movtivation：

Setup

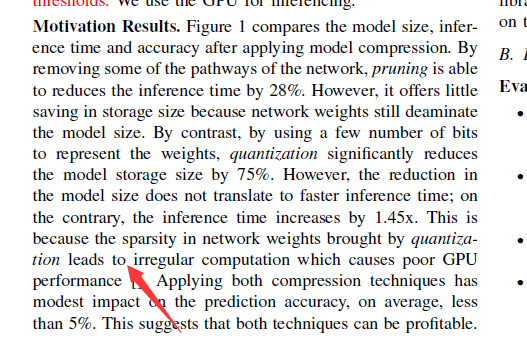
最后我们分别使用量化和剪裁两种方法对模型进行压缩。

量化模型时我们采用的是tensorflow开源的量化工具，量化步骤：首先在源码下编译tensorflow的量化工具；然后将ckpt模型进行成pb，从graph中读取模型的输入输出节点；最后对模型使用工具进行8bit量化。在使用量化工具时需要根据要求输入模型的具体参数，例如输入输出节点，图片的尺寸等参数。剪裁我们采用的主要方法是根据权值绝对值的大小评估神经元的重要性，然后进行剪裁，具体的步骤：首先计算每个神经元权值绝对值的大小；然后设定百分比移除权值的百分比，将权值绝对值小的神经元进行移除；最后重新训练剪裁后的模型恢复精度。在剪裁CNN模型时，需要对每一层模型进行上述的步骤，然后根据每一层的精度损失确定剪裁的阈值，最终完成整个模型的剪裁。

movtivation：Setup

We use pruning and quantization for model compression. when pruning we judge importance of neurons determinted by the absolute value of their weights and calculate the sum of absolute kernel weights firstly. Then we setup different thresholds according to different models remove the neurons and retrain the whole model to recover accuracy. When quantifying the model, we use the tensorflow open source quantitative tool. When using the quantitative tool We need to compile the source code firstly. Then we convert ckpt to pb and get the input and output nodes from the pb .Because tensorflow only support for eight-bit quantization, so we just quantify models eight-bit.

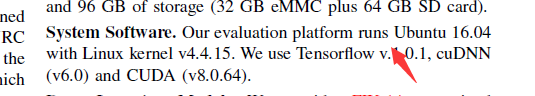
Result:



补充解释量化时间增大的原因：

根据实验结果可以看到：量化的平均精度损失在2%左右，这主要是由于神经网络的健壮性，同时可以减少模型的存储空间75%，但是推理时间增大。通过使用tensorboard查看量化前后的pb发现，为了保证输出层的输入数据的准确性，量化后的graph里加入了含有转换函数的子图，把8位转回32位作为输出层的输入，导致模型的推理时间增大。

使用tensorflow版本是1.3.0



Deep Learning Models

The modles we used including Mobilenet 、Inceptionnet 、Resnet、Vggnet and LSTM model for machine translation(NMT) .The dataset of NMT model is the WMT16 German and English dataset with news-test2013 as the dev set and news-test2015 as the test set.

不同的压缩技术：

因此模型压缩技术在近几年得到了发展，目前主要的压缩方法包括：参数修剪和共享、低秩分解和知识提纯。参数共享从数据存储方法入手，通过将原本的参数信息用更精简的方式存储达到压缩深度模型存储空间的目的，量化属于其中一种。参数修剪关注于探索模型参数中冗余的部分，并尝试去除冗余和不重要的参数，而如何找到一个有效的参数重要性的评价手段，在权值修剪方法中就尤为重要。目前这种评价标准也是研究的热点。低秩分解（Low-rank factorization）技术主要使用矩阵分解的方法以估计深层网络中最具信息量的参数，然后通过将一个大矩阵分解为多个小矩阵进行计算，来减少总体计算量和模型大小。这种方法用于全连接层效果较好，但低秩分解的压缩算法用于卷积层时，会出现误差累积的效果，对最后精度损失影像较大，需要对网络进行逐层的微调，费时费力。知识提纯（Knowledge Distillation）则是通过学习一个精炼模型来起到模型压缩的目的，即训练一个更加紧凑的神经网络以再现大型网络的输出结果。这种方法从本质上改变了原有的神经网络，产生出一个新的网络。

Different compression techniques

Therefore, model compression technology has been developed in recent years. The main compression methods include parameters sharing, weights pruning, Low-rank factorization and Knowledge Distillation. Parameters sharing aims to change the data storage way, it converts the original parameter information to with a more streamlined manner to compress model storage space. Weights pruning aims to remove some parameters in the network which is redundant to reduce the computation costs without hurting accuracy, and how to find an effective standard to evaluate the redundant weights is a research hotspot. Low-rank factorization use matrix decomposition method to estimate the most informative parameters in the deep network, and then reduce the total computation and model size by decomposing a large matrix into multiple small matrices. But the error accumulation effect will occur when using low rank decomposition in the convolution layer, the layer needs to be fine-tuned layer by layer and need a long time to finish. Knowledge Distillation aims to model compression by learning a small and compact model, the new model can also express the original model output information.