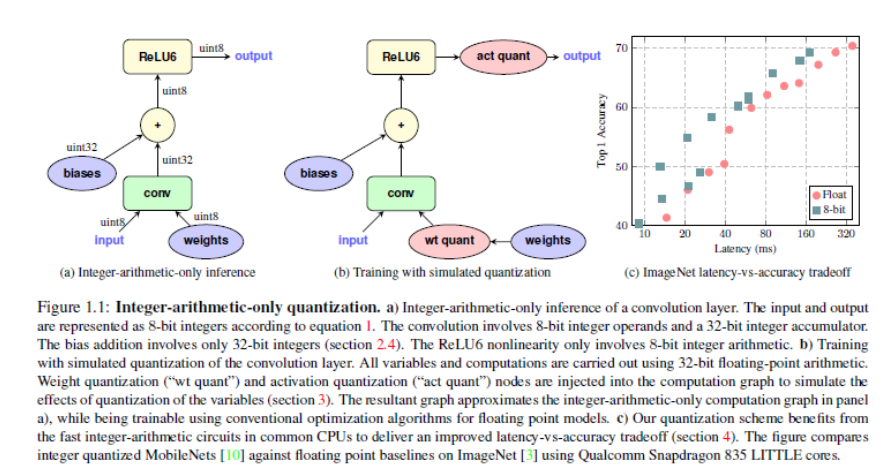
텐서를 부동 소수점에서 양자화된 유형으로 변환하고 그 반대의 경우도 마찬가지로 진행한다

본론으로 돌아와서,

양자화 인식 훈련(QAT)는 모델의 가중치(Weights)와 활성화(Activations)를 학습하면서 양자화를 하는 것이다.

**특징**

1. 학습은 CPU, GPU 환경에서 모두 가능하지만, 추론은 CPU에서만 가능하다.
2. Dynamic Quantization, Static Quantization보다 높은 Accuracy를 확보할 수 있다.
3. Quantization Aware Training중에 활성함수(Activation)의 실제 최대 및 최소 출력 범위 확인도 진행한다.



위 그림에서 볼 수 있듯이, **Quantization Aware Training**은 Training과정에서 Quantize를 수행하는 것으로,

**Fake Node**를 첨가하여, Quantize되었을 시 어떻게 수행이 될지 시뮬레이션을 돌려보는 방식이다.

여기서, 원래 Training Loss가 수행되는 것이 약간 틀어져서 Quantize된 것까지 고려해

양자화 모델의 정확도 하락을 최소화하는 방식이다.

 QAT를 사용하면 모든 가중치와 활성화는 훈련의 전진 및 후진 통과 동안 "가짜 양자화"된다. 즉, 부동소수점 값은 int8 값을 모방하도록 반올림되지만, 모든 계산은 여전히 부동소수점 숫자로 수행된다. 따라서 훈련 중 모든 가중치 조정은 모델이 궁극적으로 양자화될 것이라는 사실을 "인식"하면서 이루어진다. 따라서 양자화 후 이 방법은 일반적으로 다른 두 가지 방법보다 높은 정확도를 산출한다.

테이블이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

**PyTorch Version 1.3 부터 양자화가 가능!**

**Quantization Example Codes**

텍스트이(가) 표시된 사진

자동 생성된 설명

**텍스트이(가) 표시된 사진

자동 생성된 설명**

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

**Quantization Tools in PyTorch**

**Quantization References**

**[PyTorch Documentation & Guide]**

1. ***Quantization Recipe*** **– Click** [**Here**](https://pytorch.org/tutorials/recipes/quantization.html)
2. ***Quantization Documentation*** **– Click** [**Here**](https://pytorch.org/docs/stable/quantization.html)

**[TensorFlow Documentation & Guide]**

1. ***양자화 인식 훈련* – Click** [**Here**](https://www.tensorflow.org/model_optimization/guide/quantization/training)
2. ***양자화 인식 훈련 종합 가이드* – Click** [**Here**](https://www.tensorflow.org/model_optimization/guide/quantization/training_comprehensive_guide)
3. ***Keras 예제의 양자화 인식 훈련* – Click** [**Here**](https://www.tensorflow.org/model_optimization/guide/quantization/training_example)

**[참고자료]**

1. **Deep Learning Quantization Summary – Click** [**Here**](https://velog.io/@jooh95/%EB%94%A5%EB%9F%AC%EB%8B%9D-Quantization%EC%96%91%EC%9E%90%ED%99%94-%EC%A0%95%EB%A6%AC)
2. **Deep Learning Theory Quantization – Click** [**Here**](https://velog.io/@ganta/Quantization)
3. **Introduction to Quantization on PyTorch – Click** [**Here**](https://pytorch.org/blog/introduction-to-quantization-on-pytorch/)
4. **모델 압축 기법 – Click** [**Here**](http://dankernel.sciomagelab.com/2021/05/12/distiller-%EB%AA%A8%EB%8D%B8-%EC%95%95%EC%B6%95-%EA%B8%B0%EB%B2%95-3-quantization-%EC%96%91%EC%9E%90%ED%99%94/)