文章编号：1003-0077(2019)01-0093-10

面向神经机器翻译的模型存储压缩方法分析

林 野，姜雨帆，肖 桐，李恒雨

（东北大学 自然语言处理实验室，辽宁 沈阳 110819）

摘 要：模型存储压缩，旨在在不改变模型性能的同时，大幅度降低神经网络中过多的模型参数带来的存储空间浪费。研究人员对于模型存储压缩方法的研究大多数在计算机视觉任务上，缺乏对机器翻译模型压缩方法的研究。该文在机器翻译任务上通过实验对比剪枝、量化、低精度三种模型压缩方法在 Transformer 和 RNN(recurrent neural network)两种模型上的模型压缩效果，最终使用剪枝、量化、低精度三种方法的组合方法可在不损失原有模型性能的前提下在 Transformer 和 RNN 模型上分别达到 5.8×和 11.7×的压缩率。同时，该文还针对三种模型压缩方法在不同模型上的优缺点进行了分析。

关键词：模型压缩；剪枝；量化；低精度；机器翻译

中图分类号：TP391 **文献标识码**：A

On Storage Compression for Neural Machine Translation

LIN Ye, JIANG Yufan, XIAO Tong, LI Hengyu

(NLP Laboratory, Northeastern University, Shenyang, Liaoning 110819, China)

Abstract: The model storage compression is to significantly reduce the storage cost by removing redundant model parameters without quality loss. Previous efforts are mostly devoted to computer vision tasks, leaving neural machine translation less touched. In this paper, we compare the model compression methods including pruning, quantification, and low-precision compression on Transformer and RNN models. Finally, we achieve 5.8× and 11.7× compression ratio on the Transformer and RNN models by a combined approach, while maintaining the same BLEU score.

Keywords: model compression; pruning; quantization; low-precision; machine translation

0 引言

当前深层神经网络在很多任务上都取得了巨大的进步[1-3]，但随着网络规模的增大，网络中的参数增多，网络变得更冗余和繁重[4]，这种冗余的网络不仅会对存储资源造成浪费，还会导致严重的过参数化和过拟合问题[5]。相比传统的统计机器翻译，神经机器翻译在翻译质量上有了显著提升；并占用更少的存储空间，但由于神经网络仍然规模巨大，受网络计算复杂度以及有限内存、有限存储资源、有限能源的限制，高性能的基于深层神经网络的神经机器翻译系统只能在有限的平台下使用，无法移植到资源和能源都有限的小型嵌入式系统当中[6]，如何有效减小机器翻译系统模型存储便成了一个亟待解决的问题。

针对此问题，研究人员引入模型压缩方法来进行模型的存储压缩，在降低模型存储需求的同时保证模型的性能不受损害。模型压缩的目标分为两种：(1)减少运行时刻内存/显存消耗，指通常所说的运行时刻模型压缩；(2)减少模型在磁盘上的存储消耗(但是运行内存并不减小)，这个方法可以把模型变得更小，有利于模型在设备间的传输。更小存储的模型有很多优点，比如，小的存储空间可以使

收稿日期：2018-09-21 **定稿日期**：2018-10-15

基金项目：国家自然科学基金(61876035, 61432013, 61732005)；中央高校基本科研业务费；辽宁省高等学校创新人才支持计划

得很多手机端机器翻译应用使用更小的离线下载包而提升用户体验，减小带宽并提高并行性，本文实验针对第二种目的。

模型压缩方法总体可以分为两大类：第一类方法修改模型结构，减小模型存储大小[7-9]，设计更精细的网络结构；第二类方法不改变模型结构，在现有模型基础上减少模型参数以减小模型存储。本文在第二类方法上进行研究。研究人员在对第二类模型压缩方法的研究中提出了很多可行方法，如剪枝[10]、量化[5]、低精度[11]、哈夫曼变长编码[12]、知识精炼[13]等。本文针对现有剪枝、量化和低精度三种方法，基于 Transformer 和 GNMT[14] 双层 RNN 基线系统在 NIST12 数据集上进行实验，对比分析以上三种模型压缩方法在两种模型上可达到的不同压缩效果及其原因，本文中实验仅针对模型存储空间进行压缩，并未对模型运行内存进行压缩。

实验中，为了分析使用不同模型对每种模型压缩方法效果的影响，我们选取了 RNN 和 Transformer 两个基线系统。在最后的实验结果中，对于 RNN 和 Transformer，单独使用剪枝方法可分别剪掉 40%和 15%模型参数而保持模型性能不变；单独使用量化方法，在保证模型性能的同时可以将 RNN 最多由 32-bit 浮点数压缩至 4-bit，可以将 Transformer 的参数最多由 32-bit 浮点数压缩至 6-bit；单独使用半精度方法可将两套模型均压缩至原有大小的 50%而没有任何性能损失。由于模型在解码过程中仅需要一部分参数，对其余模型参数我们不进行存储，在仅存储解码相关参数的基础上，使用剪枝、量化、低精度三种方法的组合方法，可在不损失原有模型精度的前提下在 Transformer 和 RNN 模型上达到 5.8×和 11.7×的压缩率。

1 模型压缩方法

本节对本文实验中所使用的模型压缩方法进行具体介绍。模型压缩方法包括剪枝、量化、低精度中的半精度方法。

1.1 剪枝

模型裁剪的方法最早由 Cun 等人提出[10]，是目前模型压缩中使用最广泛的方法，其主要思想是通过减少参数空间中的冗余来缩小模型规模。近年来，在基于卷积神经网络的计算机视觉领域任务中，模型裁剪方法已经取得了不小的成果，有些研究工作甚至可以在完全不损失模型性能的前提下大幅裁剪模型的参数[15]。

我们以 RNN 模型为例观察模型的参数分布，从图 1(a)可以看出，神经网络中绝大部分参数分布在 0 点附近，在整个参数分布中，裁剪掉越靠近 0 的参数往往对模型造成的破坏就越小。在实验中，我们可以硬性设置一个阈值，裁剪掉绝对值低于这个阈值的权值或者整个神经元，也可以裁剪掉所有权值绝对值最小的前 $x\%$。总体来说，剪枝的整个过程就是一个将稠密的网络变成稀疏网络的过程。

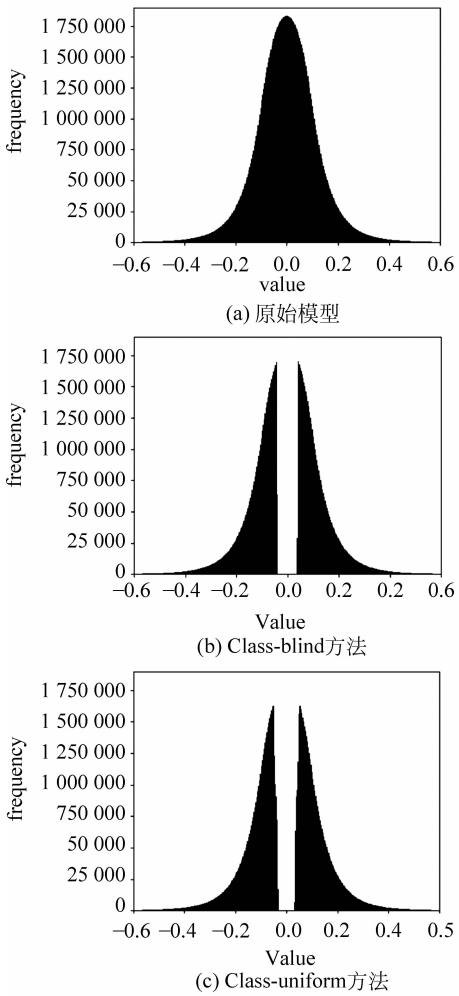

图 1 模型参数分布图

本文中实验中采取了 Class-blind 和 Class-uniform 两种剪枝策略[16]。Class-blind 方法是将模型中所有参数按绝对值大小进行排序，裁剪掉绝对值最小的前 $x\%$参数，这种方法可以保证裁剪掉的模型参数值一定最接近 0，缺点是导致模型中每层的剪枝比例不均匀。Class-uniform 方法首先需对模型中参数分层，在每层中剪掉绝对值最小的前 $x\%$参数来保证每层的剪枝比例均匀，缺点是若某一层

中参数绝对值普遍偏大或这一层中大部分参数对模型来说都很重要，硬性的分层剪枝会对模型精度造成更大损失。

按 Class-blind 和 Class-uniform 方法进行剪枝后模型参数分布如图 1(b)和(c)所示，由图中可以看出按 Class-blind 方法比 Class-uniform 方法所剪参数分布更集中，Class-uniform 方法所剪参数分布范围更广，体现在图中就是图 1(b)和图 1(c)中间无参数部分宽度相比，图 1(c)中间空白部分比图 1(b)更宽且更有层次。

1.2　量化

量化方法最早由 Oliver 等人提出[17]，Denil 等人的工作将量化方法用于神经网络模型的压缩并证实了神经网络当中的过参数化问题[5]。在神经网络当中，参数都是用 32-bit 浮点数表示，量化方法实际上就是通过牺牲参数精度的方式减少每个模型参数存储所需要的比特位数。

在进行量化实验时，我们首先需选定量化区间，量化区间是执行量化操作的区域，对于量化区间外的值仍以原值保存，由于神经网络参数总体呈正态分布，选定量化区间以 0 为中心。以 1--bit 量化为例，量化过程如图 2 所示，量化过程即用索引值代替量化区间内原值的过程，当进行 n-bit 量化时，将选定量化区间分为 2^n 个等长区间，每个区间依次由索引 0 至 2^n-1 来表示。进行解码时，需将索引值恢复为原区间内的值，本文实验中我们分别将参数值恢复为原区间的左值、中值和右值，并进行对比分析。

经过量化，模型以一个更低精度的形式来保存，使其成功地嵌入到小型设备上成为可能[18]。量化过程中，可以将网络的参数量化至固定的 4～12-bit[19]，更极端的情况下也可以用 1～2-bit 来表示模型参数[20]。

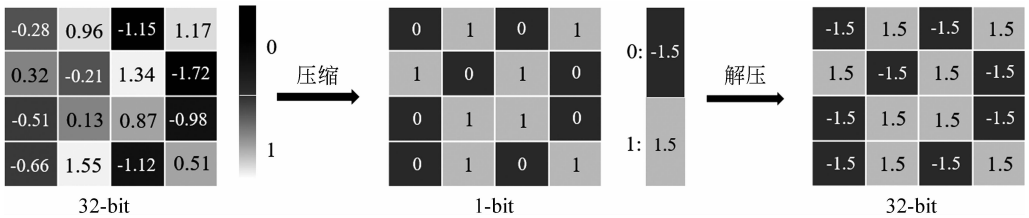

图 2　量化过程图

1.3　低精度

神经网络中的权值都用浮点数表示，单精度浮点数在计算机中用 32-bit 表示，其中符号位 1-bit，指数位 8-bit，尾数位 23-bit。在进行模型的训练时，保持权值的高精度可以确保得到更优的模型，而当模型训练完之后，权值精度适度降低一般对模型性能影响不大。低精度与量化方法思想相近，都是通过改变模型参数表示的形式来达到压缩存储的目的，在低精度方法中，通过将浮点数的尾数部分去掉来节省模型参数存储空间。

本文采取低精度中的半精度方法进行实验，在半精度方法中把 32-bit 单精度浮点数裁剪成 16-bit 半精度浮点数来进行参数的压缩存储。具体实验时可分为压缩和解压两个步骤，压缩步骤中将原有的 32-bit 直接舍去尾数位中的 16-bit，在解压缩步骤中把舍去的 16-bit 用 0 补全恢复为 32-bit，压缩和解压过程如图 3 所示。

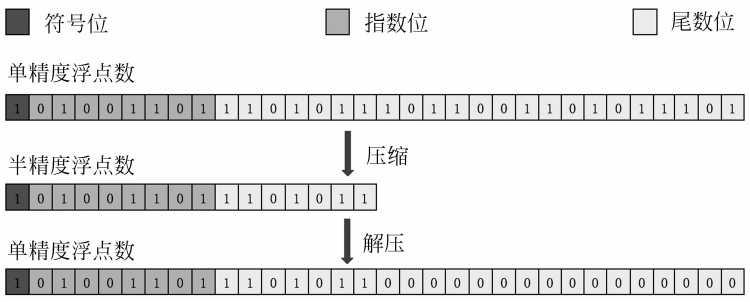

图 3　半精度方法参数压缩和解压过程

2 基线系统

本文基于 Tensorflow 开源框架，搭建 Transformer 和 GNMT 双层 RNN 两套基线系统。主要对比分析剪枝、量化和半精度三种方法在 Transformer 和 RNN 两种模型上不同的实验效果及出现这些实验现象的原因。

本文中使用 Transformer[21] 作为实验基线之一。Transformer 由 Google 在 2017 年提出，此模型摒弃了传统的循环神经网络结构和卷积神经网络结构，是一个完全基于自注意机制的网络结构，模型的优点是提高了模型的可并行程度，减少了模型的训练代价。

Transformer 编码器由 6 个相同的层堆叠而成，每层都有两个子层，第一个子层是多头自注意力机制，第二个子层是一个简单的，对位置敏感的前馈神经网络。解码器同样由 6 个相同的层堆叠而成，其中每层有三个子层，其中有两个子层和编码器相同，第三个子层是负责处理编码器输出的多头注意力机制。为了减小训练压力并加速模型收敛，在编码器和解码器中每一个子层后都会有残差连接和层正则化操作。

Transformer 使用的多头注意力机制（mulit-head attention）的基本单元是按比例的点积注意力（scaled dot-product attention），这种注意力的输入分为 K、Q、V，计算过程如下所示

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V \tag{1}$$

由于在点积的注意力机制上我们进行了按比例缩放，缩放后每个参数都被映射至更小的值，使模型参数范围压缩至更小的区间内，因此 Transformer 相较于传统的 RNN 参数分布范围更小。

表 1 Class-blind 和 Class-uniform 剪枝策略对 Transformer 和 RNN 的性能影响

Class-blind 方法	Transformer	剪枝比例/%	0	5	10	15	20	25	30	35	40	45
		BLEU	41.76	41.81	41.94	41.61	41.39	40.80	39.88	38.02	34.49	24.16
	RNN	剪枝比例/%	0	5	10	15	20	25	30	35	40	45
		BLEU	39.55	39.52	39.70	39.48	39.47	39.50	39.36	39.46	39.50	39.04
Class-uniform 方法	Transformer	剪枝比例/%	0	5	10	15	20	25	30	35	40	45
		BLEU	41.76	41.67	42.07	41.94	41.52	41.28	39.82	38.84	36.41	29.75
	RNN	剪枝比例/%	0	5	10	15	20	25	30	35	40	45
		BLEU	39.55	39.42	39.57	39.61	39.52	39.70	39.42	39.42	39.10	39.16

2.1 实验设置

本文实验主要在中英翻译任务上进行，评价指标采用 Moses 公开计算 BLEU 的脚本 multi-bleu.pl，对于 Transformer 和 RNN，实验数据均采用 NIST12 OpenMT① 作为训练集，其中包括 39.42MB 的中文词和 44.92MB 的英文词，用 NIST2006（MT06）作校验集和测试集，所有中文句子都采取 NiuTrans[22] 提供的工具进行分词，长度超过 50 词的句子都被过滤掉，Transformer 和 RNN 中均使用词频最高的 30KB 大小的词表，所有词表外的词都用 UNK 来代替。

模型结构方面，在 Transformer 中，编码端和解码端各有六层，RNN 编码端和解码端各为两层。模型参数方面，Transformer 基线系统中训练参数共 90MB，训练参数占存储空间大小 362MB，RNN 训练参数共有 134MB，占存储空间大小 537MB。

2.2 实现细节

对于 Transformer，本工作采用 tensor2tensor 的 1.0.14 版本，模型编码端和解码端各有六层，隐藏层维度大小为 512，batch 大小为 4 096，num_heads 为 8，最大句长为 50，残差 dropout 为 0.1，初始学习率设置为 0.001，优化器采用 Adam，训练 15 轮，没有采用 BPE、Ensemble 和 Finetune。

对于 RNN 基线系统，本工作采用 GNMT 的 RNN，模型编码端和解码端各有两层且编码器为单

① LDC2000T46，LDC2000T47，LDC2000T50，LDC2003E14，LDC2005T10，LDC2002E18，LDC2007T09，LDC2004T08

向结构，隐藏层维度大小为 1 024，batch 大小为 128，最大句子长度设为 50，dropout 为 0.2，初始学习率为 1.0，优化器采用 SGD（stochastic gradient descent），训练步数为 150k，没有采用 BPE、Ensemble 和 Finetune。

3　对比分析

3.1　剪枝方法对比分析

本部分实验基于 Transformer 和 RNN 两套系统，运用 Class-blind 和 Class-uniform 方法进行模型剪枝，主要对比分析两种剪枝方法在两种模型上不同的剪枝效果。在不损失模型性能的前提下，Class-blind 方法在 Transformer 和 RNN 上分别可以剪枝 15%和 40%；Class-uniform 方法在 Transformer 和 RNN 上分别可以剪枝 15%和 25%。剪枝方法在 RNN 及 Transformer 上的实验结果如表 1 所示，其中剪枝比例为 0 即代表基线，剪枝过程中模型性能整体变化趋势如图 4 所示。

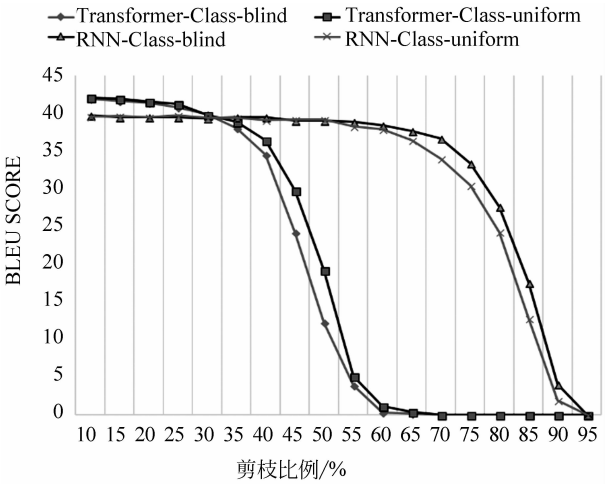

图 4　Transformer 和 RNN 在 Class-blind 和 Class-uniform 剪枝方法下的性能对比

3.1.1　相同剪枝方法在不同模型上的效果分析

此部分主要介绍在相同的剪枝方法下，随着剪枝比例的增大，Transformer 和 RNN 的性能变化趋势。本文中采取 Class-blind 和 Class-uniform 两种剪枝策略，具体方法见 1.1 节。值得注意的是，在进行 Class-uniform 剪枝实验时，由于需要分层进行剪枝，我们首先将神经网络模型按结构分成五部分，Source embedding、Target embedding、Encoder、Decoder 和 Softmax，在分层实验时，我们在 Encoder 和 Decoder 中进一步按照不同模型结构进行分层，其中 Transformer 中有 6 层，RNN 中有 2 层。

从表 1 实验结果可以看出，随着剪枝比例的增大，BLEU 值趋向于变低，剪枝比例越大，BLEU 值下降得越快，这与我们的预期相符，被剪枝参数数量越多，必然对模型造成的破坏越大，BLEU 值就会越快地降低。

从图 4 实验结果看出，无论采取 Class-blind 还是 Class-uniform 方法，RNN 在保持模型精度的条件下剪枝程度都大于 Transformer，这说明在剪枝比例相同的情况下，Transformer 的性能损失明显大于 RNN。从总体剪枝实验情况来看，RNN 最高可以剪到 40%性能不掉，而 Transformer 最高只能剪到 15%左右。这也从侧面说明了 Transformer 模型中相较于 RNN 中冗余参数更少。

造成以上实验现象的原因与 RNN 和 Transformer 的参数分布特点有关。虽然二者参数均服从正态分布，但 Transformer 较 RNN 参数分布得更集中，导致其参数矩阵的方差更大，因此其对于 0 点附近的值不敏感，而剪枝时我们剪掉的是 0 附近的值，因此对于 RNN 可以剪掉较大比例的参数值而不对模型性能产生影响。Transformer 则与 RNN 相反，其参数分布集中，因此其对于 0 附近的值比较敏感，导致剪枝对 Transformer 的影响更大。

由此可见，剪枝方法更适用于压缩方差较大、分布离散的模型。

3.1.2　不同剪枝方法在相同模型中的效果分析

此部分主要介绍 Cass-blind 和 Class-uniform 两种剪枝策略分别在 Transformer 和 RNN 上进行实验的实验现象以及出现这些现象的原因，剪枝过程中模型性能整体变化趋势如图 4 所示，剪枝比例小于 10%时，两种模型的 BLEU 值均没有降低，图中两套模型的起始点即为其各自的基线 BLEU 值。

对于 RNN 来说，随着剪枝比例的增大，在剪枝比例小于 50%时，两种剪枝策略在同一剪枝比例所达到的模型性能相同，当剪枝比例大于 50%时，按 Class-blind 策略进行剪枝的性能明显优于 Class-uniform，Class-blind 策略剪枝达到性能更接近于基线系统，原因在于 RNN 在每层之间的参数分布范围差异较大，而且 RNN 中参数值普遍偏大，运用 Class-blind 方法进行剪枝可以保证所剪参数一定是最接近于 0，而对于 Class-uniform 方法，按比例硬性剪掉每层中参数则会导致剪掉 RNN 中绝对值较大的参数，从而对模型的性能产生影响。

对于 Transformer 模型来说，随着剪枝比例的

增大，当剪枝比例小于15%至20%时，两种方法在性能上无明显差异，当剪枝比例大于这一比例时，按Class-uniform策略进行剪枝的模型性能明显优于Class-blind方法，这与在RNN上实验现象相反，主要是因为Transformer每层间参数分布差异较小，按Class-uniform策略进行剪枝可以发挥该方法的优势，即Class-uniform方法可以使模型的剪枝更加均匀，并在每个分类中都剪掉更多对于模型冗余的参数。

总体来说，对于RNN，由于RNN参数范围偏大且各层间参数分布不均匀，因此用Class-blind方法比Class-uniform剪枝效果好。对于Transformer，由于模型参数范围小且各层间参数分布较均匀，因此在其上用Class-uniform来剪枝效果更好。

3.1.3 相同剪枝方法在相同模型上不同层次效果分析

在进一步分析模型层次对剪枝效果的影响的时候，我们在Transformer和RNN上分别进行实验，实验中统一采取Class-blind方法，在其余层不变的情况下对每一层进行单独剪枝，我们对编码器和解码器中的每一层都单独进行了实验。由于选取的两套基线系统中Transformer层数较多，我们以其为例分析模型层次加深对剪枝效果的影响以得到更有合理性的实验结果，在保持模型性能的前提下各层可以进行剪枝的最大比例如图5所示。

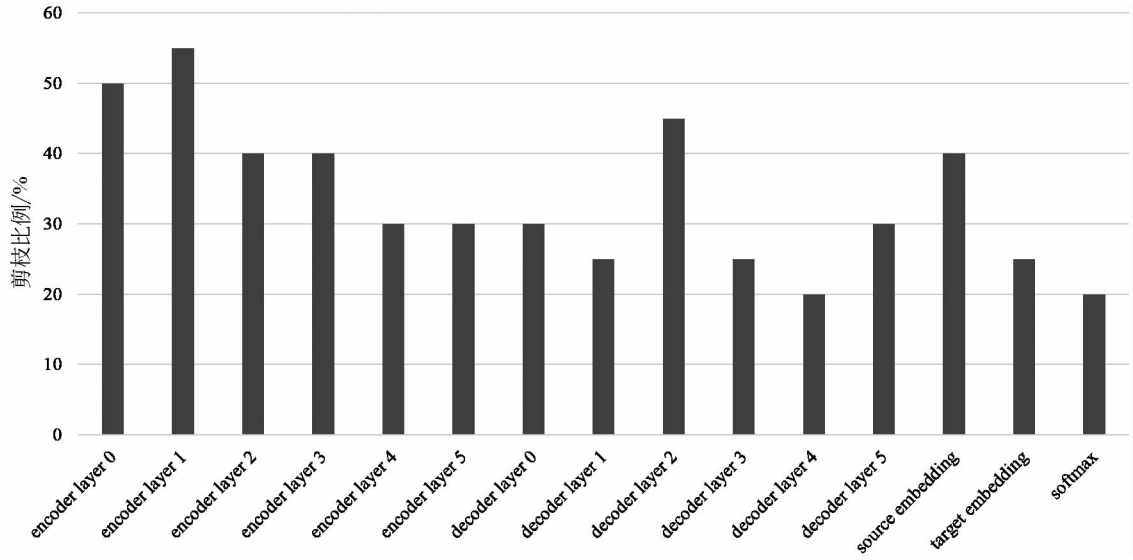

图5 Transformer分层剪枝效果图

从实验结果看出，随着模型层次的加深，在保持模型性能的前提下剪枝比例逐渐减小，原因在于神经网络层与层之间是递进关系，模型更高层中包含前面所有层中信息，也就是说越高层中包含重要的信息越多，因此越高的层次趋向于对剪枝更敏感。

在图5实验结果中发现，对剪枝越敏感的层趋向于有更大的权值，比如此模型中decoder layer 1参数分布范围较大，对剪枝比较敏感。也就是说，在剪枝相同比例的情况下，对参数越大的层进行剪枝对模型造成的损失越大。在实验中发现，模型中每一层的参数大小具有一定随机性，因此具体每一层中剪枝比例缺乏普适性。

在Transformer上进行分层剪枝实验时，若将所有层中保持模型性能的最大比例相结合进行组合剪枝实验，剪枝比例如图5所示，当基线系统BLEU值为41.76时组合剪枝可以达到41.65的BLEU值。当我们在RNN模型上用相同方法进行尝试性实验时，将每层中可以达到基线性能的最大剪枝比例相结合后模型性能却很差，在基线BLEU值为39.55时组合剪枝BLEU值仅达到39.07。产生以上实验现象的一部分原因是由于RNN模型中每层剪枝的幅度比Transformer大，而神经网络层与层之间都联系十分紧密，改变其中任何一层的参数都可能对其他层产生不可预估的影响。另一部分原因是由于RNN模型每层间参数分布没有Transformer分布均匀，对单独层进行剪枝会对其他层产生更大的影响。

3.2 量化方法对比分析

实验中将Transformer和RNN原始训练系统

作为基线，运用 n-bit 量化方法对模型进行压缩，由于仅压缩了存储空间，解码时需将参数恢复为 32-bit 浮点数。在量化的恢复阶段，我们分别将参数值恢复为区间中左值、中值和右值和进行实验，在进行 8-bit量化过程中，两种模型均可在保持模型性能不变的前提下，取到包含全部模型参数 99％以上的量化区间。

3.2.1　量化时取值方式对模型性能的影响分析

为了验证量化恢复时取值方式对模型性能的影响，我们在 Transformer 上进行实验，对于同一量化区间（包含全部参数范围的 99.1％），在量化的恢复阶段我们分别将参数值恢复至量化区间中左值、中值和右值，实验结果如图 6 所示。

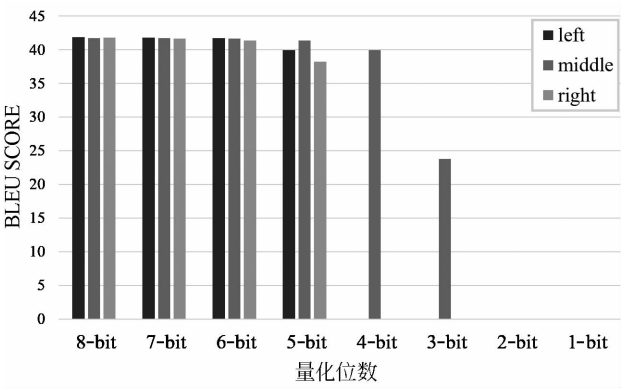

图 6　Transformer 量化时取值方法对模型性能的影响

从图 6 中可以看出，随着 n-bit 量化中 n 的减小，开始时取左值、中值和右值模型性能无明显差异，但当 n 逐渐减小直至小于 5 时，取左值和右值对模型性能破坏极大，当 $n=4$ 或更小时，BLEU 值几乎降为 0。也就是说随着 n 的减小，量化恢复时取中值的优势越来越明显，取中值时模型的性能会明显优于取左值和右值。

这也验证了我们的一个想法，一个区间的中值会接近区间中的大部分参数，因此将量化后参数恢复为 32-bit 时用中值来代替一个区间的参数会更加合理。

3.2.2　Transformer 和 RNN 对量化方法的敏感程度分析

为了验证 Transformer 和 RNN 两套模型对量化方法的敏感程度，我们在 8-bit 量化的基础上对模型进行了更精细的 n-bit 量化实验，这部分的所有实验在参数恢复阶段都将参数恢复至区间中值，实验结果如图 7 和表 2 所示。

图 7 显示内容为进行 n-bit 量化时在保证模型

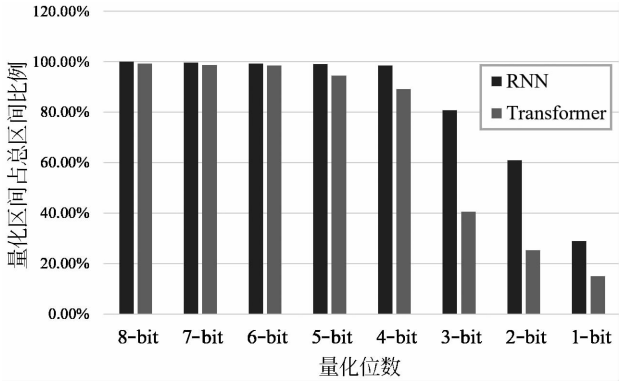

图 7　Transformer 和 RNN 最大量化区间趋势图

性能不变的情况下所能取到的最大范围的量化区间。从图中可以看出，随着 n-bit 量化实验过程中 n 的减小，所取量化区间也逐渐减小，这与我们期望相符。n 越小，可以取到索引值的总数 2^n 就越少，量化区间分成等长区间的份数越小，对模型造成的破坏就越大。因此随着 n 的减小，本文倾向于选择更小范围的量化区间。从图中还可以看出 Transformer 相较于 RNN 对量化更敏感，量化为相同的比特数时 Transformer 的量化区间更小。产生以上实验现象的原因与 Transformer 模型结构有关，由于 Transformer 有一个参数矩阵缩放过程，模型参数被缩小到一定范围内，导致模型参数变化对 Transformer 造成破坏更大。

单独使用 n-bit 量化方式时，随着 n 的减小，模型压缩率变化趋势如图 8 所示。模型压缩率为

$$r_{\text{quan}}=\frac{n\cdot b}{x_{\text{quan}}\%\cdot n\cdot b_{\text{index}}+(1-x_{\text{quan}}\%)\cdot n\cdot(b+b_{\text{mark}})} \tag{2}$$

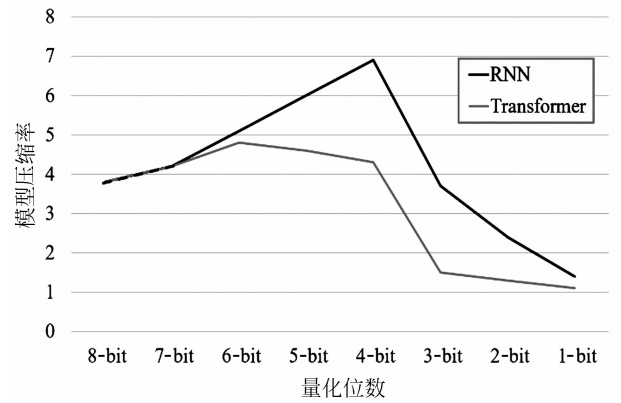

图 8　Transformer 和 RNN n-bit 量化模型压缩率变化

其中 r_{quan} 代表 n-bit 量化后模型压缩率，x_{quan} 代表量化区间占总参数区间的百分比，n 代表总模型参数数量，b 代表每个模型参数原本需要用来表示

的比特位数，b_{index}代表用来表示模型参数索引需要的比特位数，b_{mark}代表用来区分是否在量化区间外的标志的比特位数。

从图8可以看出，随着n-bit量化中n的减小，模型压缩率均呈现出先增大后减小的趋势。产生以上实验现象的原因主要是由于在模型达到最大压缩程度前，随着n的减小，量化区间内的参数值被极大程度压缩，量化区间外的值所占比例比较小，模型压缩率增大；在模型达到最大压缩程度后，随着n的减小，虽然量化区间内的参数值被压缩，但量化区间缩小后量化外区间参数所占范围增大，整体模型压缩程度减小。

3.3 半精度方法对比分析

在实验中我们发现，无论对于Transformer还是RNN模型，用半精度方法进行全范围参数的压缩均不会对其造成性能上的损失，这也向我们证明了神经网络中存在冗余。由于在半精度方法中通过把32-bit单精度浮点数裁剪成16-bit半精度浮点数来进行参数的压缩存储，因此用半精度方法可以将模型压缩至原有大小的50%而不对模型性能产生影响。

而半精度模型压缩方法在以上两个模型中的使用效果也向我们证明，在神经网络中我们并没有必要使用32-bit单精度浮点数来保存参数，尾数位的后16位对模型性能没有任何意义，用16-bit或更少的比特数来保存参数对神经网络模型就已经足够，多余的比特位会导致网络的冗余，并且对存储资源造成十分不必要的浪费。

基于以上实验结果，本文的下一步工作计划就是尝试使用更低精度的参数，如整数，进行模型的推断和训练，不仅仅可以减少模型的存储空间，还可以节省模型运行内存及更多的计算资源。

3.4 总体压缩程度分析

进行剪枝、量化、半精度三种方法的组合实验的实验结果如表2所示。三种模型压缩方法的参数操作范围为：total＝半精度＞量化＞剪枝，其中total代表全部参数。

表2 模型压缩组合实验的实验结果

Network		BLEU	参数	压缩率
RNN	baseline	39.55	537MB	1×
	剪枝(40%)	39.50	323MB	1.6×
	剪枝＋量化(4-bit＋98%)	39.44	51MB	10.5×
	剪枝＋量化＋半精度	39.43	46MB	11.7×
Transformer	baseline	41.76	362MB	1×
	剪枝(15%)	41.94	308MB	1.2×
	剪枝＋量化(6-bit＋98%)	41.50	65MB	5.6×
	剪枝＋量化＋半精度	41.57	62MB	5.8×

对于RNN，在组合实验中，我们采取40%剪枝，4-bit量化(量化区间为98%，其中98%包括剪枝的40%)，全范围的半精度压缩，最终能达到11.7×的压缩率；对于Transformer，我们采取15%范围剪枝，6-bit量化(量化区间为98%，98%包括剪枝的15%)，全范围的半精度压缩，最终能达到5.8×的压缩率。可以看出，RNN的可压缩程度明显大于Transformer，二者在总体压缩程度上的区别也可反映出Transformer相比于RNN冗余较小。

从表2实验结果可以看出，在剪枝、量化、半精度三种模型压缩方法组合实验过程中，量化方法对整个模型压缩率的影响最大。与剪枝方法相比，量化方法操作的参数区间范围大，与半精度方法相比，量化将参数存储压缩的程度更大，因此在实验过程中量化方法占主要优势，经过量化后，模型被极大地压缩。

3.5 更深层网络分析

为了进一步验证Transformer和RNN之间压缩程度的差异，本文在更深层网络(4层和8层)的GNMT RNN模型上进行实验，在两个更深层RNN上得到的最大压缩率如表3所示。

进行更深层 RNN 网络的实验进一步证明了我们之前实验结果的普适性，RNN 可压缩程度比 Transformer 更大，从侧面反映出 RNN 中冗余参数较多。

表 3　更深层网络实验结果

Network		BLEU	参数	压缩率
RNN-4layer	baseline	40.37	648MB	1×
	剪枝(45%)	40.35	357MB	1.8×
	剪枝＋量化(4-bit＋98%)	40.33	58MB	11.3×
	剪枝＋量化＋半精度	40.34	51MB	12.7×
RNN-8layer	baseline	38.10	1048MB	1×
	剪枝(35%)	38.00	682MB	1.5×
	剪枝＋量化(4-bit＋98%)	37.91	107MB	9.8×
	剪枝＋量化＋半精度	37.96	97MB	10.9×

4　下一步工作

本文，实验对神经网络模型的压缩进一步证明了神经网络中模型参数的冗余特性。针对该特性，我们在未来工作中会考虑进行神经网络的整数或更低精度的训练和推断，将模型压缩与模型加速相结合，在减小模型存储空间的同时也达到模型加速的目的。

5　结论

本文主要针对如何降低模型参数存储进行实验，对比不同压缩方法在 Transformer 和 RNN 上的模型压缩效果。

对于剪枝方法，Transformer 这种参数分布范围较小且每层间参数分布较均匀的模型，进行分层剪枝效果更好，RNN 这种模型参数分布范围广且每层间参数分布相差较大的模型运用整体剪枝效果更好，越深的层次趋向于对剪枝越敏感。

对于量化方法，参数分布集中的 Transformer 模型比参数分布离散的 RNN 模型更敏感，在进行量化时应注意选择合理的量化区间和 n 值。

对于半精度方法，无论在 Transformer 还是 RNN 模型上，半精度方法均可在全参数范围内进行模型的压缩而不会对模型造成任何损失。

在进行剪枝、量化、半精度三种模型压缩方法的组合实验中我们发现，Transformer 比 RNN 可压缩程度小，从侧面反映出 Transformer 相对于 RNN 冗余参数较少。

参考文献

[1] Hinton G, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups[J]. IEEE Signal Processing Magazine, 2012, 29(6): 82-97.

[2] Collobert R, Weston J. A unified architecture for natural language processing: Deep neural networks with multitask learning[C]//Proceedings of the 25th International Conference on Machine Learning, 2008: 160-167.

[3] Zhang J, Zong C. Deep neural networks in machine translation: An overview[J]. IEEE Intelligent Systems, 2015, 30(5): 16-25.

[4] Dean J, et al. Large scale distributed deep networks [C]//Proceedings of the 25th International Conference on Neural Information Processing Systems. Curran Associates Inc. 2012: 1223-1231.

[5] Denil M, et al. Predicting parameters in deep learning [J]. Neural Information Processing Systems, 2013: 2148-2156.

[6] Howard A G, et al. MobileNets: Efficient convolutional neural networks for mobile vision applications[J]. arXiv preprint arXiv: 1704.04861, 2017.

[7] Zhang X, et al. ShuffleNet: An extremely efficient convolutional neural network for mobile devices[J]. arXiv preprint arXiv: 1707.01083, 2017.

[8] Iandola F N, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size [J]. arXiv preprint arXiv: 1602.07360, 2016.

[9] Lin M, et al. NetworkIn network[J]. arXiv preprint

arXiv: 1312.4400,2014.

[10] Cun Y L,Denker J S,Solla S A. Optimal brain damage[M]. Davids T. Advances in Neural Information Processing Systems2. San Francisco: Morgan Kanfmann Publisher Incs,1990: 598-605.

[11] Courbariaux M,Bengio Y,David J P. Training deep neural networks with low precision multiplications [J]. arXiv preprint arXiv: 1412.7024,2015.

[12] Mamta Sharma. Compression using huffman coding [J]. International Journal of Computer Science and Network Security: IJCSNS,2010,(5): 133-141.

[13] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network [J]. Computer Science, 2015,14(7): 38-39.

[14] Wu Y,et al. Google's neural machine translation system: Bridging the gap between human and machine translation[J]. arXiv preprint arXiv: 1609.98140, 2016.

[15] Han S,Mao H,Dally W J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding[J]. Fiber,2015,56 (4): 3-7.

[16] See A,et al. Compression of neural machine translation models via pruning[C]//Proceedings of the Conference on Computational Natural Language Learning,2016: 291-301.

[17] Oliver B M,Pierce J R,Shannon C E. Philosophy of PCM[J]. Proceedings of the Ire,1948,36(11): 1324-1331.

[18] Mishra A, Marr D. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy[J]. arXiv preprint arXiv: 1711.05852, 2017.

[19] Lin D D,et al. Fixed point quantization of deep convolutional networks[C]//Proceedings of the International Conference on Machine Learning,2016: 2849-2858.

[20] Courbariaux M, et al. Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or −1[J]. arXiv preprint arXiv: 1602.02830,2016.

[21] Vaswani A,et al. Attention is all you need[J]. Neural Information Processing Systems,2017: 5998-6008.

[22] Tong Xiao,et al. Niutrans: An open source toolkit for phrase-based and syntax-based machine translation[C]//Proceedings of the ACL 2012 System Demonstrations, 2012, 19-24. Association for Computational Linguistics.

林野(1995—),硕士研究生,主要研究领域为机器翻译。
E-mail: linye2015@outlook.com

肖桐(1982—),通信作者,博士,副教授,主要研究领域为机器翻译。
E-mail: xiaotong@mail.neu.edu.cn

姜雨帆(1994—),硕士研究生,主要研究领域为机器翻译。
E-mail: jiangyufan2018@outlook.com