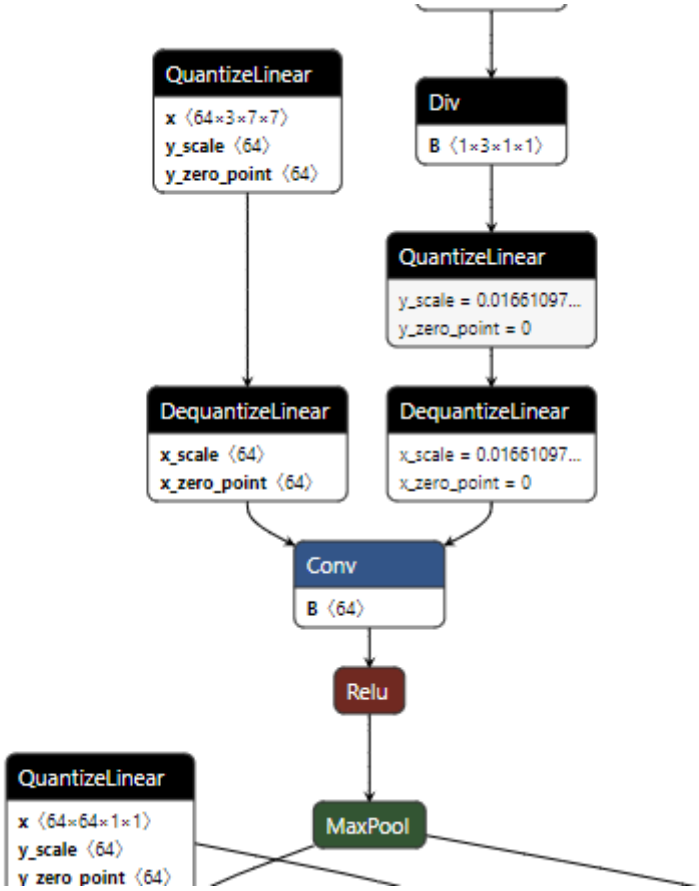
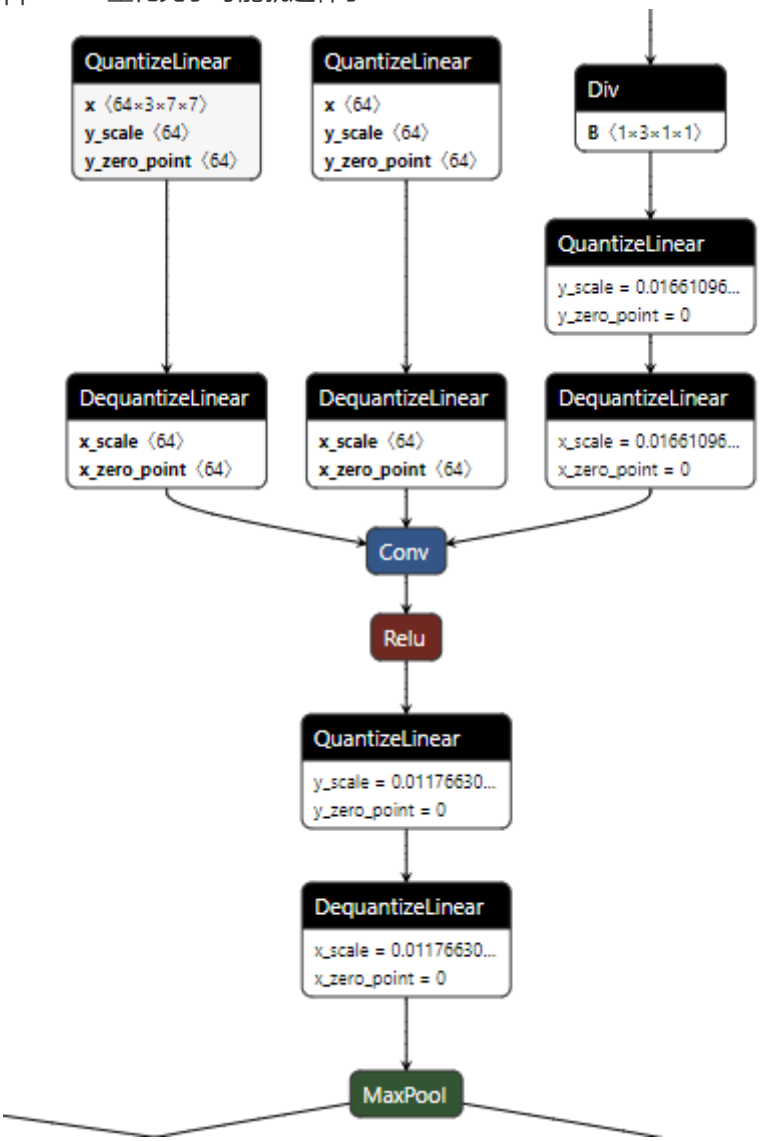
- 首先这个量化各家差异是显著的，量化差异的来源一般就是推理写的不一样，或者硬件不一样
- 比如同样一个网络，trt量化完了可能是这样



- ppl cuda量化完了可能就这样了



- 那这个就是硬件一样，推理写的不一样
- 然后对于这一个量化点而言各家的定义也不一样，主要是体现在是不是对称量化、量化粒度是tensorwise的还是channelwise的，还有rounding是咋弄的，还有scale要不要做power-of-2
- 我们先抛开写推理的这帮人瞎几把乱写的可能性，这个地方的差异其实就直接反映了硬件的特性。比如有些硬件它没有浮点运算能力，或者浮点贼几把慢，那就会选power-of-2的量化
- 比如你想要推理速度尽可能快，那一般就选tensorwise的量化，这个一般比channelwise的快个10%~30%
- 那个rounding也是直接跟硬件相关的，一般也就是用它硬件的round指令
- 抛开这些不算，还有一些更特殊的差异，在于计算的具体过程
- 比如就卷积而言你硬件上具体咋算的，16位累加的还是32位累加的，还是64位累加的，还有那些奇奇怪怪的sigmoid, softmax，
- 每家的算法也不一样，归根结底也是硬件的不同导致的，比如ncnn这个地方不是搞了个6bit的winograd，trt就是直接tensorcore上去8bit算，因为人家有这种运算器是不是，人家不用winograd。sigmoid这些东西，cpu, gpu上它大不了就回fp32去算，fpga上你就只能查表，这里面其实各种实现都会有，都是针对不同硬件的特性去做一些性能和精度的取舍。
- 所以我们解决的不是简单说这个玩意怎么从float->int8的问题，我们解决的主要是你这玩意怎么快，同时它精度别归零的问题。而且你知道推理框架这玩意很难改的，我们有些时候知道推理写的有问题，但是已经那样了，我们已经解决不了这里面的问题了。它不是那种error你知道吧，它就是性能达不到最优
- 就是同一个网络量化完，比如resnet这种，trt要70个量化信息，我们商汤可能要200个，我们速度是能赶上trt，但是精度上肯定不如trt那么夸张，trt它这个很夸张的地方就是它只要很少的量化信息，就能跑到很高的速度，它就只量化那些最重要的东西，同时保证中间精度转换的损耗不大，这玩意，实话实说，我们写的时候没有这样的经验。

- 就是量化这个事情你看着它很容易，其实它也不难， 但是就是需要从系统的角度去做，这个东西我们很少有机会有经验接触到。
- 张志大佬的意思就是， 从算法设计，到推理，到硬件设计。 （大佬吐槽：做硬件的人不听我的，还打不过做硬件的）