大规模语言模型的跨云联合训练关键技术

Key Technologies for Cross-Cloud Joint Training of Large-Scale Language Models

潘囿丞/PAN Youcheng，侯永帅/HOU Yongshuai，
杨卿/YANG Qing，余跃/YU Yue，相洋/XIANG Yang
（鹏城实验室，中国 深圳 518055）
(Peng Cheng Laboratory, Shenzhen 518055, China)

DOI：10.12142/ZTETJ.202304010
网络出版地址：http://kns.cnki.net/kcms/detail/34.1228.TN.20230724.1522.002.html
网络出版日期：2023-07-25
收稿日期：2023-06-08

摘要：模型参数规模的不断增加使模型训练所需的算力资源变得更加庞大，导致很多情况下单个算力集群难以满足大规模语言模型的训练需求。大规模语言模型的跨云联合训练成为解决这一问题的有效方式。以自然语言处理大模型的跨云预训练和微调为例，介绍了大规模语言模型跨云训练的主要挑战和关键技术，并探讨了这些技术在跨云训练过程中的具体应用、实际效果和未来场景。这些技术将为智能化应用和人机交互等提供有力支持。

关键词：大规模语言模型；算力资源；跨云训练；自然语言处理

Abstract: As the scale of model parameters continues to grow, the computational resources required for model training become significantly larger. This often leads to situations where a single computing cluster is insufficient to meet the training needs of large-scale language models. Cross-cloud joint training of large-scale language models has emerged as an effective solution to addressing this challenge. In this study, taking cross-cloud pre-training and fine-tuning of natural language processing models as examples, we introduce the main challenges and key technologies involved in cross-cloud training of large-scale language models. The specific applications, practical effects, and future scenarios of these technologies in the cross-cloud training process are explored. These technologies will provide strong support for intelligent applications and human-computer interaction.

Keywords: large-scale language model; computational resource; cross-cloud training; natural language processing

引用格式：潘囿丞, 侯永帅, 杨卿, 等 . 大规模语言模型的跨云联合训练关键技术 [J]. 中兴通讯技术, 2023, 29(4): 49-56. DOI: 10.12142/ZTETJ.202304010

Citation: PAN Y C, HOU Y S, YANG Q, et al. Key technologies for cross-cloud joint training of large-scale language models [J]. ZTE technology journal, 2023, 29(4): 49-56. DOI: 10.12142/ZTETJ.202304010

大规模语言模型是一种使用深度学习方法技术在大规模无标注文本语料数据上进行训练的人工智能方法。近年来，这类模型得到了快速发展，模型能力实现极大提升。然而，模型的参数规模也变得越来越大。例如，2018年谷歌的BERT-Base模型只有1.1亿个参数[1]，而到了2020年，OpenAI的GPT-3模型的参数量已经达到1 750亿个[2]。随着模型参数的增加，模型训练所需的算力资源也变得更加庞大。BERT-Base模型可以在单张图形处理器（GPU）上训练，而GPT-3模型则需要在数千张GPU上进行数月的训练。当前，单个算力集群很少具备数千张GPU算力卡的规模，即使是那些具有数千张卡的算力集群，也很难将它们在长时间内集中用于同一个任务。因此，为了满足大规模语言模型的训练需求，需要将多个算力集群的资源联合训练来提高效率。随着“东数西算”工程的逐步开展，中国各地建立了大量的算力集群。异地跨云计算将成为今后大模型训练的可行方式。

基金项目：科技创新2030—“新一代人工智能”重大项目（2022ZD0115301）

1 基于多算力集群的跨云训练方法

1.1 跨云计算的并行训练方式

在跨云集群环境中进行模型训练，需要解决不同云集群

之间参数的传递和同步问题，以及由大量数据跨云传输的时间开销导致模型训练速度慢的问题。为了提升训练速度，训练任务被拆分到多个不同的算力集群上。利用这些集群的算力，可以实现对任务的并行处理。根据不同的任务需求和场景，跨云训练可以采用不同的并行策略，包括数据并行、模型并行和流水线并行等。

数据并行是提升训练速度的一种并行策略，能够将训练任务切分到多个算力集群上。每个集群维护相同的模型参数和计算任务，只是处理不同的批数据。通过这种方式，全局的数据被分配到不同的进程，从而减轻单个集群上的计算和存储压力。

模型并行主要用于模型太大、无法在单个设备上加载的场景，对计算图按层切分以减少单个存储的容量需求，每个集群只保留模型的一部分。因此，多个算力集群可以共同训练一个更大的模型。

当模型并行在某个集群进行计算时，其余集群都会处于闲置状态，这样会极大地降低整体的使用效率。于是，在模型并行的基础上，如图1所示，把原先的批数据再划分成若干个微批次，按流水线方式送入各个算力集群进行训练，也就是流水线并行[3]。

当在跨云场景下进行大规模语言模型训练时，由于巨大的数据量和参数规模，不论是对训练数据还是模型张量进行切分，在进行跨云同步传输时都会产生较大的耗时，会影响整体的训练速度。由此可见，数据并行和模型并行这两种方式能够支持的模型参数规模有限。而流水线并行训练则将模型参数按照层次进行拆分，把不同层的模型参数放到不同集群中进行训练。训练过程中不需要同步全部模型参数，集群之间只需要串行传递训练过程的中间计算变量。该方法受模型参数规模影响较小，更适合大规模语言模型的跨云训练。

1.2 跨云流水线并行的主要挑战及关键技术

跨云流水线并行和普通流水线并行的最大区别在于处理通信数据的方式。目前，普通流水线并行策略通常仅在单个计算资源中心内部使用，这意味着计算设备之间存在专用的高带宽网络连接。此时，通信代价极低，通常可以忽略不计。然而，当普通流水线并行策略应用于跨云场景时，计算设备之间的连接带宽远低于上述连接，通信代价将显著增加，这将极大地影响训练效率。图1的左图和右图分别展示了普通流水线并行和跨云流水线并行的处理流程。

普通流水线并行的效率评价指标为并行空泡占用率比例（parallelism bubble ration），该比例越小代表效率越高。假设并行的阶段（stage）数为 p，微批次的数量（micro-batch）为 m，每个micro-batch的前向和后向执行时间为 t_f 和 t_b，则空泡率为：

$$bubbleration = \frac{p-1}{m+p-1}。\tag{1}$$

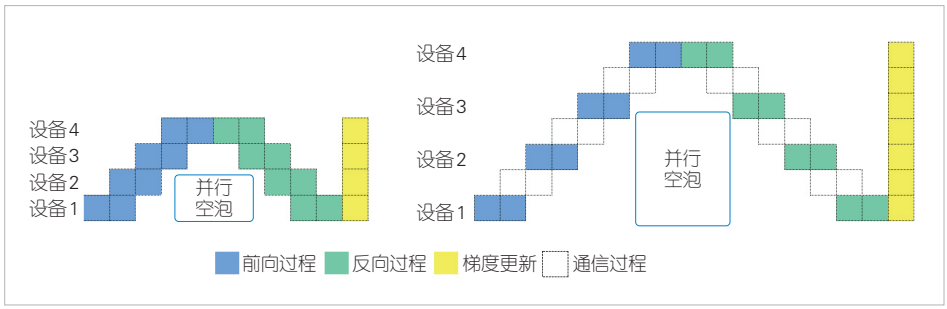
而在跨云流水线并行中，会出现因为通信而导致的额外空泡。假设通信时间为 t_t，在不做任何处理的情况下，前向和后向的通信时间相等，此时空泡率为：

$$bubbleration = \frac{(p-1)(t_f+t_b+2mt_t)}{m(t_f+t_b)+(p-1)(t_f+t_b+2mt_t)}。\tag{2}$$

因此，跨云流水线并行所面临的主要挑战是如何提高训练效率，即如何降低并行空泡的占用率。从上述公式（2）中可以看出，在跨云场景中，与普通流水线并行不同，增加微批次的数量并不一定会提高效率，需要根据实际情况进行分析，并计算出最优的微批次数量。此外，公式（2）还表明，缩短通信时间、减少阶段数量均有助于降低空泡率。特别是由于通信时间的存在，阶段数量对空泡率的影响更为显著。因此，减少阶段数量可以带来更大的收益。下面我们将从这两个方面介绍相关的技术。

缩短通信时间的核心在于减少通信的数据量。为此，可以采用稀疏化、量化和低秩训练等技术。另外，阶段数量主要受到节点总内存的限制。如果能够降低训练占用的内存，就可以使每个节点容纳更多的参数，从而有可能降低阶段数。需要注意的是，在此处，以增加通信量为代价来降低内存的方案并不适用。

稀疏化的主要思想是，神经网络层的输出中绝对值较大的数值通常承载了更多的信息量。因此，将中间层数据中的大多数数值变为0就不会损失主要信息。对此可以利用稀疏化数据的表示方式来压缩数据，从而减少通信量和存储空间



▲图1 普通流水线并行和跨云流水线并行

的占用。

量化则是将传输的中间结果从原本32位比特的浮点数映射到8位或者更少比特表示的整型数据上。这种方式可以有效压缩通信数据，但是会带来额外的误差，进而会影响到训练的精度。因此，需要根据实际的数据分布情况来设计量化的位数和方式。

大型模型通常存在“过参数化”的问题，即虽然模型的参数众多，但实际上模型主要依赖于低秩维度的内容。为此，可以采用一些基于低秩分解的训练方法，例如低秩适应（LoRA）[4]算法。该方法新增了一个先降维再升维的旁路。这样的设计可以天然地降低中间数据的维度。将降维矩阵的输出位置作为切分点也可以达到减少通信时间的目的。

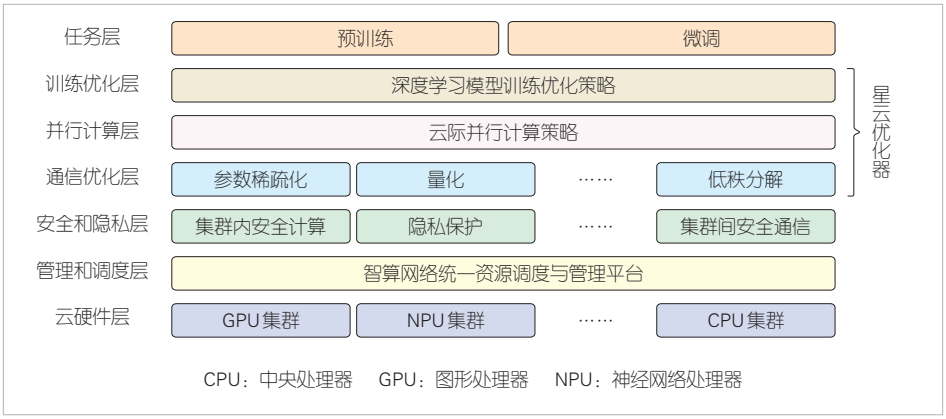
2 一种面向大规模语言模型的跨云训练方法

大规模语言模型的训练任务包括语言模型预训练和下游任务微调两个阶段。为了应对跨云模型训练的挑战，本文中我们将介绍一种基于跨云大模型训练框架“星云”[5]的预训练和微调方法。如图2所示，“星云”是一个专门面向云际环境的深度学习模型统一训练框架，该框架包含了任务层、训练优化层、并行计算层、通信优化层、安全和隐私层、管理和调度层以及云硬件层等7个功能层，支持在低带宽网络环境下，利用不同算力集群的异构算力进行大模型的跨云训练，在通信优化方面采用了参数稀疏化、量化以及低秩分解等有效技术来确保集群间信息传输的轻量化和最小化模型精度损失，并主要采取流水线并行的方式来实现在多个算力集群间的并行计算。

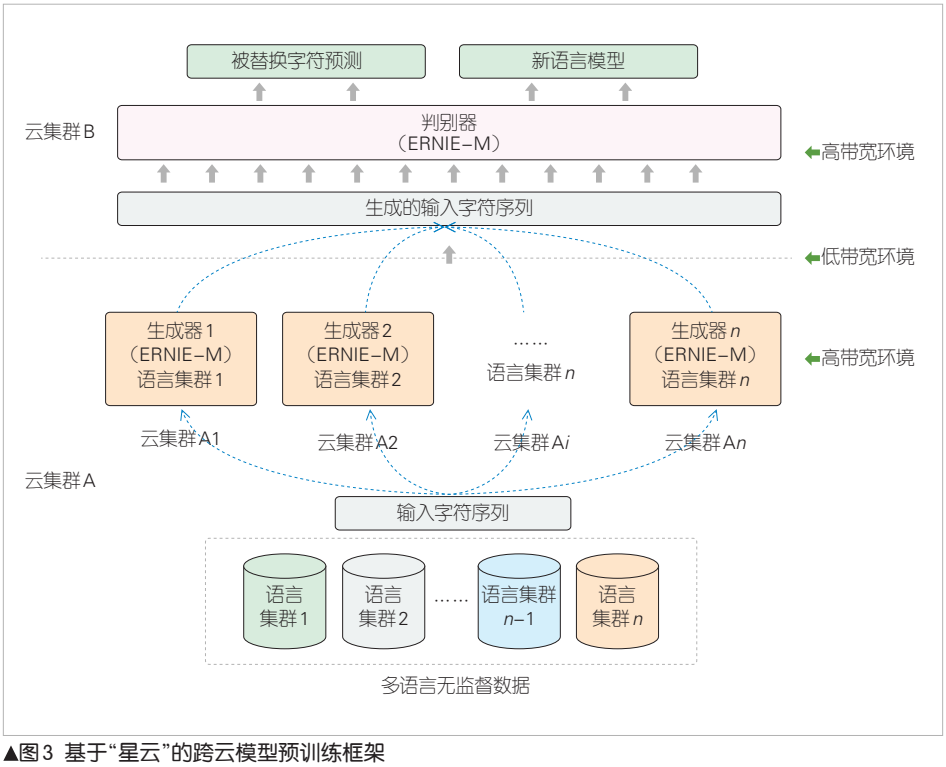
2.1 多语言大模型的跨云预训练方法

针对多语言模型预训练任务，我们基于“星云”实现了一套支持跨云多源数据训练的多语言模型预训练方案，如图3所示。为了优化训练过程，该方案参考ELECTRA[6]架构设计了一种适合跨云使用的模型架构，由生成器（Generator）和判别器（Discriminator）两部分组成。其中，生成器根据输入内容生成对应的字符序列，判别器则对生成的字符序列进行判断，以达到优化训练的目的。

在模型训练过程中，生成器只需要将输出的字符序列单向传递给判别器。当进行跨云训练时，生成器和判别器会被部署在不同的云集群上，此时生成器只需向判别器传输字符串序列即可。在这个过程中，所需的数据传输量较少，带宽需求也较低，这有利于跨云大模型的训练。此外，通过共享生成器和判别器间的词表、跨云只传输字符ID序列的方式



▲图2 “星云”的框架结构示意图



▲图3 基于“星云”的跨云模型预训练框架

不仅可以进一步减少数据传输量，还可以避免数据泄露。

为了支持多源数据多方协同训练，该架构需要使用多个生成器来共同训练判别器。不同的生成器对应不同的训练数据和不同的预训练模型，例如：可以让每个生成器负责一个语种的生成，多个生成器共同支持多语言判别器的训练，这样可以提高训练效率，增强判别器的泛化能力。

在模型训练过程中，生成器和判别器之间只有单向的字符标识序列传输，数据量小，受网络带宽瓶颈影响较小。为了提高集群资源的利用率和训练速度，本文中我们采用了数据并行的方式在生成器集群和判别器集群内部分别进行训练。为了验证该框架在异构算力环境下的模型训练能力，我们将生成器部署在GPU算力集群，将判别器部署在NPU算力集群。该框架的跨云集群部署及并行计算方式如图4所示。这种部署和计算方式可以提高训练效率，优化资源利用率。

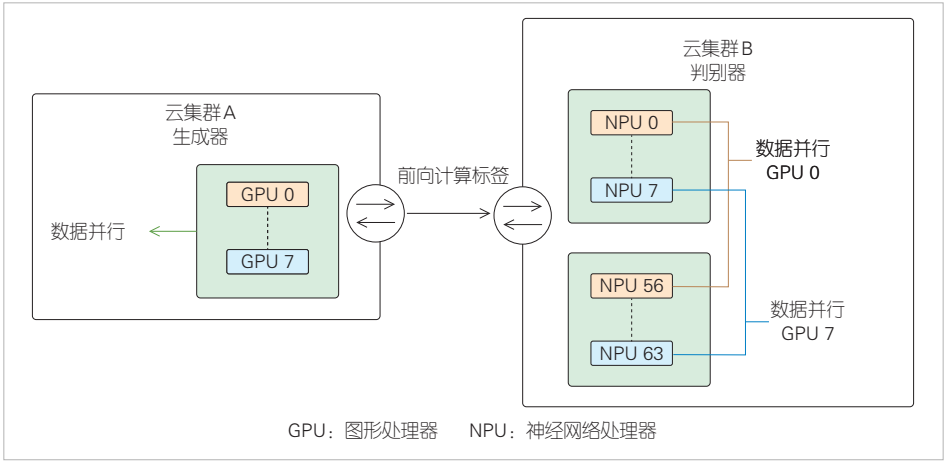
为了测试跨云模型预训练的效果，实验中我们利用包含116种语言的单语数据和15种语言的平行语料数据，进行基于生成器–判别器架构的跨云大模型训练。使用多语言预训练语言模型ERNIE–M–Base来初始化生成器，使用ERNIE–M–Large来初始化判别器，训练得到的判别器ERNIE–M–Extra则作为最终的多语言大模型。为了测试ERNIE–M–Extra模型的多语言能力，本文中我们首先使用英语数据进行微调，然后在15种语言的跨语言推理任务上进行了测试。测试结果如表1所示。

由表1可知，ERNIE–M–Extra模型在15种语言的跨语言推理任务中表现出最优的平均成绩，相比于基础模型ERNIE–M–Large，其精度提高了0.2。

为了测试模型训练过程的吞吐率，我们进行了在云集群内和跨云集群环境下的测试。实验结果显示，跨云训练的吞吐率达到了单云集群训练的85%。在GPU算力集群和NPU算力集群环境下，针对异构环境下硬件加速效果进行了实验，并对比了由8卡NPU算力增加到64卡的模型训练速度。实验结果表明，增加算力卡后训练速度提高了4.34倍。

为了验证模型在跨云集群训练中的有效性，本文对比了单云环境和跨云环境下模型训练的损失曲线，如图5所示。可以看出，跨云集群训练可以保持训练过程的持续收敛。

综上所述，采用生成器–判别器架构进行多语言大模型训练，可以在跨云环境下保持较高的吞吐率，确保训练过程持续收敛。此外，增加算力资源可以有效提高训练速度。



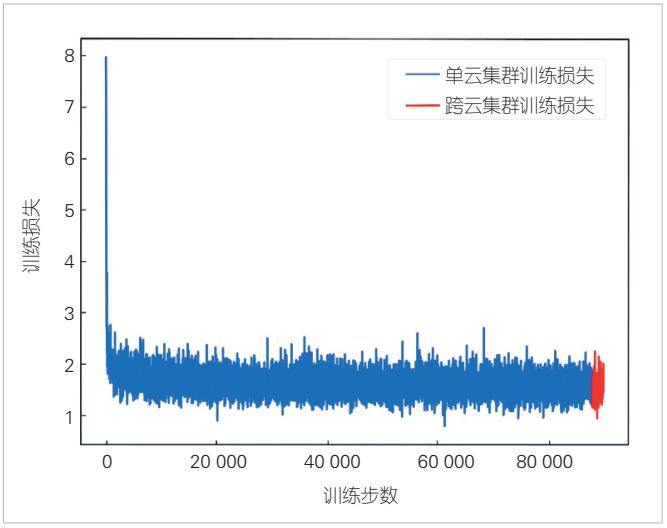
▲图4 跨云预训练集群算力互联及并行计算方式

2.2 大规模语言模型的跨云微调方法

微调是指在预训练大模型的基础上，为了特定的任务进行有针对性的模型训练。本文中我们将分别介绍基于编码器–解码器架构的自然

▼表1 跨云模型预训练最终模型精度对比

模型	En	Fr	Es	De	El	Bg	Ru	Tr	Ar	Vi	Th	Zh	Hi	Sw	Ur	平均
XLM[7]	85.0	78.7	78.9	77.8	76.6	77.4	75.3	72.5	73.1	76.1	73.2	76.5	69.6	68.4	67.3	75.1
Unicoder[8]	85.1	79.0	79.4	77.8	77.2	77.2	76.3	72.8	73.5	76.4	73.6	76.2	69.4	69.7	66.7	75.4
XLM–R[9]	85.8	79.7	80.7	78.7	77.5	79.6	78.1	74.2	73.8	76.5	74.6	76.7	72.4	66.5	68.3	76.2
INFOXLM[10]	86.4	80.6	80.8	78.9	77.8	78.9	77.6	75.6	74.0	77.0	73.7	76.7	72.0	66.4	67.1	76.2
ERNIE–M[11]	85.5	80.1	81.2	79.2	79.1	80.4	78.1	76.8	76.3	78.3	75.8	77.4	72.9	69.5	68.8	77.3
XLM–RLARGE[9]	89.1	84.1	85.1	83.9	82.9	84.0	81.2	79.6	79.8	80.8	78.1	80.2	76.9	73.9	73.8	80.9
INFOXLMLARGE[10]	89.7	84.5	85.5	84.1	83.4	84.2	81.3	80.9	80.4	80.8	78.9	80.9	77.9	74.8	73.7	81.4
VECOLARGE[12]	88.2	79.2	83.1	82.9	81.2	84.2	82.8	76.2	80.3	74.3	77.0	78.4	71.3	80.4	79.1	79.9
ERNIE–MLARGE[11]	89.3	85.1	85.7	84.4	83.7	84.5	82.0	81.2	81.2	81.9	79.2	81.0	78.6	76.2	75.4	82.0
ERNIE–M–Extra	89.4	85.1	86.0	84.5	84.4	84.6	81.8	81.7	81.8	81.9	79.3	81.2	79.1	76.3	75.7	82.2



▲图5 单云训练和跨云训练损失对比

语言生成微调训练和基于编码器架构的自然语言理解微调训练。

2.2.1 针对自然语言生成任务的微调

针对基于编码器–解码器架构的自然语言生成模型，本文以机器翻译任务为例，参照ABNet[13]模型架构设计，实现基于“星云”的跨云机器翻译模型微调训练。ABNet是一种用于微调训练的模型架构，在编码器和解码器的各个子层之间插入需要训练的适配器模块。在训练过程中，预训练模型的参数被冻结。该微调方法利用预训练语言模型的知识，但不调整预训练模型的参数。如图6所示，针对源语言和目标语言的预训练模型分别被部署在两个云集群中。

在模型训练时，每进行一步前向计算和反向传播，编码端和解码端都需要进行一次跨云中间数据传输。数据传输量与数据批处理大小（B）、序列长度（S）、隐藏层维度（H）等因素相关。需要传递的数据规模如公式（3）所示：

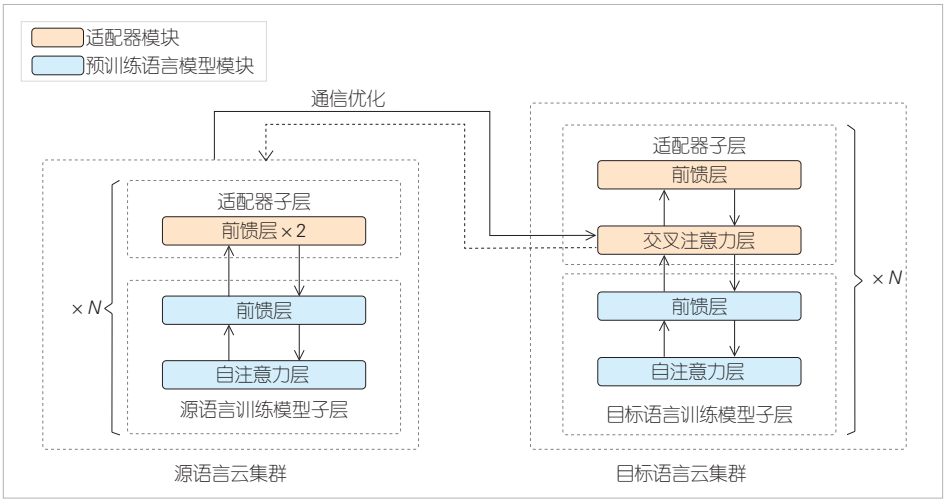
$$Data\ size = B \times S \times H。 \quad (3)$$

在微调训练过程中，数据传输占用了大量的网络带宽资源。传输时间的长短对训练速度的影响很大。当网络带宽过低时，跨云训练就无法达到加速训练的目的。因此，为了提高模型的训练速度，“星云”框架从云间通信和并行训练两个方面进行综合优化。

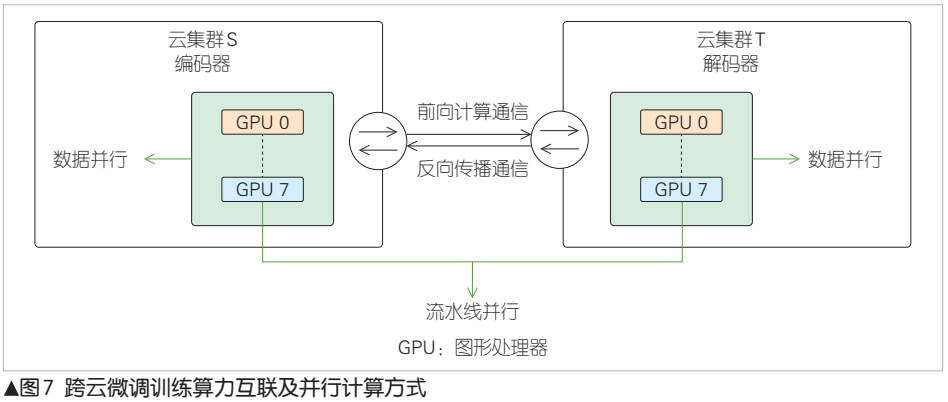
为了解决在训练过程中数据传输量大、传输时间长的问题，针对需要跨云传输的中间数据，可以采用压缩通信的策略进行优化，以减少单次传输的数据量。可采用的压缩通信方法主要包括量化、稀疏化、低秩分解等。为了减小压缩通信对模型精度的影响，可以组合使用不同的压缩策略，并在训练的不同阶段采用不同的压缩传输策略。

为了解决在模型训练过程中由串行计算导致的资源利用率不高的问题，“星云”采用并行优化策略来优化训练过程。在云集群间采用流水线并行，云集群内采用数据并行的方式，采用多微批次以流水线并行的方式在云集群间执行计算和数据传输任务，可以减少同一时刻资源的停等，提高参与训练各资源的利用率。ABNet架构在跨云环境的部署及并行计算方式如图7所示。

为了进行跨云集群模型微调的实验，我们选择IWSLT’



▲图6 基于ABNet的跨云微调训练方法



▲图7 跨云微调训练算力互联及并行计算方式

14的西班牙语（Es）到英语（En）的机器翻译任务，并采用ABNet跨云架构基于预训练语言模型进行微调训练。在该实验中，我们使用多语言预训练模型ERNIE-M-base-cased作为编码端，使用英文预训练模型BERT-Base作为解码端，并将它们分别部署在两个配备了8张NVIDIA V100 GPU显卡的云集群上。

实验结果显示，完全重新训练的Transformer-Base模型[14]的双语评估替换（BLEU）值[15]为39.60，在本地微调训练的ABNet-Local模型为43.19，采用跨云微调训练的ABNet-Cloud模型为41.92。实验结果表明，采用基于预训练模型微调的翻译模型性能优于仅使用训练数据重新训练的Transformer-Base模型。相对于仅在本地集群训练的ABNet-Local模型，跨云微调的ABNet-Cloud模型的BLEU值降低了1.27个，这是由于压缩通信导致了模型精度损失。然而，相对于Transformer-Base模型，ABNet-Cloud仍然提高了2.32个BLEU值。这表明在跨云环境中，基于预训练语言模型进行微调训练可以复用预训练模型的知识，从而提高最终翻译模型的精度。

为了研究压缩通信策略对模型训练的影响，我们对不同压缩通信策略下的模型训练速度和最终模型精度进行了对比。其中，前向计算数据传输采用FP16半精度及其与不同压缩率的SVD分解的组合，反向传播采用固定的INT8量化压缩。实验结果如表2所示，压缩率越高，模型训练速度越快。在FP16(SVD(0.2))+INT8的压缩策略下，模型训练单步消耗时间仅为不压缩训练的19%。然而，该策略下模型精度损失了4.19个BLEU值。在所验证的压缩策略中，FP16(SVD(0.6))+INT8策略下得到的模型精度最佳（达到41.92），单步训练时间仅为不压缩的32%，训练速度提升了3倍以上。

▼表2 不同压缩通信方法性能对比

压缩方法	BLEU	训练速度(s/步)
ABNet-Local	43.19	4.42
FP16+INT8	38.82	1.60
FP16(SVD(0.8))+INT8	41.15	1.50
FP16(SVD(0.6))+INT8	41.92	1.42
FP16(SVD(0.4))+INT8	39.56	0.94
FP16(SVD(0.2))+INT8	39.00	0.86

BLEU：双语评估替换　SVD：奇异值分解
FP16：半精度　INT8：8比特量化

2.2.2 针对自然语言理解任务的微调

自然语言理解包括文本分类、文本蕴含、阅读理解等任务。通常人们采用基于编码器类型的预训练模型进行微调训练。为了在跨云环境下微调这类模型，可以采用低秩结构的思想对通信数据进行压缩[16]。具体的做法如下：

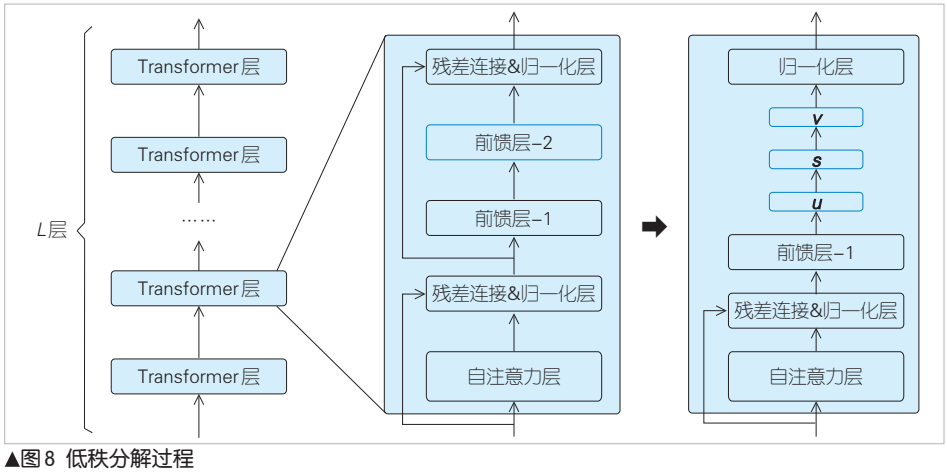
1）对于模型中的每一个Transformer块，假设其输入和输出矩阵的维度为$\boldsymbol{R}^{b\times d}$，即在跨云训练时，通信数据的维度也为$\boldsymbol{R}^{b\times d}$。其中，$b$表示batch_size，$d$表示模型的维度参数。

2）对于其中一个Transformer块的线性层，可以进行奇异值分解来降低通信数据的维度。具体做法是：将该线性层的权重矩阵$\boldsymbol{W}\in\boldsymbol{R}^{m\times d}$进行奇异值分解，选取前$r$个奇异值，得到3个矩阵$\boldsymbol{u}$、$\boldsymbol{s}$和$\boldsymbol{v}$，维度分别为$\boldsymbol{R}^{m\times r}$、$\boldsymbol{R}^{r\times r}$和$\boldsymbol{R}^{r\times d}$；然后，使用3个连续的线性层来替代原始的线性层，这3个线性层的权重分别为U、S和V，如图8所示。

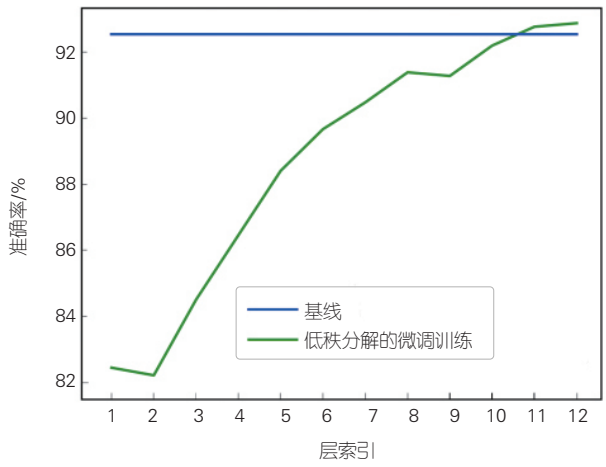
3）将用于跨云通信的模型拆分点设置在S和V层之间，并移除该Transformer块的直接连接分支。这样，通信数据的维度会变成$\boldsymbol{R}^{b\times r}$，即原有数据的$r/d$倍。

根据上述的压缩方案，以BERT-Base为基础模型，在GLUE数据集[17]和SQuAD数据集[18]上进行跨云微调训练，并分析该算法在不同层索引上对训练精度的影响。将上述算法中的r设置为8，实验结果如图9所示。其中，横轴表示拆分的层级索引，纵轴表示准确率。需要说明的是，由于各个数据集表现出的规律一致，这里仅以SST-2和QNLI数据集为代表。

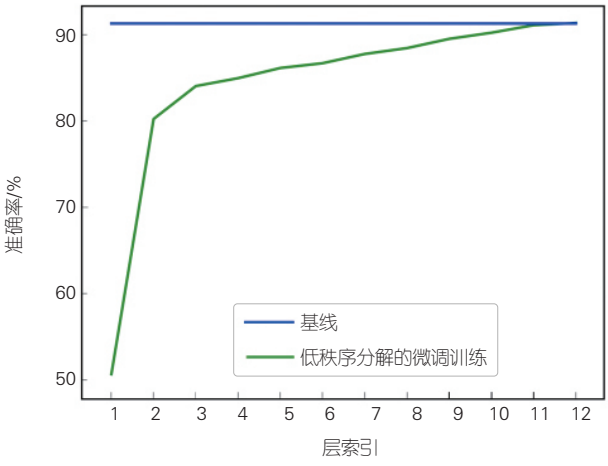
由图9可知，r值较小且拆分位置处于模型的底层会导致训练精度显著下降。但是，当拆分位置位于模型的高层时，r值的大小对训练精度没有影响。在实验中，我们选择



▲图8 低秩分解过程



（a）SST-2



（b）QNLI

注：QNLI、SST-2是两种数据集

▲图9 基于低秩分解的跨云微调

将模型拆分在第11层，然后针对不同的r值（分别为8、16和32）进行测试，结果如表3所示。特别地，在r等于8的情况下，传输数据量降为原有的1/96，同时精度维持在原有模型的相当水平。

通过跨云场景的模型微调训练实验验证，我们证实了跨云微调的可行性。用户可以利用分布在不同云集群上的预训练模型来微调目标任务模型，并通过复用已有模型的知识来提升模型性能。这比仅使用自身数据训练模型更为优越。由于模型被拆分成多个部分，用户可以将模型的底层部分置于可信集群上，从而确保其他集群无法获得标注数据，保障用户标注数据的安全性。

3 跨云训练算力互联及未来场景

生成算法、预训练模型、多模态等技术的融合催生了以ChatGPT为代表的人工智能生成内容（AIGC）的爆发，进而带来了高算力需求。以ChatGPT为例，它使用了10 000块A100 GPU进行训练。此外，它的部署成本也很高，根据国盛证券报告估算，它的每日咨询量对应的算力需求达到了上万块A100。所以，利用跨云训练可以将广泛分布的算力结合起来，这是应对大模型对算力高需求的一种解决方案，从而有效应对算力对大模型训练的制约。同时，跨云训练可以利用闲散算力，有效解决碎片化问题，提高云集群资源的利用率。

除了算力限制，与个人信息强相关的应用，例如语音助手、心理咨询等，也关注隐私保护问题。跨云训练机制具备较好的隐私保护能力。用户可以通过构建本地设备与云的协同训练来实现个人信息在本地处理、云端提供算力的方式，从而保证个人信息不被泄露。

4 结束语

本文的研究表明，在跨云环境下进行大规模语言模型训练是可行的，是一种提高算力利用率的方案。通过采用模型分割、拆分学习、跨云协同、压缩通信和模型复用等关键技术，该方案能够有效解决跨云训练过程中可能出现的算力和数据不足的问题，并提高训练速度和效率。这些技术在自然语言处理领域的应用将有望带来更为精准和高效的文本处理和语义分析结果，并具备较好的隐私保护能力，为智能化应用和人机交互等领域的发展提供有力的支持。

致谢

感谢百度飞桨团队吴志华和巩伟宝，以及哈尔滨工业大

▼表3 11层拆分微调结果（k表示1 000）

	SST-2（67k）	QNLI（105k）	MNLI（364k）	QQP（91.2k）	CoLA（8.5k）	RTE（2.5k）	STS-B（7k）	MRPC（3.7k）	SQuAD（88k）
基线模型	92.54	91.24	84.56	90.73	55.3	66.06	88.38	85.33	88.25
r=8	92.43	90.98	83.98	90.93	57.13	64.25	86.46	84.81	88.33
r=16	92.31	91.22	84.33	90.75	57.35	62.09	86.78	83.47	88.75
r=32	92.77	91.04	84.27	90.99	57.87	62.81	87.46	84.23	88.56

学(深圳)施少怀教授对本文写作提供的帮助!

参考文献

[1] DEVLIN J, CHANG M W, LEE K, et al. BERT: pre-training of deep bidirectional transformers for language understanding [EB/OL]. [2023-06-08]. https://arxiv.org/abs/1810.04805
[2] BROWN T B, MANN B, RYDER N, et al. Language models are few-shot learners [EB/OL]. [2023-06-08]. https://arxiv.org/abs/2005.14165
[3] HUANG Y P, CHENG Y L, CHEN D H, et al. GPipe: efficient training of giant neural networks using pipeline parallelism [EB/OL]. [2023-06-08]. https://arxiv.org/abs/1811.06965
[4] HU E J, SHEN Y L, WALLIS P, et al. LoRA: low-rank adaptation of large language models [EB/OL]. [2023-06-08]. https://arxiv.org/abs/2106.09685
[5] XIANG Y, WU Z H, GONG W B, et al. Nebula-I: a general framework for collaboratively training deep learning models on low-bandwidth cloud clusters [EB/OL]. [2023-06-08]. https://arxiv.org/abs/2205.09470
[6] CLARK K, LUONG M T, LE Q V, et al. ELECTRA: pre-training text encoders as discriminators rather than generators [EB/OL]. [2023-06-08]. https://arxiv.org/abs/2003.10555
[7] LAMPLE G, CONNEAU A. Cross-lingual language model pretraining [EB/OL]. [2023-06-08]. https://arxiv.org/abs/1901.07291
[8] HUANG H Y, LIANG Y B, DUAN N, et al. Unicoder: a universal language encoder by pre-training with multiple cross-lingual tasks [EB/OL]. [2023-06-08]. https://arxiv.org/abs/1909.00964
[9] CONNEAU A, KHANDELWAL K, GOYAL N, et al. Unsupervised cross-lingual representation learning at scale [EB/OL]. [2023-06-08]. https://arxiv.org/abs/1911.02116
[10] CHI Z W, DONG L, WEI F R, et al. InfoXLM: an information-theoretic framework for cross-lingual language model pre-training [EB/OL]. [2023-06-08]. https://arxiv.org/abs/2007.07834
[11] OUYANG X, WANG S H, PANG C, et al. ERNIE-M: enhanced multilingual representation by aligning cross-lingual semantics with monolingual corpora [EB/OL]. [2023-06-08]. https://arxiv.org/abs/2012.15674
[12] LUO F L, WANG W, LIU J H, et al. VECO: variable and flexible cross-lingual pre-training for language understanding and generation [EB/OL]. [2023-06-08]. https://arxiv.org/abs/2010.16046
[13] GUO J L, ZHANG Z R, XU L L, et al. Incorporating BERT into parallel sequence decoding with adapters [C]//Proceedings of the 34th International Conference on Neural Information Processing Systems. ACM, 2020: 10843 - 10854. DOI: 10.5555/3495724.3496634
[14] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all You need [C]// Proceedings of the 31st International Conference on Neural Information Processing Systems. ACM, 2017: 6000 - 6010. DOI: 10.5555/3295222.3295349
[15] 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 - ACL '02. Association for Computational Linguistics, 2001: 311-318. DOI: 10.3115/1073083.1073135
[16] SHI S H, YANG Q, XIANG Y, et al. An efficient split fine-tuning framework for edge and cloud collaborative learning [EB/OL]. [2023-06-08]. https://arxiv.org/abs/2211.16703
[17] WANG A, SINGH A, MICHAEL J, et al. GLUE: a multi-task benchmark and analysis platform for natural language understanding [C]//Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. Association for Computational Linguistics, 2018: 353 - 355. DOI: 10.18653/v1/w18-5446
[18] RAJPURKAR P, ZHANG J, LOPYREV K, et al. SQuAD: 100, 000+ questions for machine comprehension of text [C]//Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2016: 2383-2392. DOI: 10.18653/v1/d16-1264

作 者 简 介

潘囿丞，鹏城实验室网络智能部云计算所在站博士后；主要研究方向为自然语言处理、文本生成、机器翻译等；发表论文8篇，获授权国家发明专利1项。

侯永帅，鹏城实验室网络智能部云计算所工程师；主要从事智能问答、机器翻译、云际协同学习等方面的研究；发表论文15篇，获授权国家发明专利2项。

杨卿，鹏城实验室网络智能部云计算所工程师；主要从事自然语言处理、协同计算等方面的研究。

余跃，鹏城实验室人工智能开源技术总师、AITISA联盟智算中心和智算网络标准工作组联合组长、算力网络推进组组长；主要从事智能计算、云计算、开源软件等相关领域的研究工作；作为技术负责人负责AITISA联盟智能计算中心与算力网相关标准体系的制定与开源平台研发；发表论文50余篇。

相洋，鹏城实验室网络智能部云计算所副所长、深圳“孔雀计划”海外高层次人才；主要研究方向为自然语言处理、人工智能、大模型、云计算等；主持两项国家级科研项目，参与多项重大科研攻关项目；获深圳市科技进步二等奖1项；发表论文80余篇。