**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

**fbgemm:** More about [here](https://blog.ggaman.com/1028)

* Quantization을 지원하는 두개의 backend가 있는데, x86용으로는 FBGEMM, arm용으로는 QNPACK이 있다고 한다.
* FBGEMM은 페이스북에서 만든 행렬처리용 라이브러리다. 그리고 Caffe2 와 PyTorch의 양자화 연산에 사용된다고 적혀 있다.

**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/)

**(+)**

**To Apply Static Quantization**

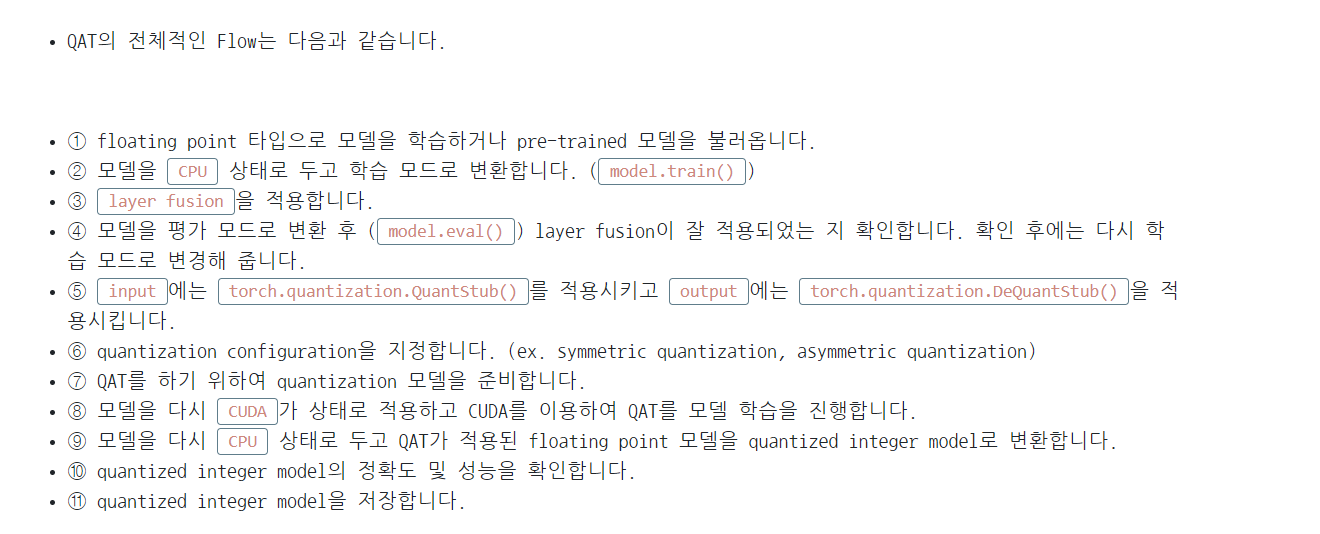
텍스트이(가) 표시된 사진

자동 생성된 설명

|  |  |  |  |
| --- | --- | --- | --- |
| **Quantization Modes** | **Features** | **Inference Latency** | **Inference Accuracy Loss** |
| **Dynamic Quantization** | Dynamic, Real Time Calculating Data Range | Faster | **Smallest** |
| **Static Quantization(=PTQ)** | Calibration & Limit Data Range | **Fastest** | Smaller |
| **Quantization Aware Training** | - | **Fastest** | **Smallest** |

|  |  |  |
| --- | --- | --- |
|  | **PTQ(Post Training Quantization)** | **QAT(Quantization Aware Training)** |
| **Definition** | Floating Point 모델로 학습을 한 뒤 결과 Weight값들에 대하여 Quantization하는 방식으로 학습을 완전히 끝내 놓고 Quantization error를 최소화하는 방식 | 학습 진행 시점에 Inference 시 Quantization 적용에 의한 영향을 미리 시뮬레이션을 하는 방식으로 최적의 Weight를 구하는 것과 동시에 Quantization을 하는 방식 |
| **Advantage** | 파라미터 Size 큰 대형 모델에 대해서는 정확도 하락의 폭이 작음 | Quantization 이후 모델의 **정확도 감소 폭을 최소화**할 수 있음 |
| **Disadvantage** | 파라미터 Size가 작은 소형 모델에 대해서는 정확도 하락의 폭이 큼 | 모델 학습 이후 추가 학습이 필요 |

**Tutorial**

****

**To Presentation**

**Quantization 동기**

while models are becoming more efficient, high accuracy still implies high complexity!

발달되고 있는 Model들을 Embedded Board나 Mobile 환경에서도 손쉽게 사용하고 싶은데,

이를 위해서는 모델의 용량뿐만 아니라 전력 사용량도 줄이는 등 최적화 작업을 수행해야 합니다.

즉, 더 나은 성능을 얻되 에너지 사용은 줄이는 방안을 연구한 것이 양자화입니다.

**Quantization의 핵심 개념**

32bit의 float type 이지만,

이를 8 bit의 int type으로 바꿔서

메모리를 1/4로 줄이고, 연산 속도는 2배~4배 빠르게 구현하는 것.

**Quantization 종류**

훈련 후 양자화는

먼저 훈련을 다 끝내놓고 Quantization Error를 Minimum하게 하는 방향으로 Quantization을 진행

좀더 편리하고 쉽지만, Weight가 완벽히 최적화되기는 힘든 구조.

이보다 좀 더 정확한게 Quantization Aware Training인데,

Training하면서 즉, 최적의 Weight를 결정하면서 동시에 Quantization을 진행하는 것.

좀 복잡하지만 Weight를 보다 최적화시키는 구조.

**테이블이(가) 표시된 사진

자동 생성된 설명**

**PTQ - Dynamic Quantization**

텍스트이(가) 표시된 사진

자동 생성된 설명

동적 양자화에 대한 것인데, LSTM이나 Multi Layer Perceptron에 적합하며

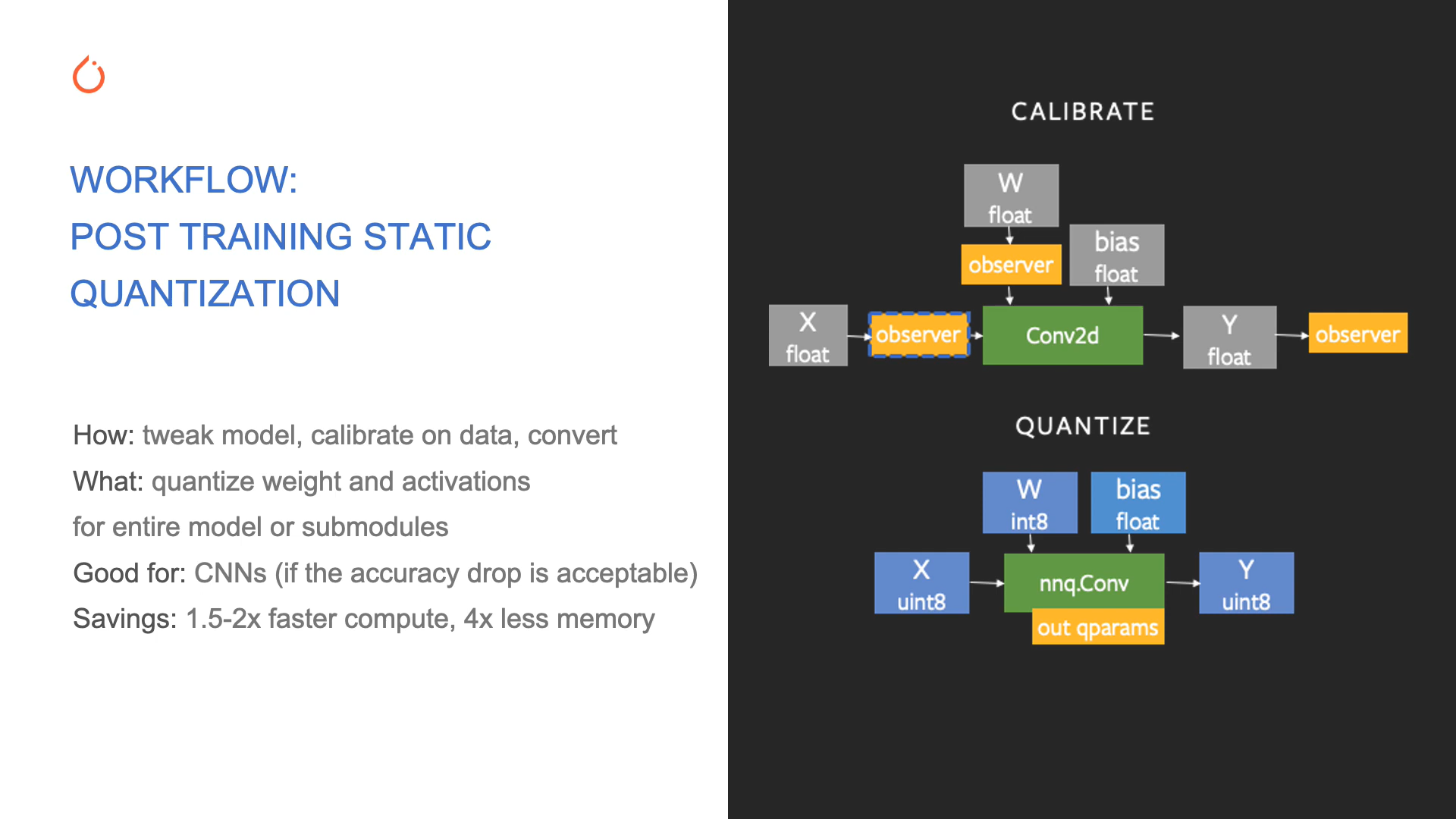
우측에 있는 간단한 코드로 실행이 가능합니다.

우선 모델을 로드하는 것으로 시작하며,

Quantization module을 지정하고 저장하면 됩니다.

매우 간단한 프로세스입니다.

**PTQ - Static Quantization**



CNN에 적합한 것이 특징

약간 조정한 일부 데이터를 기반으로 하여 모델을 보정한 다음

보정된 모델을 양자화된 모델로 바꾸는 과정입니다.

가중치와 활성화를 모두 양자화하는 것입니다.

다음 페이지에서는 Step by Step Approach를 제시하겠습니다.

**PTQ = Post Training Quantization**

**PTQ – Static Quantization Step by Step Approach**

텍스트이(가) 표시된 사진

자동 생성된 설명

Static Quantization을 적용하는 과정입니다.

우선, 모델을 로드하고 레어들을 fuse해줍니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

앞에서 모델을 약간 수정했다면, 여기서는 양자화 구성을 설정합니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

모델 수정과 양자화에 대한 설정을 마쳤다면

Convert 명령어를 통해 실제적으로 양자화를 진행해주도록 합니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Quantized Model Save!

**그러나 때때로,**

**CNN은 훈련 후 정적 양자화 (Static Quantization After Training)에서 좋은 성능을 보이지 않을 수 있다.**

**이때는, QAT을 적용하면 된다.**

**입력 및 가중치가 양자화 되었을 때,**

**올바른 답을 제공하는 방법을 학습하는 방식으로 네트워크를 조정하므로**

**기본적으로 Fine Tuning을 필요로 합니다.**

**QAT – Quantization Aware Training**

텍스트이(가) 표시된 사진

자동 생성된 설명Calibration 대신 Fine Tuning을 수행한다는 점을 제외하면 Static Quantization과 매우 유사하다.

우선 Model을 불러오고,

양자화와 학습을 동시에 진행해야 하기 때문에 불러온 모델을 train모드로 설정합니다.

그리고, 유사한 성격을 가지는 Layer끼리 Fusion하는 과정을 진행해줍니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Train Data를 사용하여 모델을 훈련하는데,

Fake Quantization Node로부터 수집한 데이터로 정보를 활용하여 양자화를 최종적으로 진행

이후의 진행은 Static Quantization과 동일!

테이블이(가) 표시된 사진

자동 생성된 설명

**Quantization 결과 분석 표는 이와 같이 만들기!**

Top5 Accuracy / Top1 Accuracy 둘 다 사용하면 좋다.

테이블이(가) 표시된 사진

자동 생성된 설명

**Quantization Aware training성능이 제일 좋구만!**

텍스트이(가) 표시된 사진

자동 생성된 설명

**(+) Graph Mode Quantization – prototype version of PyTorch Quantization**