基于递进式半知识蒸馏的神经机器翻译

摘要：神经机器翻译（NMT）模型通常具有庞大的参数量，例如 Transformer 在词表设为 3 万时有将近 1 亿的神经元，模型的参数量越大，模型越难优化且存储模型的资源需求也越高。本文提出了一种压缩方法用于将复杂且参数量大的 NMT 模型压缩为精简参数量小的 NMT 模型。本文提出半知识蒸馏方法和递进式半知识蒸馏方法，其中半知识蒸馏是从参数多、性能好的教师模型中获取半部分的权重作为精简、参数少的学生模型训练的起点；递进式半知识蒸馏方法指运用过一次半知识蒸馏方法压缩以后，再把当前的半知识蒸馏压缩的模型作为新的教师模型，再次运用半知识蒸馏方法得到全压缩模型。在广泛使用的中英和日英数据集上进行实验，结果表明本文方法对 NMT 系统有积极影响。本文提出的方法的最佳性能明显优于基准模型 2.16 个 BLEU 值。与词级别和句子级别的传统知识蒸馏方法相比，本文提出的方法比词级别知识蒸馏方法优于 1.15 个 BLEU 值，并且高于句子级别的知识蒸馏方法 0.32 个 BLEU 值。

关键词：机器翻译；模型压缩；知识蒸馏

Progressive Semi-knowledge Distillation for Neural Machine Translation

Abstract： The neural machine translation (NMT) model usually has a large amount of parameters. For example, Transformer has nearly 100 million neurons when the vocabulary is set to 30,000. The larger the parameter size of the model, the more difficult it is to optimize the model and the resource requirements of the storage model. This paper proposes a compression method for compressing a complex and large-parameter NMT model into a NMT model with a small amount of reduced parameters. This paper proposes a semi-knowledge distillation method and a progressive semi-distillation method, in which semi-knowledge distillation is to obtain half-weight from the teacher model with many parameters and good performance as the starting point for the training of student models with reduced and few parameters; progressive half The distillation method refers to the compression of the current semi-knowledge distillation method as a new teacher model after using the one-and-a-half knowledge distillation method, and the semi-knowledge distillation method is used again to obtain the full compression model. Experiments on two widely used two language datasets show that the proposed method has a positive impact on the NMT system. The best performance of the proposed method is significantly better than the 2.16 BLEU values of the benchmark model. Compared with the word-level and sentence-level traditional knowledge distillation methods, the proposed method is better than the word-level knowledge distillation method by 1.15 BLEU values and higher than the sentence-level knowledge distillation method by 0.32 BLEU values.

Key words:机器翻译；模型压缩；知识蒸馏

0 引言

机器翻译包括统计机器翻译和神经机器翻译，近年来，神经机器翻译（Neural machine translation，NMT）系统[1,2,3,4,5]取得了巨大成功，成为机器翻译主流方法。神经网络的机器翻译模型的性能很大程度上受限于神经网络参数规模和训练数据。神经机器翻译模型参数过多也会存在一些问题，例如：模型的存储空间要求变高、优化时间长等。因此神经网络的庞大参数量限制了神经机器翻译系统在手机等小型设备上应用。

收稿收稿日期：；定稿日期：
基金项目：

模型压缩是解决此类问题的一类有效方案。现有的压缩方法有剪枝、量化、低精度和知识蒸馏等方法。剪枝是通过移除对目标函数几乎不产生影响的权重[6]，或者是对隐藏层单元进行阈值处理[7]。知识蒸馏是通过引入教师网络（Teacher Network）的相关知识或信息作为学生优化中的一部分，以启发学生网络（Student Network）的训练[8]，其中教师网络有模型复杂但性能优越的特点，学生网络具有具有精简低复杂度的特点。目前神经网络压缩现有工作包括模型参数压缩和模型存储压缩两类，一类的目标是减少模型参数，另一类减少存储，模型参数压缩有剪枝、量化、降低精度和知识蒸馏等方法。在深度神经网络压缩方法中，通常剪枝是基于一些准则对权重或神经元进行剪枝;剪枝准则主要包括使用近似海森权重剪枝[6]和基于值的大小进行剪枝[9];以及一些移除神经元的方法[10, 11]。深度神经网络压缩的其他方法包括稀疏正则化[12]；权重矩阵的低秩分解[13,14,15, 16]；权重共享[17,18]和权重的二值化[19]。在机器翻译领域中，See 等人提出使用基于权重的幅值进行修剪或按权重的类别进行剪枝[12]；Kim 等人提出知识蒸馏[20]；林野等对比了剪枝、量化、低精度三种模型压缩方法在 Transformer 和递归神经网络（Recurrent neural network, RNN）两种模型上的效果。[21]

本文提出半知识蒸馏方法来压缩模型。半知识蒸馏方法将已训练的教师模型的部分知识，通过权重赋值直接传递给学生网络，学生网络以此为起点开始训练，对于压缩权重，根据大数定律，采用高斯分布初始化，这样能够充分利用已训练的模型知识来压缩模型。为了使得整个模型完全被压缩，可以递进式运用半知识蒸馏方法来压缩另外一部分的权重，将当前的学生网络充当新的教师，将新的知识传递给更小参数规模的学生网络。本方法可以应用到常见的端到端模型，本文以 Transformer[5] 模型结构为例使用此方法。通过划分 Transformer 参数为分成两个部分，压缩和未压缩部分，然后压缩和未压缩的部分通过中间映射的权重作为压缩兼容性矩阵，以保证压缩的权重转换为与其他非压缩权重兼容，从而不需要添加新的参数，实现压缩整个模型参数规模。

本文进行了充分的实验，实验结果显示递进式知识蒸馏得到的全压缩模型和半压缩模型均比相应的基准模型有显着的改善。半知识蒸馏可以将 Transformer 模型压缩成更小的学生模型，其性能甚至超过教师模型的性能。通过现有工作（词级别和句子级的知识蒸馏[21]）和本文提出的方法进行对比，在中英任务上，本文和词级别和句子级的知识蒸馏性能相当，并且在日英任务上，本文方法获得了比词级别和句子级别知识蒸馏更好的性能。在日英翻译和中英翻译实验中，最佳性能比其相应的基准模型提高了 2.16 个 BLEU 值和 0.722 个 BLEU 值。

1 背景

Transformer 是机器翻译中最成功的 NMT 系统之一，同时也是一个典型的 NMT 系统，具有多层结构且有大规模参数的特点，在许多基于端到端序列建模任务中表现出色。因此，选择它作为基准模型进行模型压缩具有代表性和一般性。与其它基于递归神经网络（Recurrent Neural Network，RNN）[123]和卷积神经网络（Convolutional neural network，CNN）[4]的翻译系统相比，Transformer 仅基于注意力机制进行序列建模。Transformer 的架构组成如图 1 所示。

编码器由L个相同的层堆叠组成，每个层包含两个子层：多头自注意层和前馈神经网络层。每个子层通过残差连接（Residual Connection）[22]进行连接，然后是层规范化[23](Layer Normalization)。可以由 layernorm(h + sublayer(h)) 表示，其中 layernorm(·) 表示层规范化函数的输出和 sublayer(h)表示子层的输出; h表示当前子层输入的隐藏层状态。Transformer 的层与层之间的采用残差连接，因此模型子层输入的尺寸和子层输出的维度相同。

解码器同样由L个相同的层堆叠组成。与编码器不同的是每个层包含三个子层。每一层比编码器多一个子层，该子层是对编码器最后一层的输出进

行注意力权重计算。其余与编码器一样，子层与子层之间应用残差连接和层规范化。

模型在给定原文句子序列$\mathrm{X}=(x_1,\ldots,x_N)$以及目标语言序列$\mathrm{Y}=(y_1,\ldots,y_M)$下，第$i$个目标词的条件概率$\hat{y}_i$可以表示为式（1）所示。

$$\arg\max_{y\in\mathcal{H}} P(\hat{y}_i|\boldsymbol{Y}_{<i};\mathbf{X};\theta)\ \#(1)$$

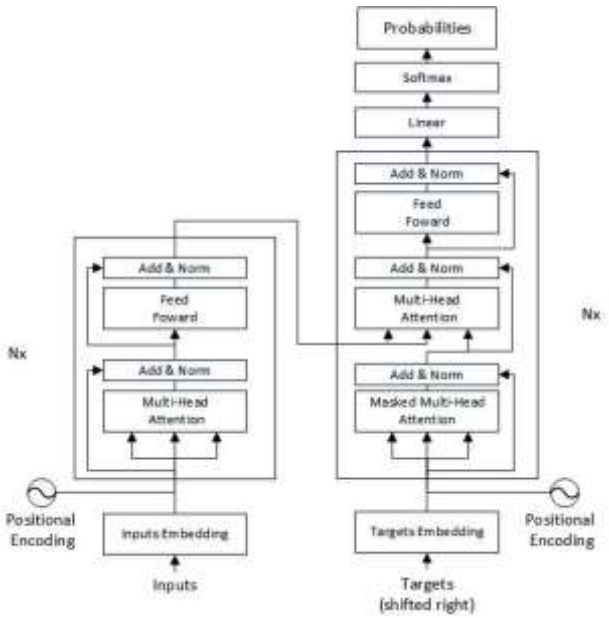

图 1 Transformer 模型架构

其中，$\mathcal{H}$是所有可能候选翻译的集合，$P(y_i|y_{<i};x;\theta)$是生成第i个目标词y_i的条件概率；$\boldsymbol{Y}_{<i}$表示历史生成的单词序列；$\boldsymbol{X}$为源端的句子序列；θ为模型的参数。

注意力机制是式（2）函数的一个统称，描述的一个查询项（Query, Q）以及一组键值对(Key-Value, $K-V$)的之间的进行映射得到注意力结果。

$$\mathrm{Att}(Q,K,V)=\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V\#(2)$$

其中Q,K,V分别表示查询项、键和值。Q,K,V在进行注意力机制计算前，均会经过一个线性映射，得到的注意力结果最终也会经过一个线性映射得到注意力层的输出。线性映射可以表示如式（3）：

$$k=\mathrm{linear}(h)=h*W\#(3)$$

其中$h\in\mathbf{R}^{m\times d}$表示注意力机制的输入；$W\in\mathbf{R}^{d\times q}$表示线性映射的权重，$k\in\mathbf{R}^{m\times q}$表示线性映射后的输出结果;$m$表示句长，$q$表示变换后的维度信息，d表示模型维度信息。

2 知识蒸馏

神经机器翻译模型可以看作是在样本为(x,y)数据集上的一个具有词表大小V类的多分类，通常的训练目标是在训练集上最小化负极大似然损失（Negative likelihood loss，NLL）函数，可以用式（4）表示

$$\mathcal{L}_{NLL}(\theta)=-\sum_{k=1}^{V}\mathrm{I}(y=k)\log p(y=k|X;\theta)\ \#(4)$$

其中$\mathrm{I}(\cdot)$表示指示函数，$\mathrm{I}(y=k)$当y等于k时返回1，否则返回 0。$p(y=k|X;\theta)$是模型输出的概率分布。

知识蒸馏是模型压缩方法的中一类方法，通常表示精简的神经网络（学生）在具有更大规模的、性能更好的神经网络（教师）的指导下进行学习更新[8]。

知识蒸馏的目标函数是让学生网络拟合教师网络输出的概率分布，使得学生模型的输出与教师模型的输出概率$q(y|X;\theta^t)$更加接近[9]，知识蒸馏的目标函数如式（4）所示。

$$\mathcal{L}_{KD}(\theta^s;\theta^t)=-\sum_{k=1}^{V}q(y|X;\theta^t)\log p(y=k|X;\theta)\ \#(5)$$

其中$\mathcal{L}_{KD}(\theta^s;\theta^t)$表示知识蒸馏的目标函数，$\theta^s$表示学生网络的参数，$\theta^t$表示教师网络的参数。

词级别的知识蒸馏是一种标准的知识蒸馏，在原有的最小化$\mathcal{L}_{NLL}(\theta)$的基础上，新增一个目标函数$\mathcal{L}_{KD}(\theta^s;\theta^t)$，并通过一个超参数α来结合$\mathcal{L}_{NLL}(\theta)$和$\mathcal{L}_{KD}(\theta^s;\theta^t)$，表示如式（5）所示。与词级别知识蒸馏方法不同的是，本文提出的方法在训练期间不需要计算教师网络的输出概率分布，让学生网络的概率与之接近，而是通过把教师网络的权重信息指导学生网络，让学生网络的学习起点更高，缩短优化时间。

$$\mathcal{L}(\theta^s;\theta^t)=(1-\alpha)\mathcal{L}_{NLL}(\theta^s)+\alpha\mathcal{L}_{KD}(\theta^s;\theta^t)\#(6)$$

其中θ^s表示学生模型的参数，θ^t表示教师模型的参数。

句子级别的知识蒸馏是学生网络是用教师网络生成的翻译结果，与训练数据的原文构成新的平行句对进行训练。与句子级别的知识蒸馏不同，本文方法进行知识蒸馏是把教师模型的权重信息传递给学生网络，而不改变训练数据集。

3 递进式半知识蒸馏方法

3.1 递进式半知识蒸馏方法

递进式知识蒸馏方法总体概览图可以表示如图 2 所示，图中的三个精简架构是 Transformer 的略缩图，图中虚线是编码器和解码器的分割线；箭头代表压缩前和压缩后的模型尺寸变化的对应关系；图中采用阴影填充的模块表示被压缩维度后的权重，未填充的结构表示未被压缩维度的权重。递进式半知识蒸馏应用半知识蒸馏 2 次，逐次递进式压缩模型。第一次压缩：由图 2 中左侧模型架构应用一次半知识蒸馏方法得到中间模型架构。第二次压缩，是把上一次压缩的模型（即图 2 中间的模型架构）作为新的教师模型，进行第二次半知识蒸馏，由中间的模型架构基础，得到最终的全压缩网络（即图 2 中右侧的网络架构）。

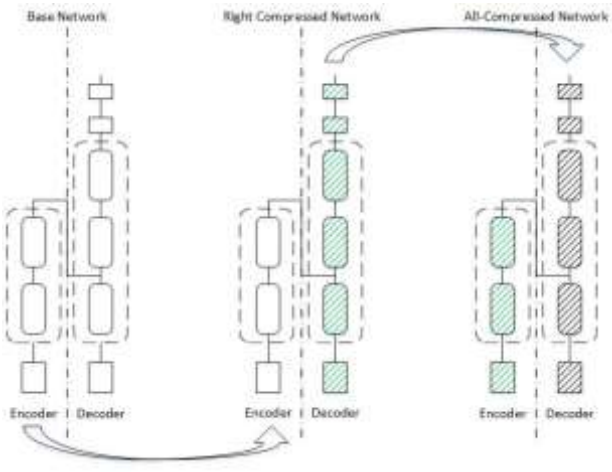

图 2 递进式知识蒸馏概览

对于学生网络，学生模型中未被压缩的参数，其维度和教师模型是同样大小，模型被压缩的权重矩阵变成新的维度大小。递进式压缩两次，交替压缩模型的不同部分的参数（编码器和解码器），从而实现模型完整压缩，即如图 2 中右侧的带阴影填充的全压缩模型。

3.2 结构配置

在不引入新参数的同时保证半知识蒸馏方法完美兼容 Transformer，本节对 Transformer 的结构划分进行说明，从模型压缩的层面来划分，模型可以分为两个部分：压缩和不压缩部分。从 Transformer 结构上来划分，前面提到由于 Transformer 采用残差连接，编码器的所有层的输入输出的维度信息要保持一致，解码器同样如此，故 Tansformer 划分为编码器（左边）和解码器（右边），其中在解码器中对编码器的输出的注意力机制层中的映射权重作为共享部分或称为中间部分（即每次压缩，均需要改变维度，以配合兼容编码器和解码器的对应关系）。如图 1 所示，左侧框内为编码器，右侧框内为解码器，同时将源端词嵌入（Word Embedding）矩阵分到编码器（左边），目标端词嵌入（Word Embedding）分到编码器（右边）。

中间部分是编码器的输出和解码器隐藏单元数不同而能兼容的关键。中间部分由解码器中每一个多头注意力层中处理输入处的线性投影（Linear Projection）的权重$W \in \mathbf{R}^{d \times q}$组成，如式（3）中表示。

3.3 半知识蒸馏方法

半知识蒸馏方法每次压缩模型的一部分权重。学生网络的非压缩部分直接利用教师模型的参数信息进行初始化，作为训练的起点；而学生网络的被压缩权重采用高斯分布初始化。网络参数θ划分成左边半部分参数θ_l和右边半部分参数θ_r，教师模型参数和学生模型参数分别用θ^t和θ^s表示。在递进式压缩方法中，对于先压缩左边半部分参数θ_l和先压缩右边半部分参数θ_r几乎无差别，本文统一默认使用先压缩右半部分参数θ_r再压缩左半部分参数θ_l。压缩参数和非压缩参数统一构成了比前者小的

新模型（即学生模型，参数用θ^s表示），划分后学生网络的目标函数可以用等式（7）表示。

$$\mathcal{L}(\theta_l;\theta_r) = -\sum_{i=1}^{T} logP(y_i|\boldsymbol{Y}_{<i};\mathbf{X};\theta_l^s;\theta_r^s)\ \#(7)$$

其中，T表示当前目标译文的句长；$P(y_i|\boldsymbol{Y}_{<i};\mathbf{X};\theta_l^s;\theta_r^s)$是左半部分参数$\theta_l^s$和右半部分参数$\theta_r^s$在生成第$i$个目标词$y_i$的的条件概率。

右半部分权重参数θ_r^s被压缩，被压缩后的权重维度发生改变，根据大数定律，被压缩的权重维度采用高斯分布进行随机初始化，和标准 Transformer 的初始化方法一致。对于压缩的权重初始化过程表示为等式（8）。

$$\theta_r^s = rand(\theta_r^s)\#(8)$$

其中，θ_r^s是当前学生模型中被压缩的右半部分（解码器）权重;$rand(\theta_r^s)$是一个输出高斯分布的初始化函数，其中均值为 0，方差为$\sqrt{\frac{1}{d_k}}$。

左半部分则是一个知识蒸馏的过程，作为当前的学生模型，从未压缩的教师模型中获得训练所需要的知识，加快模型收敛，如式（9）所示。

$$\theta_l^s = assign(\theta_l^t)\#(9)$$

其中，θ_l^s是学生模型未被压缩的左半部分（编码器）权重，$assign(\cdot)$表示的是赋值操作函数，$assign(\theta_l^t)$即将教师模型参数（θ^t）中的左半部分参数θ_l^t的权重赋值给学生模型参数（θ^s）的左半部分参数θ_l^s的相应权重（即模型中名字相同的权重）。

对源语言和目标语言的词嵌入（Word Embedding）矩阵的处理，将源语言和目标语言的词嵌入矩阵同样划分到模型左右部分中，即每次模型左（右）半部分压缩的同时，将目标语言（源语言）词嵌入矩阵进行压缩。

4. 实验

4.1 实验数据

本文进行充分的实验，在两个广泛使用的中英和日英数据集上进行了比较。这两个翻译数据集是不同规模和不同领域的语料库（范围为 32 万到 125 万个句对），这有助于彻底验证模型对不同大小的训练数据的能力。

对于日英翻译任务，数据集来自公开语料 Kyoto Free Translation Task[1]。使用预处理的训练集作为训练数据，其中处理后的训练数据去掉了句长小于 1 和大于 40 的句子，有关数据处理的更多细节可以在官方网站上看到。处理后包括是 33 万个平行句对用于训练，对语料采用字符编码进行联合字符编码处理[24]，字符编码切分规则大小设为 30k。最后，随机乱序训练集并生成词汇表。

中英翻译任务的语料是用语言数据联盟（Linguistic Data Consortium, LDC）提供的语料，分别选用美国国家标准与技术研究院 2002 年发布的数据，其中训练数据由 1.25M 句子对组成；NIST02、NIST03、NIST04、NIST05 和 NIST08 作为分别作为一个测试集，共 5 个测试集，每个测试集包含 4 个参考译文；NIST06 作为开发集。对英文语料进行切分（tokenized）和中文进行分词处理，将源端句子和目标译文的最大句子长度均限制为 50 个词，并将中英文词汇表的大小限制为 3 万，频次统计前 30 万个高频词作为词表，其余低频词统一替换成<*UNK*>，然后生成词表。

4.2 模型设置

4.2.1 基础设置

将基准（Base）模型的注意力机制的维度、源端和目标端统一设置为 512，将前馈网络层的维度设置为 2048. 为充分验证比较应用方法，本文在表 1 中准备了几个不同大小的不同基准模型，其中不同维度的基准模型分别用不同的标签(Base、R、SR、S)标注，表格中的 ATT 表示注意力机制的维度信息和 FFN 表示前馈神经网络的维度信息。

表 1 不同基准模型的维度信息

系统名	编码器		解码器	
	ATT	FFN	ATT	FFN
基准(Base)	512	2048	512	2048
基准（R）	512	2048	256	1024
基准	256	1024	256	1024

[1] http://www.phontron.com/kftt

基准(SR)	256	1024	128	512
基准（S）	128	512	128	512

在编码器和解码器的各个层中的多头注意力机制均使用了 8 个头。在训练期间，标签平滑（Label Smoothing）参数$\epsilon = 0.1$，神经元的随机失活率（Dropout）设为0.1。将最大训练步数设置为2万，并且以每隔 1500 个训练步保存一次模型。

关于解码参数设置，集束搜索(Beam search)算法并设置束大小（Beam size）K= 4。使用 Adam 优化器[25]其中优化器参数为$\beta_1 = 0.9$，$\beta_1 = 0.98, \varepsilon = 10^{-9}$，学习率使用热启动（warm-up）更新策略[5]，其中$\text{warmup_steps} = 4000$ 的策略。本文在开源的代码 THUMT[2]上进行开发。

4.2.2 训练设置

因为中英和日英语料大小的不同，因此日英和中英的具体的参数设置有所不同。

日英模型将句长接近的句子放到一起，以至于一个批次（Mini-Batch）的数据的句子长度接近，每一个批次大概 6250 个词，训练 2 万步，模型间隔 1500 步保存一次模型，模型的解码器的词嵌入权重矩阵和目标端映射到词表的权重矩阵之间共享参数。

在中英翻译实验中，训练设置，将句子对组合在一起，每批具有大约 6025 个词。将最大训练步数设置为 20 万，并且以间隔 1000 个训练步保存模型。中英翻译模型在解码器的词嵌入权重矩阵和目标端映射到词表的权重矩阵之间均不共享参数。

关于模型评估，在日语到英语的翻译评估中，采用不区分大小写的机器双语互译评估[26](BLEU)得分来评估模型的翻译质量，这是使用 *multi-bleu.perl*[3]。对保存的最后五个模型进行平均，间隔为 1500 个训练更新轮数。

中文到英文的模型评估，采用不区分大小写的 BLEU 评分机制，并使用单一模型进行评估。

4.2.3 参数设置细节

词级别知识蒸馏超参数设置$\alpha = 0.5$，采用基准（Base）模型作为教师模型。

句子级别的知识蒸馏，以基准（Base）模型作为教师模型获取翻译结果，获取翻译结果的参数设置参照 Kim 等人文章[9]中的参数设置，日英翻译任务仅设置束大小（Beam size）为 K=5 进行比较；中英翻译任务运行集束搜索并设置束大小为 K = 5 和 K=1 两种参数配置，分别返回 nbest 个候选翻译译文（其中$\text{nbest} = 5$），最终选择具有更高的 BLEU 分数序列作为新的最终的翻译结果。此外，如果返回 nbest 是空行，则跳过该句子，以此保证句子级别知识蒸馏提供给学生模型的训练语料质量。

4.3 实验结果

日英实验结果，表 2 中是翻译的实验结果，表中，Word-KD 代表词级别的知识蒸馏方法[20]；Seq-KD 代表句子级别的知识蒸馏方法[20]；Our work 代表半知识蒸馏方法；BLEU 差值代表模型翻译结果与模型相应基准模型的差值，其中加号(+)表示提升，空（-）表示基准模型自身；*表示模型与相应基准模型之间显著性检验，$p < 0.05$，检验工具为开源工具[27]。三种不同的压缩方法都取得了显著性优于基准系统的结果，其中，本文提出的方法超过基准模型 2.16 个 BLEU 值，分别超过其他两种知识蒸馏方法，其中超过词级别的知识蒸馏 1.15 个 BLEU 值，超过句子级别的 BLEU 值 0.37 个 BLEU 值。

表 2 日英翻译压缩率与翻译性能

模型	参数	压缩率	BLEU	BLEU 差值
基准(BASE)	61.6m	0	22.54	-
基准（R）	39.1	0	21.66	-
Word+KD[20]	39.1	36.5	22.31	+0.65
Seq+KD[20]	39.1	36.5	22.86	+1.2
Our work	39.1	36.5	23.23	+1.57
基准	19.8	0	21.00	-
Word-KD[20]	19.8	67.9	22.01	+1.01
Seq-KD[20]	19.8	67.9	22.88	+1.88
Our work	19.8	67.9	23.16	+2.16

[2] https://github.com/THUNLP-MT/THUMT

[3] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

中英翻译结果如表 3 所示，其中 Seq-KD(K=1) [20]和 Seq-KD(K=5) [20]分别表示束大小为 1 和 5 的情况下的句子级别的知识蒸馏;+Seq-KD(K=1) [20]和+Seq-KD(K=5) [20]分别表示半知识蒸馏方法使用的训练语料为教师模型在束大小为K=1 和K=5 的情况产生的伪语料。表 3 的结果包含了运用了两次递进式压缩，从 512 维的基准（Base）模型进行递进式压缩得到 256 维的全压缩模型，然后进一步进行递进式压缩，得到 128 维的全压缩模型。根据表 3 的结果显示，本文方法性能优于基准模型，在同样使用教师模型翻译的结果作为训练语料的情况下，性能高于句子级别的知识蒸馏。

表 3 各模型在中英语料上的翻译性能

模型	MT02	MT03	MT04	MT05	MT08	AVG	delta
基准（Base）	42.87	41.24	43.81	41.53	32.03	40.30	-
基准（R）	42.84	41.31	43.96	41.96	31.82	40.38	–
Word-KD[20]	43.71*	41.25	44.25	41.98	32.67*	40.77	+0.43
Seq-KD(K=1) [20]	43.28	41.35	43.75	41.42	31.48	40.26	−0.04
Seq-KD(K=5) [20]	44.29*	42.63*	44.82*	42.90*	32.88*	41.50	+1.20
Our work	43.33*	41.78	44.21	42.61*	32.43	40.87	+0.57
+Seq-KD(K=1)	43.60*	41.68	43.79	41.67	31.80	40.51	+0.21
+Seq-KD(K=5)	44.28*	42.67*	44.82*	42.93*	32.79*	41.50	+1.20
基准	42.88	41.17	43.81	42.10	31.93	40.38	–
Word-KD[20]	43.37	41.62	43.97	41.88	32.03	40.57	+0.19
Seq-KD(K=1) [20]	42.46	41.93	43.3	41.33	31.05	40.01	−0.37
Seq-KD(K=5) [20]	43.18	42.48	44.54	42.06	32.70	40.99	+0.61
Our wrok	43.11	41.63	43.49	42.21	32.20	40.53	+0.15
+Seq-KD(K=1)	42.30	41.38	43.80	41.78	31.80	40.21	−0.17
+Seq-KD(K=5)	43.85	41.75	44.64*	42.56	32.57	41.02*	+0.64
基准（SR）	42.18	39.83	43.35	40.87	31.41	39.53	–
Word-KD[20]	42.82	39.74	42.93	41.26	31.52	39.65	+0.12
Seq-KD(K=1) [20]	42.90	40.89*	43.99	41.77*	31.70	40.25*	+0.72
Seq-KD(K=5) [20]	43.29*	41.91*	44.59*	42.52*	32.94*	41.05*	+1.52
Our work	43.23*	40.37*	43.68	42.16*	31.81	40.25*	+0.72
+Seq-KD(K=1)	43.55*	41.25*	44.08*	41.79*	31.69	40.47*	+0.94
+Seq-KD(K=5)	44.03*	42.79*	45.16*	42.99*	33.14*	41.62*	+2.09
基准（S）	40.88	39.36	42.26	39.29	30.09	38.38	–
Word-KD[20]	41.01	39.03	41.98	39.64	30.66	38.46	+0.08
Seq-KD(K=1) [20]	41.04	39.47	42.12	39.63	29.96	38.44	+0.06
Seq-KD(K=5) [20]	41.91*	40.11	42.93	41.06*	30.87	39.38*	+1.00
Our work	41.72	39.07	42.24	39.21	30.51	38.55	+0.17
+Seq-KD(K=1)	41.30	40.23*	42.46	39.86	29.96	38.76	+0.38
+Seq-KD(K=5)	42.02*	40.52*	43.54*	40.81*	31.09*	39.60*	+1.22

5 分析

5.1 收敛性

图 3 揭示了 BLEU 值从第 0 个训练步稳步上升，可以从图中看出，在 0～5000 步，本文方法（红色曲线）高于其它三个模型基准模型，词级别知识蒸馏和句子级别知识蒸馏的曲线，表明本文方法在前期学习能力明显高于其它模型。本文方法的红色曲线在 10000 步时开始趋于稳定且处于其它三条曲线的上方。这表明本文方法具有早于其它模型收敛的特性。

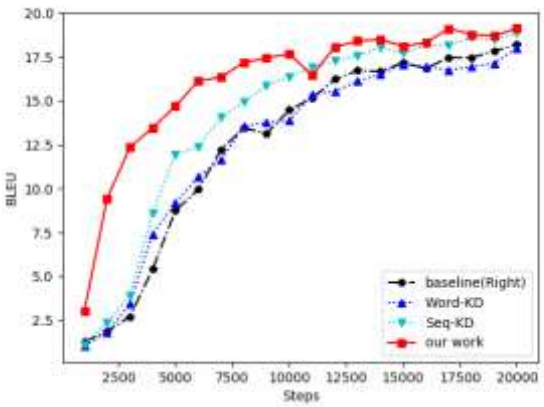

图 3 验证集上 BLEU 值随训练步数变化曲线

5.2 时间和速度

首先，为了公平的衡量词级别和句子级别的时间消耗，表 4 统计了训练时间和数据准备的时间，其中，数据代表新增的数据获取时间，单位：分钟；训练代表训练时长，单位：分钟；GPU 表示 GPU 数目；kw/s 表示千词每秒。默认的数据预处理是相同的，因此未在表中体现，主要以增加的时间作为对比依据。

表 4 模型时间消耗和速度统计表

模型	数据	训练	共计	GPU	kw/s
基准(R)	+0	181	181	1	8.5
Word-KD[20]	+0	301	301	2	9.1
Seq-KD[20]	+248	172	420	1	16.4
Our work	+0	181	181	1	16.0

其中句子级别的知识蒸馏，需要教师模型进行翻译，得到的翻译文本进行挑选 BLEU 最高的的句子作为学生模型的训练语料，此处统计的时间仅为教师模型获取翻译花费的时间，未包括挑选处理的时间。此外，关于 GPU 数量，在实现词级别的时候，需要计算教师模型的输出概率分布，由于实验室设备有限，无法在一个 GPU 上训练，故用了 2 个 GPU，尽管可能对本文提出的方法并不公平，本文的方法总时间消耗明显小于词级别和句子级别的知识蒸馏，在训练阶段的时间消耗，本文方法与句子级别几乎持平，且小于词级别的时间消耗。在训练速度上，本文方法与句子级别的知识蒸馏持平，且高于词级别的训练速度。

结论

在本文中，提出了半知识蒸馏方法，该方法从教师模型中提取权重分布作为知识，通过直接赋值的方式作为学生网络的优化起点，从而影响学生模型的训练，以此加快学生模型的收敛，达到知识蒸馏的效果。与词级别的知识蒸馏相比，本文方法不需要对每次获取教师模型的输出概率进行拟合，减少了计算，与句子级别的知识蒸馏相比，由教师模型残生的句子中学习变为直接从教师模型的权重获取知识。在进行充分实验后，日英翻译实验表明，本文方法比基准模型高 2.16 BLEU。本文方法高于词级别的知识蒸馏 1.15 个 BLEU 值和高于句子级别的知识蒸馏 0.32 个 BLEU 值。中英翻译实验表明，本文方法最佳性能高于基准模型 0.72，优于词级别的知识蒸馏 0.6 BLEU，达到和句子级别知识蒸馏可比的性能。

参考文献

[1] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[C]//Advances in neural information processing systems. 2014: 3104-3112.

[2] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and

translate[J]. arXiv preprint arXiv:1409.0473, 2014.

[3] Luong M T, Pham H, Manning C D. Effective approaches to attention-based neural machine translation[J]. arXiv preprint arXiv:1508.04025, 2015.

[4] Gehring J, Auli M, Grangier D, et al. Convolutional sequence to sequence learning[C]//Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017: 1243-1252.

[5] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in neural information processing systems. 2017: 5998-6008.

[6] LeCun Y, Denker J S, Solla S A. Optimal brain damage[C]//Advances in neural information processing systems. 1990: 598-605.

[7] Han S, Mao H, Dally W J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding[J]. arXiv preprint arXiv:1510.00149, 2015.

[8] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network[J]. arXiv preprint arXiv:1503.02531, 2015.

[9] Srinivas S, Babu R V. Data-free parameter pruning for deep neural networks[J]. arXiv preprint arXiv:1507.06149, 2015.

[10] Mariet Z, Sra S. Diversity networks: Neural network compression using determinantal point processes[J]. arXiv preprint arXiv:1511.05077, 2015.

[11] See A, Luong M T, Manning C D. Compression of neural machine translation models via pruning[J]. arXiv preprint arXiv:1606.09274, 2016.

[12] Murray K, Chiang D. Auto-sizing neural networks: With applications to n-gram language models[J]. arXiv preprint arXiv:1508.05051, 2015.

[13] Denton E L, Zaremba W, Bruna J, et al. Exploiting linear structure within convolutional networks for efficient evaluation[C]//Advances in neural information processing systems. 2014: 1269-1277.

[14] Jaderberg M, Vedaldi A, Zisserman A. Speeding up convolutional neural networks with low rank expansions[J]. arXiv preprint arXiv:1405.3866, 2014.

[15] Lu Z, Sindhwani V, Sainath T N. Learning compact recurrent neural networks[C]//2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2016: 5960-5964.

[16] Lu Z, Sindhwani V, Sainath T N. Learning compact recurrent neural networks[C]//2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2016: 5960-5964.

[17] Chen W, Wilson J, Tyree S, et al. Compressing neural networks with the hashing trick[C]//International Conference on Machine Learning. 2015: 2285-2294.

[18] Han S, Mao H, Dally W J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding[J]. arXiv preprint arXiv:1510.00149, 2015.

[19] Lin Z, Courbariaux M, Memisevic R, et al. Neural networks with few multiplications[J]. arXiv preprint arXiv:1510.03009, 2015.

[20] Kim Y, Rush A M. Sequence-level knowledge distillation[J]. arXiv preprint arXiv:1606.07947, 2016.

[21] 林野，姜雨帆，肖桐，等. 面向神经机器翻译的模型存储压缩方法分析[J]. 中文信息学报，2019，33(1)：93-102.

[22] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[23] Lei Ba J, Kiros J R, Hinton G E. Layer normalization[J]. arXiv preprint arXiv:1607.06450, 2016.

[24] Sennrich R, Haddow B, Birch A. Neural machine translation of rare words with subword units[J]. arXiv preprint arXiv:1508.07909, 2015.

[25] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.

[26] Papineni K, Roukos S, Ward T, et al. BLEU: a method for automatic evaluation of machine translation[C]//Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, 2002: 311-318.

[27] Koehn, P.: Statistical significance tests for machine translation evaluation. In: Pro-ceedings of the 2004 conference on empirical methods in natural language processing (2004)