St-Moe：设计稳定和转移的专家模型

Barret Zoph *
Google大脑

Irwan Bello*†
Google Brain

Sameer Kumar
Google

Nan Du
Google Brain

Yanping Huang
Google Brain

Jeff Dean
Google Research

Noam Shazeer†
Google Brain

William Fedus*
Google Brain

ABSTRACT

量表已经在自然语言处理方面开放了新的边界 - 但价格很高。在响应中，Experts（MOE）和开关变压器的混合物已被实现为能量有效的途径，直达更大且功能更强大的语言模型。但是在培训期间训练不稳定性和不确定的质量阻碍的一系列自然语言任务中，推进最先进的工作。我们的工作重点是这些问题，并作为设计指南。我们通过将稀疏模型到269b参数结束，并使用计算成本complacaleto A 32B密度编码器 - 码头变压器（稳定且可转移的Experts或ST-MOE-32B）。第一次，稀疏模型在转移学习中实现了最先进的表现，包括包括推理（Superglue，Arc Easy，ARC挑战），摘要（XSUM，CNN-DM），封闭的书籍问题回答（WebQA，自然问题），并构造了对抗的任务（Winogrande，Anli R3）。 1

*相等的贡献。与{Barretzoph，liamfedus}@google.com的通信。†工作是在Google.1 for我们的型号的情况下完成的，请访问https://github.com/tensorflow/mesh/mesh/master/master/master/mesh_tensorflow/moe/transformer/moe。 py r/

CONTENTS

1 简介

稀疏的专家神经网络展示了纯粹的量表的优势，并提供了有效的替代品静态神经网络体系结构（Raffel等，2019; Brown等，2020; Rae等，2021）。稀疏的Expertnetworks并没有将相同的参数应用于所有输入，而是动态选择用于每个输入的哪些参数（Shazeer等，2017）。这使网络大大扩展其参数数量，同时使flops大致稳定。这些方法产生了最先进的翻译模型（Lepikhinet Al。，2020），4-7倍预训练的速度（Fedus等，2021; Artetxe等，2021）和GPT-3使用1/3的能量训练成本（Du等，2021）。尽管参数令人震惊，但稀疏模型却减少了培训大神经网络的碳足迹，以占据数量级（Patterson等，2021）。但是，困难仍然存在。

Fedus等。（2021）观察到，稀疏的1.6T参数模型获得了4倍的预训练速度，以前的最新训练（Raffel等人，2019年），但是当精确调节的Oncommon基准（如Supersglue）时，落后较小的模型。在Artetxe等人中观察到了类似的差距。（2021）当Moe语言模型在室外数据上进行了细胞调整。作为响应，提出并提高了自然语言理解任务的8倍计算足迹（大约等于th th的T5模型）的Switch-XXL，但较少的参数，但8倍大量的计算足迹（FLOPS大约等于Th thrage T5模型）。不稳定性以前未被发现的持续时间量表研究。后来在其他稀疏模型中确定了这些不稳定性（Du等，2021）。这些结果揭示了参数和计算的必要平衡，但留下了一个开放的问题，如何可靠地训练这些类型的模型。

我们在本文中的目的是提高稀疏模型的实用性和可靠性。我们研究这些问题并预先培训269B稀疏模型，该模型在包括SuperGlue在内的许多竞争性NLP基准（包括Superglue）时，可以实现最先进的结果。我们还为稀疏专家模型提出了额外的分析和设计指南（或至少是我们的启发式方法）。此外，这项工作强调了共同优化上游预训练和下游曲线对称性，以避免差异（Tay等，2021）。

Contributions

1。对稳定技术质量稳定性权衡的大规模研究。

2。介绍了路由器Z-loss，该Z-loss解决了不稳定性问题，而略微即兴的模型质量3。对稀疏和密集模型的细胞调整分析，突出了对批次大小和学习率的不同超聚光度。尽管预训练大量训练，但我们显示出不良的超参数会导致对密集模型的细胞调整增长。4。在分布式设置中设计帕累托有效的Sparsemodels的建筑，路由和模型设计原理。5。定性分析可以追踪跨专家层次的令牌路由决策。

6。一个269b稀疏模型（稳定的可转移的Experts或ST-MOE-32B），可在各种自然语言基准中实现最先进的性能。

2 背景

稀疏的专家模型通常用一组专家替代神经网络层，每个神经网络都有haveunique的权重（Jacobs等，1991； Jordan和Jacobs，1994）。通常，相同类型和形状（同质）的层中的所有专家都可以多样化（异质）专家类型。输入仅由专家的子集处理以节省计算，因此添加了一种机制，以确定要发送每个输入的位置。通常，路由器或门控网络确定在哪里发送输入（即单词，句子，图像贴片等），但是已经普罗普尔（Lewis等，2021; Roller等，2021; Zuo et al。，2021; Clark等，2022）。

Shazeer等人特定于自然语言处理。（2017年）提出了一个含有的Experts（MOE）层，该层将令牌表示X作为输入，并将其路由到最佳匹配的Top-K专家中，从集合{E I（x）} Ni = 1中选择了Ni = 1专家。路由器变量W r产生logitsh（x）= w r·x，通过在thatlayer的可用n专家上通过软关系分布进行了归一化。专家I的登机口是由

$$p_i(x) = \frac{e^{h(x)_i}}{\sum_j^N e^{h(x)_j}} \tag{1}$$

以及令牌X的路由到具有最高顶部K门值的专家（索引t集）。该层的外点是每个专家计算的加权总和按门值

$$y = \sum_i p_i(x) E_i(x) \tag{2}$$

Shazeer等人最初在LSTMS（Hochreiter和Schmidhuber，1997）中提出的专家层（Vaswani等，2017）。（2018）和Lepikhin等。（2020）。Fedus等人的后续工作。（2021）将MOE进一步简化，以将令牌派往单个专家（TOP-1），并降低了其他成本以提高培训效率。

为了改善硬件利用，大多数稀疏模型的实现都具有每个专家的静态批量尺寸（Shazeer等，2017; 2018; 2018; Lepikhin等，2020; Fedus等，2021）。专家行动是指可以将其路由到每个专家的代币数量。如果超过此容量（路由器向该专家发送太多输入），则超过的代币不会对其进行计算，并且通过残差连接传递到下一个层。

Terminology	**Definition**
Expert	An independently-learned neural network with unique weights.
Router	A network that computes the probability of each token getting sent to each expert.
Top-n Routing	Routing algorithm where each token is routed to n experts.
Load Balancing Loss	An auxiliary (aux) loss to encourage each group of tokens to evenly distribute across experts.
Group Size	The global batch size is split into smaller groups, each of size Group Size. Each group is considered separately for load balancing across experts. Increasing it increases memory, computation, and communication.
Capacity Factor (CF)	Each expert can only process up to a fixed number of tokens, which is often set by evenly dividing across experts, $\frac{\text{tokens}}{\text{experts}}$. The capacity factor can expand or contract this amount to CF $\cdot \frac{\text{tokens}}{\text{experts}}$.
FFN	Acronym of Feed Forward Network (FFN) layer of Transformer consisting of linear, activation, linear.
Encoder-Decoder	A Transformer architectural variant that all of our models are based on. Consists of an encoder that does all-to-all attention on the inputs and a decoder that attends to the encoder and to its own inputs in an autoregressive manner.
`allreduce`	Communication primitive which sums a subset of n tensors on n different devices, then broadcasts the summed value to all n devices. This is used in distributed training for gradient accumulation and model parallelism.
`all2all`	Communication primitive where each device sends to every other device a part of its tensor. Used in sparse Transformer models for token routing.
(↑/↓)	Indicates whether higher/lower values are better (e.g. accuracy/train loss).

表1：术语中使用的术语。

输入令牌的批处理B分为跨数据并行尺寸2，每个尺寸为b/g。专家容量等于CF代表的CF代币/专家

2我们的实现依赖于用一热张量来调度和将张量组合到/fromexperts的Einsums。该单热张量的大小二次增长，因为将令牌的数量作为一个组，可以激励将批量分解为较小的组。可以通过稀疏的查找操作来避免这种情况。

容量因子超级参数，专家是专家的数量，而代币的数量是群体规模。如果增加了动力因子，则会产生额外的缓冲液，因此在载荷蒙平衡的情况下，将减少令牌。但是，增加容量因素也增加了记忆和计算量，因此存在权衡3。

最后，辅助负载平衡损失鼓励令牌在专家中大致分布均匀分布（Shazeer等，2017）。如上所述，这通过确保所有加速器都在同时处理大量数据来提高硬件效率。 Theloss的细节在附录A中提供了。但是，存在替代方案：Lewis等。 （2021）和Clark等人（2022）将平衡的令牌分配视为分配问题，并将辅助损失置于范围内。

3稀疏模型的稳定训练

稀疏模型通常患有训练不稳定性（图1）比Stan-Dard密集激活的变压器中观察到的差异要好。

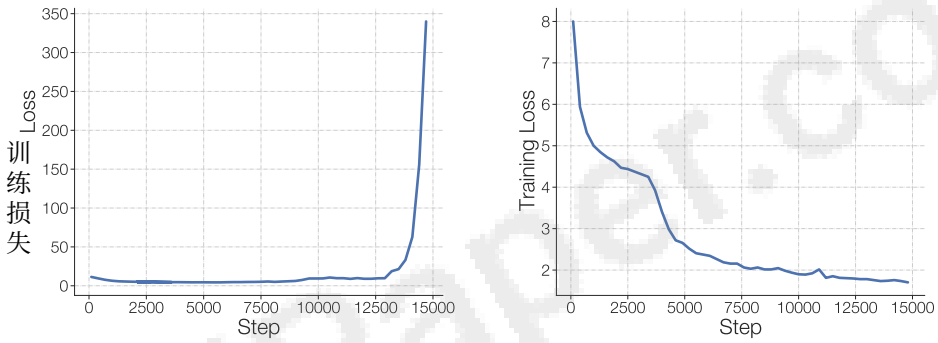


图1：稀疏模型的培训不稳定性。我们将培训不稳定性称为培训损失的分歧。以上是从稀疏模型触摸匹配到T5-XL版本的两次运行（Raffel等，2019），每个版本都使用Afafactor Optimizer培训了1M令牌（Shazeer and Stern，2018）。 （左）不稳定的训练。 （右）稳定的训练。

发现改善稳定性的更改是很简单的，但是，这些变化通常是对模型质量的不可限制的费用（例如，使用任意的学习率或使用紧密的剪裁）。我们对提高稳定性的几种方法进行分类并检查。 Sta-bility技术涉及变压器的通用固定以及特定于稀疏模型的特定模型：（1）删除乘法相互作用（2）注射模型噪声（3）约束激活和梯度。我们以我们的建议结束：新的辅助损失，路由器的Z-loss，显着地训练稳定性，没有质量降解。这是用于在网格张量电量代码库中用于最终软磁逻辑的Z-loss的改编（Shazeer等，2018）。

稳定稀疏模型

1。许多方法稳定稀疏模型，但牺牲质量较差。

2。路由器Z-loss稳定模型而没有质量降解。

3。具有更高的成分（geglu，rms和malization）的变压器修改会加剧稳定性，但提高了质量。

设计一项大规模稳定性研究。我们设计了针对T5-XL版本（Raffel等，2019）的稀疏模型的大规模稳定性研究（Xue等，2020）。每种稀疏型号都有32位专家，我们引入了一个稀疏的MUE层

3参见Fedus等。 （2021）用于图形说明容量因子的工作原理。

每四个FFN。火车容量因子为1.25，评估容量因子为2.0。参见表11，对本文使用的模型进行了更详细的描述。对于每种稳定技术，我们记录了稳定的分数，平均质量（英语上的负数为消除）以及在种子上的theSandard偏差。

构建这项研究的主要问题是，小型模型很少不稳定，但是大型不稳定的模型太昂贵了，无法运行舒适的步骤和种子。我们发现与T5-XL相匹配的稀疏模型是学习的好对象，因为它大约是不稳定的1/3跑步，Butwas训练仍然相对便宜。此外，由于我们发现了这种加剧的模型不稳定性，因此我们在多脊椎动物上进行了不稳定性实验，从而使我们能够在略微的小弹药板上进行实验。有关更多详细信息，请参见第9节。我们的基线配置使用六个随机种子和稳定技术使用三个随机种子训练。我们使用六种种子进行基线，以更好地表征不稳定性的速率，并使用蒙版语言建模目标预先训练Compute的不稳定性。，2018）。

3.1消除乘法互相时的稳定性和质量权衡

某些架构改进涉及的乘法比添加更多，或者不一次总和。例如，矩阵乘法对每次添加都有一个乘法，并且我的她不称其为“乘法”操作。我们介绍并分析了此处的变压器中乘法相互作用的两个实例。

GELU门式线性单元（geglu）。我们的第一个示例是封闭的线性单元（Dauphinet al。，2017），它是两个线性投影的组成产品，其中一个是通过Sigmoid函数最初通过的。 Shazeer（2020）将其扩展到其他变体，并呈现Agelu-linear（Hendrycks and Gimpel，2016）FFN层作为替代了通常的变压器（Nair Andhinton，2010）FFN。

$$FFN_{GEGLU}(x, W, V, b, c) = GELU(xW + b) \odot (xV + c) \tag{3}$$

在以后的工作中，这种质量增益得到了证实（Narang等，2021）。

均方根刻度参数。我们的第二个示例是Root MeansQuare（RMS）归一化中的比例参数（Zhang和Sennrich，2019年）。在变压器中，而不是背对背呼叫层，而是内部结构（称为Sublayer调用），可以改善gragradient的传播和训练动态。我们的Sublayer称之量与Raffel等人的称呼相匹配。（2019年）和包括：（1）RMS归一化，（2）层调用（例如自我注意力），（3）辍学（Srivastavaet al。，2014），（4）添加残留物（He等，2015）。 RMS归一化缩放输入矢量x∈R根据根平方的范围。然后，它通过用学习的比例参数G.y i = x i乘以乘以输出元件的元素

$$\frac{x_i}{\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}} g_i$$

表2显示，删除geglu层或RMS量表参数都可以提高稳定性，但对模型质量造成了重大损失。我们注意到，这些比例参数（g）具有与其他位置相比质量与参数（例如FFN）的不相称属性。与我们的发现一致，Shleiferet al。（2021）发现，在变形金刚制造的残差连接中添加了一个学习的乘法标量更加不稳定。

在附录C中，我们进一步研究了添加新的乘法相互作用的质量影响。我们发现，该操作几乎没有慢速模型阶梯时间，可以产生质量改进。

3.2添加噪音时的稳定性和质量权衡

接下来，我们探讨了一个假设，即在模型中增加噪声可以提高训练稳定性（Nee-Lakantan等，2015）。 Taleb（2012）认为某些系统具有抗差异性的特性，它们通过噪声改善。受到概念和我们的观察的启发

Method	Fraction Stable	Quality (↑)
Baseline	$4/6$	**-1.755** ± 0.02
Remove GEGLU	$3/3$	-1.849 ± 0.02
Remove RMS Norm. Scale Param	$3/3$	-2.020 ± 0.06

表2：删除具有更多乘法相互作用的操作。乘法相互作用会增强质量，但会破坏训练的稳定。单独删除两种多重分支机构的来源可提高稳定性，但质量却显着恶化。当我们删除geglulayer时，我们将其用等效的密度密集层代替，以匹配拖放器和参数。

（通过辍学注射噪声）很少不稳定，我们检查了训练噪声是否会破坏稀疏模型的稳定性。表3显示了稳定性的改善与基线相比，但以较低的质量为代价。我们还发现，Fedus等人引入的输入jitter。（2021），降低了XL尺度的质量，因此我们在模型中消融它。输入 - jitter将输入集乘以路由器，通过[1-10 -2，1 + 10 −2]之间的均匀随机变量乘以路由器。在整个变压器中都应用了我们的脱位。如前所述，小规模的改进在扩展时可能无法概括，因此应始终对趋势进行监控，并以增加规模来评估趋势（Kaplan等，2020）。

Method	Fraction Stable	Quality (↑)
Baseline	$4/6$	**-1.755** ± 0.02
Input jitter (10^{-2})	$3/3$	-1.777 ± 0.03
Dropout (0.1)	$3/3$	-1.822 ± 0.11

表3：在训练过程中注入噪音。输入jitter和辍学都提高了稳定性，但导致TOA明显损失模型质量。大多数方法都有明确的权衡：当一种提高性时，它通常会降低模型质量。我们的工作旨在找到可以解决稳定性而不会伤害质量的方法。

3.3限制激活和升级时的稳定性和质量权衡

稳定神经网络的最成功的方法之一是对激活的限制（Pascanu等，2013; Ioffe and Szegedy，2015; Salimans and Kingma，2016; Ba等，2016）。一种流行的方法在于逐步逐步逐渐爆炸梯度的剪辑，同时通过深层网络反向传播（Pascanu等，2013）。

在这项工作中，由于其内存效率（尽管最近引入的8位优化器（Dettmers等，2021）可能会提供更好的权衡取舍），因此我们使用了Afafactor优化器（尽管最近引入了8位优化器（Dettmers等人））。Afafactor并没有梯度插曲，而是使用更新剪辑，其中重量的更改被限制以使其具有一定的规范。我们试验将更新剪辑收紧到较小的值。

接下来，我们研究对路由器的逻辑的约束。路由器计算Float32精度专家的概率分布（即选择性精度）（Fedus等人，2021年）。但是，在最大的尺度上，我们认为这是不适合产生可靠训练的。要修复这个，请介绍路由器Z-loss，

$$L_z(x) = \frac{1}{B}\sum_{i=1}^{B}\left(\log\sum_{j=1}^{N} e^{x_j^{(i)}}\right)^2 \tag{5}$$

其中b是令牌的数量，n是专家的数量，x∈Rb×n是路由器的逻辑。这将大型logits惩罚到门控网络中，第3.4节包含了一个漫不经心的解释，说明为什么在路由器有用之前Z-loss。

表4显示，更新剪辑和路由器Z-loss都可以在所有3次运行中稳定模型，但是Theupdate剪辑显着伤害了模型质量。因此，由于质量和稳定性的提高，我们使用Z-loss方法进行固定模型稳定性4。

Method	Fraction Stable	Quality (↑)
Baseline	$4/6$	-1.755 $\pm$0.02
Update clipping (clip $= 0.1$)	$3/3$	-4.206 $\pm$0.17
Router Z-Loss	$3/3$	**-1.741** $\pm$0.02

表4：限制重量更新和路由器逻辑。限制更新剪辑的inadafactor可以提高稳定性，但质量损失却造成了灾难性的损失。较宽的剪裁值并不能稳定训练，因此我们在这里排除了培训。路由器Z-loss稳定了模型而没有质量降解（在这种情况下，我们会观察到质量略有提升）。

路由器Z-loss引入了另一个超参数（C Z），这是对总体损失优化的平均水平的能力。总损失是交叉熵（L CE），辅助负荷余额损失（L B）和路由器Z-LOSS（L Z）的线性加权组合，产生了总损失

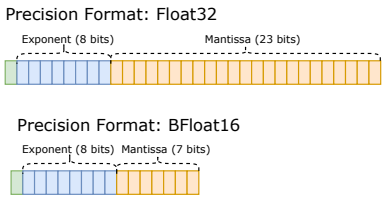
$$L_{tot} = L_{CE} + c_B L_B + c_z L_Z \tag{6}$$

我们根据高PA：扫描预训练后，根据最佳模型质量选择C Z = 0.001的值。附录B记录在预训练过程中所产生的损失。

3.4选择精确格式：交易效率和稳定性

与大多数现代分布式变压器一样，我们以混合精度训练（Micikevicius等，2017）5。权重存储在float32中以进行梯度更新，然后转换为BFLOAT16，当时进行矩阵乘法在前和后退6中。此外，可以在Bfloat16中存储和操作的所有ActivationSare，并且可以在Bfloat16或Float32数值精度中完成Alleduce通信。对于这项工作中探索的最大模型（稍后提出的ST-MOE-32B），我们发现加快了Allreduce的数值精度减半，但是这也可以破坏培训的稳定性，因此我们在整个工作中都将其保持为Float32。

较低的精度格式可以通过降低（a）处理器和内存之间的通信成本，（b）计算成本，（c）存储张量的内存（例如Activa-tions）来实现更多有效的模型。但是，较低的精度格式以较大的圆形错误为代价，这些错误以无法恢复的培训不稳定性为代价。



Number Range	Max BFloat16 Roundoff Error	Max Float32 Roundoff Error
[2, 4)	0.01563	2.34x10^(-7)
[32, 64)	0.25	3.81x10^(-6)
[1024, 2048)	8.0	0.00012
[2^20, 2^21)	8192.0	0.125
[2^30, 2^31)	8288608.0	128.0

图2：数值精度格式和圆形错误。较大的数字具有较大的圆形折射率。 BFLOAT16的圆形错误比Float32少65,536倍。路由器z-lossenciring将绝对数字的绝对幅度很小，这不会阻碍模型的表现减少圆形错误。路由器Z-loss最有效地完成了较大的错误会大大改变相对输出（例如指数和正弦函数）的功能。

4我们还尝试将Z-losses添加到注意力逻辑上，这也改善了模型的不稳定性。5请参阅 lity 实现详细信息的网状张量流量：https：//github.com/tensorflow/mesh/mesh/blob/blob/master/master/master/mesh_tensorflow/mesh_tensorflow/6 Matrix h/ 6 Matrix 6 Matrix TPU上的乘法在BFLOAT16中执行乘法，并在Float32中积累。 2.

了解精度格式和圆形错误。图2回顾了不同数字范围的不同精度格式的属性及其相应的圆形误差。数字在两个连续的范围中，有2个（例如[2,4）和[1024，2048）的数字由Mantissa位的Finumber表示（Bfloat16，7，Float32为23）。结果，（1）Bfloat16将大约65,536X（即23 - 7 = 16个额外的位和2 16 = 65536），因为较大的圆形错误Asfloat32和（2）较大的数字具有较大的圆形误差。由于8个指数位，NumberCan的幅度高达≈3E38，这甚至导致Float32遇到了一些圆形错误的问题。

稀疏的专家模型对圆形错误敏感，因为由于路由器，它们具有更多的指数功能。稀疏的专家模型引入了其他指数功能 - 通过路由器 - 可能会加剧圆形错误7并导致训练不稳定性。虽然圆形误差不会改变软磁性操作中概率的排序，但由于相对阈值，它会影响MOE中第二个令牌的路由（例如，如果第二个专家的门控概率仅在其第二名中，则仅将令牌置于其第二名。 /5作为第一位专家的大量asthat）。此外，圆形错误可能会大大改变专家输出的规模概率 - 我们发现这很重要。最后，我们猜想我们观察到的仅解码模型的稳定性（此处未显示）是因为它们具有较少的脑功能。第9节包含更详细的讨论。

路由器Z-loss上的一边。人们可能会认为，路由器Z-loss是可通过剪辑ligits来复杂的方法（Wu等，2016）。我们解释了为什么情况并非如此。目的是将大量的圆形错误变成指数函数。剪辑logits发生在任何ROUNDOFF错误之后 - 导致更大的不连续性。从一个角度来看，剪裁本身就是圆形折扣。相反，Z-loss自然会鼓励模型产生在ValueAnd中较小的逻辑，从而更准确地建模。由于这些动力学，我们确保所有凸起的张量均为float32。这暗示了神经网络具有更好数字格式的可能性，因为当整个网络中添加Z-posses时未使用的指数位（请参阅第9节）。

4稀疏模型的微调性能

最佳性能的语言模型通常是通过（1）对大量ofdata（例如互联网）进行预培训而获得的，其次是（2）对感兴趣的任务（例如Superglue）进行调整。新技术的诺言已经成为一种替代方法，包括很少的推理（Brown等，2020），预固定调整（Li and Liang，2021），及时调整（Lester等，2021）和适配器模块（ Houlsby等人，2019年） - 但是，与精细调整相比，质量差距仍然存在。由于这项工作，我们专注于这项工作中的精细调整，但在Du等人的几个镜头环境中突出了稀疏模型的最新成功。 （2021）; Artetxe等。 （2021）。此外，我们作为未来的工作技术离开，通过强化学习适应大型语言模型（Ouyang等，2022）

4.1假设：概括问题

稀疏模型在大型数据集的制度中表现出色，但是在调节时有时表现不佳（Fedus等，2021； Artetxe等，2021）。我们提供了一个证据（并不奇怪）假设，表明稀疏模型容易过度插入。我们通过Superglue中的两个任务（Wang等，2019） - 承诺银行（De Marneffeet al。，2019）和记录（Zhang等，2018）来说明这个问题。承诺银行（CB）有250个培训示例，同时记录超过100,000。这种显着尺寸的差异有助于自然研究，以便在两项任务中插入作为同一基准的一部分。

在图3中，我们比较了密度L和ST-MOE-L模型的细胞调整特性。每个模型在C4语料库的500B令牌上进行了预训练（Raffel等，2019）。模型

7指数函数具有小型输入扰动可以导致输出差异很大的属性。例如，考虑将10个逻辑输入到一个值为128和一个logitwith a值128.5的SoftMax函数。 BFLOAT16中0.5的循环误差将使SoftMax输出量增加36%，并且不正确地将所有逻辑相等。计算来自Exp（0）+10·Exp（0）≈0.091。这是因为SoftMax操作中的所有逻辑（用于数值稳定性）中的最大值，而循环错误则将数字从128.5变为128。此示例是在Bfloat16中，但是在Bfloat16中，但是类似的situsution在float32中发生，具有较大的logit值。

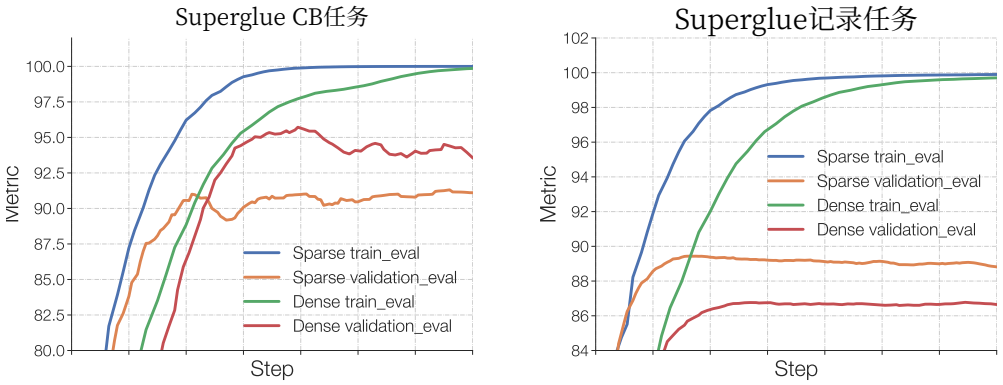


图3：稀疏模型容易拟合。我们为CB任务（250列火车序列）和记录（138K列车序列）绘制了ST-MOE-L的列车和验证曲线。在这两种情况下，稀疏模型在火车分区（Blue Eversgreen Line）上的学习更快。但是，对于较小的CB任务，密集模型的表现优于The-Ex-Out验证集（Red vs. Orange）上的稀疏模型。相比之下，在较大的记录任务上，稀疏模型Outperform将密集模型缩短了几个百分点。

被设计为从拉夫尔等人FromRaffel等人的T5-Large编码器模型的大致匹配的变体。（2019）具有770m参数。 ST-MOE型号具有32位专家的专家频率为1/4（每四个FFN层都用MOE层代替）。训练前和调整培训容量因子为1.25，而评估为2.0。我们评估持有效能和火车数据集分区的性能。

在这两个任务中，稀疏模型在数据分配变化下有效地优化了Parse模型，稀疏模型将其收敛到100%的火车设置精度。在较大的任务上，记录，稀疏模型的验证质量遵循训练的提升，显着超过了Thedense模型。但是，在较小的任务上，CB稀疏模型落在了固定数据上的密集对应物。根据Fedus等人的建议。 （2021），我们考虑增加掉落的专家隐藏状态（即专家辍学），但发现在这个规模上，更高的值只能改善质量（图4）。我们研究了第4.2节和第4.3节中的超参数敏感性的进一步改进。

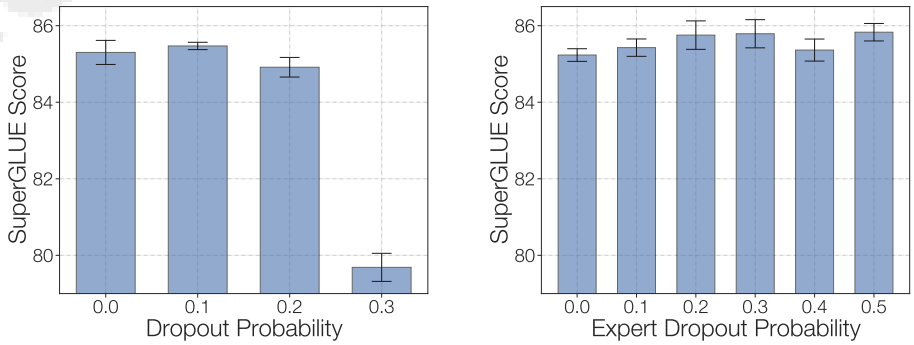


图4：稀疏模型的正规化研究。对于每种设置，我们都会训练随机种子，直到超级粘液上收敛。我们发现，增加正则化的遍布提供了适度的提升。 （左）以0.1的aglobal辍学率显示了峰值超粘合质量的质量。较高的价值过度限制并严重损害质量。 （右）以0.1的最著名全球辍学率开始，我们有选择地增加了专家辍学的效果（对专家隐藏激活的依赖性辍学率）。这使得进一步的概括有益于Fedus等人的发现。 （2021）。

4.2微调模型参数的子集以改善概括

为了打击过度拟合，我们在精细调整过程中仅更新模型参数的子集。图5衡量更新5个不同参数子集的质量：所有参数（全），仅非MOE参数（非MOE），仅MOE参数（MOE），只有自我关注和ENC-DEC注意参数（注意）和仅此非MOE FFN参数（FFN）。

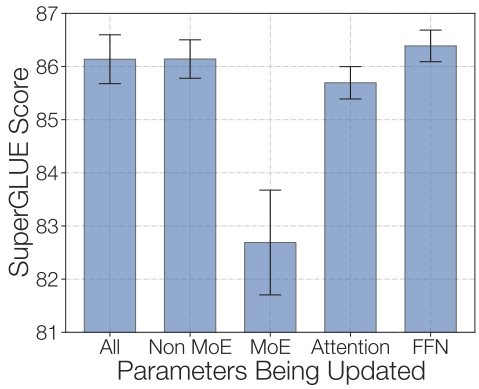


图5：仅更新细胞调整过程中的模型参数的子集。为了改善稀疏模型的发电和战斗过度拟合，我们为模型参数的子集进行了调整。 Allresult与St-MoE-L模型相关，平均是5种不同的随机种子。我们观察到更新参数的3/5似乎相同的作用，而仅对MoE参数进行调整会导致质量急剧降低。

我们观察到，更新非MOE参数可用，并更新所有参数，并且仅更新FFN参数的效果更好。仅更新MOE参数签名，从而降低了调整性能，这是≈80%的模型参数所在的位置。仅更新非MOE参数可以是一种有效的方法来加速和减少记忆以进行调整。

我们假设只对MOE参数进行调整会导致性能不佳，因为专家工作人员仅发生每1/4层一次，而代币每层最多会看到最多两个专家。因此，与更新我们尝试的参数的任何其他值相比，更新MOE参数的层和失败的影响要少得多。仅更新MOE参数会导致比更新非MOE参数更大的训练损失，即使有明显的PA框架。我们进一步观察到，更新所有非MOE参数会导致更高的培训效果，而不是更新所有参数，但是不幸的是，这种正则化效果并没有转化Tobetter验证性能。

此外，我们尝试过的一个常规化器是辍学变体，在训练过程中，整个专家都被掩盖了。但是，这无法改善我们的初步研究的概括。附录J在此实验上扩展，并包含其他负面结果。

4.3稀疏和密集的模型需要不同的微调协议

稀疏和密集的模型对精细调整方案的敏感性有多敏感？我们研究了两个超帕拉姆群：批处理大小和学习率。我们在500b Tokensof C4上为lenth-l和st-moe-l预算了一个密度，然后在超胶上填充。图6总结了我们的实验，其中表20（附录F）中的完整数据提示。在所有超参数设置中，稀疏模型（橙色）的表现都优于密集（蓝色）对应物 - 但是，每个设置的最佳设置可以实质性地改变。稀疏和密集的模型在不同的批次尺寸和学习速率上具有截然不同的性能。稀疏的模型从较小的批量大小和更高的学习率中受益。对过度拟合假设的态度（第4.1节），这两种变化都可能会改善概括过程中较高的噪声。最后，我们指出了在调节过程中纠正批次大小和学习率的重要性。简单地使用相同的调整超级

对密集模型运行良好的参数可以掩盖稀疏模型获得的任何预训练改进。

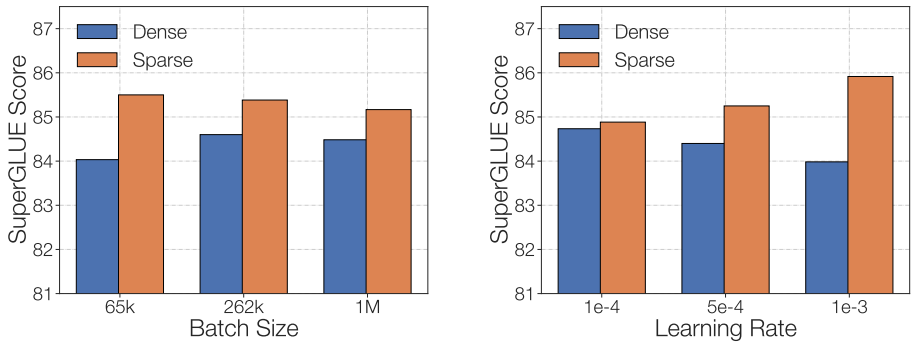


图6：批次尺寸和学习率灵敏度。我们测量密集（蓝色）和稀疏（橙色）模型之间对填充方案的差异和敏感性。每个条的平均在6差异的运行中，具有不同的超参数。在Superglue上，稀疏模型从Noisierhyperparameter中受益，包括小批量和高学习率。密集的模型几乎可以行事。有关所有数据，请参见附录F。

4.4稀疏模型在微调期间掉落令牌很强

稀疏模型将令牌路由到每一层的一个或多个专家。为了使这些模型具有现代硬件的SPMD范式中的有效，专家容量（每个专家程序的代币数量）需要提前固定（请参阅第2节）。当专家接收的代币比其容量时，额外的代币将被删除 - 不应用于Thosetokens。我们再次尝试通过（1）进行辅助损失进行预训练，该辅助损失促进了将令牌平等的代币发送给每个专家，以及（2）容量因子（一个超参数），该功能因素（一个超参数）为每个专家增添了额外的代币。我们试验在调节过程中关闭辅助损失并使用不同的容量因素。表5揭示了一个令人惊讶的结果，即通过下降到8个令牌8的10-15%而不会产生重大影响。关于ST-MOE-32BCOROBATER的研究，高容量因素不能提高调整质量。这是与Yang等人的发现结合的。（2021）不平等的负载平衡可能不会显着影响模型质量。

Model	Train CF	Eval CF	Aux Loss	Percent Tokens Dropped	SuperGLUE (↑)
Sparse	0.75	2.0	Yes	10.6%	86.5 ± 0.21
Sparse	1.25	2.0	Yes	0.3%	86.7
Sparse	2.0	3.0	Yes	0.0%	85.8
Sparse	4.0	5.0	Yes	0.0%	86.4
Sparse	0.75	2.0	No	15.6%	85.7
Sparse	1.25	2.0	No	2.9%	85.8
Sparse	2.0	3.0	No	0.4%	85.9
Sparse	4.0	5.0	No	0.0%	86.4

表5：稀疏模型在填充时掉落令牌。我们发现，在探讨的值中，超级粘液上的细胞量不会受到显着影响。有趣的是，掉落10-15%的令牌可以执行大约降低<1%的模型。我们还认为负载平衡损失（AUX损失）可以改善精细调整。下降的令牌百分比对应于在峰值验证精度下在所有专家层上掉落的令牌的比例。

8令牌掉落可能是正规化的一种形式，更广泛的研究可能是未来工作的一个有趣的方向。 ion

4.5微调期间插入前哨令牌

哨兵代币表示跨度腐败目标中的蒙面序列（Fedus等，2018； Devlinet al。，2018）。这与我们可能会遇到的任何填充任务不同，导致培训和精细调整之间的域匹配。表6说明了差异。我们检查了是否修改了调整任务，以看起来更像是训练前的任务效果结果。

Objective	Inputs	Targets
Span Corruption	I like **<X>** the pool **<Y>** day .	**<X>** going to **<Y>** on a sunny
Fine-Tuning	What is the capital of Illinois ?	Springfield
Fine-Tuning + Sentinels	What is the capital of Illinois ? **<X>**	**<X>** Springfield

表6：在填充训练过程中插入前哨的训练跨度目标。 wehighhighlight跨度腐败与精细调节之间的典型差异。我们建议修改调整任务，以类似于插入前哨代币的预训练。

在表7中，我们发现，在调节时添加哨兵令牌只能改善语法错误校正（GEC）（Rothe等，2021），而不是Superglue。我们试图通过插入多个哨兵令牌来进一步减少数据脱位转移（就像模型训练时遇到的那样），但再次发现没有普遍的好处。但是，尽管对持有的数据没有一致的好处，但我们发现训练收敛均加速了密集和稀疏模型。

Model	Insert Sentinel Tokens	SuperGLUE (↑)	GEC (↑)
Dense		84.9 ± 0.33	22.3 ± 0.25
Dense	✓	85.1 ± 0.25	22.1 ± 0.42
Sparse		86.6 ± 0.18	22.2 ± 0.04
Sparse	✓	86.6 ± 0.24	**22.9** ± 0.09

表7：哨兵令牌对修补的影响。在精细调整过程中，添加了前哨令牌（Lester等人（2021）中使用的类似概念）在Weconsider的两个任务上具有混合性能。 Superglue记录了平均得分，而GEC记录了确切的匹配。尽管我们发现ITDO并没有改善概括，但Sentinel代币可以加速培训融合。

5设计稀疏模型

Kaplan等人的基础工作指导着密集模型的设计。 （2020）。但是Sparse模型提出了无数其他问题：（1）使用多少专家？ （2）哪种开发算法？ （3）容量因子的价值是多少？ （4）硬件如何更改这些决策？在本节中，我们对这些评论，并为建立帕累托的稀疏模型提供建议。同时，Clark等。 （2022）提供了其他设计建议，包括根据Fedus等人的较高层频率和TOP-1路由。 （2021）。

设计稀疏模型

1。在我们的设置中，我们建议使用1.25容量因子的TOP-2路由，最多每核）2。可以在评估期间更改容量因子，以适应新的内存/计算标语3。密集的层堆叠和乘法偏差可以提高质量（附录C）。

5.1设置专家数量

第一个问题之一是要使用的专家人数。 Fedus等。 （2021）介绍了开关变压器的缩放范围，该缩放剂产生了单调预训练的好处（以步骤为基础）

C4高达512-Experts，Kim等人。（2021）多达64个专家和Clark等。（2022）最多512-Experts。但是，与许多专家（> 256）或等效地相同的增量收益迅速减少了非常庞大的模型（<1%的专家被激活）。

但是，反映特定的硬件系统可以进一步指导此选择。计算与记忆比（操作强度）可以作为对不同操作的效率的估计（Williams等，2009； Shazeer，2019）。如果安装张量的时间（例如ALU/MMU）大大超过了对张量进行计算所需的时间，则模型是绑定的。在现代GPU和TPU上，增加此计算与记忆比提高了效率。

返回稀疏的专家模型，每个核心使用多个专家会增加内存转移，从而损害效率。增加专家的数量不会更改计算方法（稀疏模型对每个输入应用固定量的计算），但增加了mem-ordor传输要求（必须从设备内存中加载其他专家变量）。这将降低计算与记忆比9。

在我们的TPU系统上，我们建议每个核心的专家（或更少）。我们最大的模型都使用数据和模型并行性，其中数据并行性超过了逻辑网格的“列”上的“行”和模型并行性。我们使用每个数据并行性行≤1专家，以确保计算到膜到 - 默里层高，并减少评估和推理所需的核心。此外，使用较少的专家，使我们可以将更多的核心分配给模型并行性“列”以在我们的模型中具有更多的拖鞋。附录H解释了我们的网格布局，因为当我们拥有的专家少于数据并行词。

5.2选择容量因子和路由算法

我们概括了TOP-1路由（Fedus等，2021; Roller等，2021）和Top-2（Shazeer等，2017; Lepikhin等，2020）以研究每个令牌为top-N路由由大多数N专家处理。在这项研究中，所有模型均针对100K步骤进行预训练，每个批处理为1M令牌，而稀疏模型具有32位专家，并且与T5-Large Raffel等人的翻牌相匹配。（2019）。我们得出两个关键结论。

首先，增加火车和评估能力因子（CF）可以提高质量，如表8的分段块所见。logperp。从1.0→1.25列车CF和TOP-2路由增加时，+0.009从1.25→2.0 Train CF提高。为了提供这些数字的上下文：将密集模型的大小（密度-l至密度XL）的大小增加三倍，可产生+0.090 neg。 log perp。促进。因此，这些CF增强量约为该幅度的1/10。但这是有代价的。提高能力因素线性增加了Einsums的成本，激活的内存，All2all通信成本以及专家层的模型 - 平行式通信成本10。

其次，给定固定容量因子（表8），在顶部N路由上有很小的顶部 -（n +1）。1/20 the the the the Model型三倍。这修改了Fedus等人的早期建议。（2021）。这些实验设置之间的主要差异是计算的规模。 Fedus等。（2021）训练有220m-flop匹配的型号，适用于50B令牌。我们以8倍大规模的训练（1B-Flop匹配的100B代币模型）找到，而是向多个专家提供的途径很小。此外，在较大的实验量表上，TOP-N与TOP-（N + 1）路由的速度差可以忽略不计。在Fedus等人中观察到速度差异。（2021）由于计算是总模型计算的较大部分。

9作为对读者的练习，验证第一个专家计算的操作强度为b·hb+h·e，具有bbtch b 尺寸，h隐藏尺寸，e专家数量。10ALL2ALL和ALLEDUCE成本取决于设备数量，取决于设备的数量，批量尺寸，D模型和容量因子，但没有专家数量。 or,

Algorithm	Train CF	Eval CF	Neg. Log Perp. (↑)
Dense-L	—	—	-1.474
Dense-XL	—	—	-1.384
Top-1	0.75	0.75	-1.428
Top-1	0.75	2.0	-1.404
Top-2	0.75	0.75	-1.424
Top-2	0.75	2.0	-1.402
Top-1	1.0	1.0	-1.397
Top-1	1.0	2.0	-1.384
Top-2	1.0	1.0	-1.392
Top-2	1.0	2.0	-1.378
Top-1	1.25	1.25	-1.378
Top-1	1.25	2.0	-1.373
Top-2	1.25	1.25	-1.375
Top-2	1.25	2.0	-1.369
Top-2	2.0	2.0	-1.360
Top-2	2.0	3.0	-1.359
Top-3	2.0	2.0	-1.360
Top-3	2.0	3.0	-1.356

表8：比较容量因子（CF）和路由算法。提高火车和EDARCF可以提高性能。如果您在评估时进行计算或更少的计算，则增加或减少了评估CF会提供额外的杠杆。接下来，比顶级N RoutingAcross容量因素的顶部（N + 1）较小。由于质量有所提高，但是随着CF的增加，速度会减慢，因此Pareto效率CF必须由特定的硬件系统确定。

特定的硬件软件系统将确定最佳的N和容量因子。对于现场，如果系统支持快速的All2All和Alleduce通信，那么在TOP-N路由中，较大的容量依赖器和较大的N可能是最佳的。但是，如果All2All和/或Allreducecommunications缓慢，那么较小的容量因素可能会占主导地位。在我们的情况下，硬件软件堆栈是TPU和网状张量流。我们在表9中记录了我们的St-Moe-Land ST-MOE-32B模型的训练速度，因为我们增加了火车容量因子。随着模型的扩展，更高的容量因素使模型越来越慢。 ST-MOE-L不需要并不需要并行性（在加速器内存中拟合，这意味着没有其他AllreduceCommunications）使其比我们的ST-MOE-32B模型更适合高容量因素。因此，由于最大的模型，我们继续使用Fedus等人提倡的1.25较小的火车容量因子。 （2021）对于帕累托的效率，与使用更大，更高的2.0容量因子的其他工作不同（Lepikhin等，2020; Du等，2021）。

Model	Train CF	Step Time (s) (↓)
ST-MoE-L	1.25	2.397
ST-MoE-L	2.0	2.447 (+7%)
ST-MoE-32B	1.25	4.244
ST-MoE-32B	2.0	4.819 (+14%)

表9：TPU上的稀疏模型。对于大型（1B）模型，将列车容量因子从1.25增加到2.0 increas，而我们的32B模型则将步骤时间提高到 +7%。随着theodel尺寸的增加，我们发现，从表8中，较高的火车容量因子的质量较小，而降低了14%的降低。注意：ST-MOE-L和St-MOE-32B之间的阶段时间是不可比服的，因为它们使用了不同数量的核心。

我们在本节中的结果侧重于Top-N路由，但是我们还尝试了附录J中的各种其他涂层技术。我们发现与顶级跑步相比，表现最多或更差。但是，我们发现Riquelme等人引入的批次优先路由（BPR）。 （2021），显着帮助少于一个的能力因素的性能（附录D）。我们建议

BPR用于较大的型号，其中All2All和Alleduce更昂贵，并且较低的容量因子是最佳的。

6 实验结果

鉴于我们对训练稳定性，精细调整和模型设计的改进，我们首先要验证稀疏模型，该模型近似与T5-Large匹配（Raffel等，2019）。我们通过设计和训练269B稀疏参数模型（flop匹配32B登录模型）来结束此分段，该模型在广泛的NLP任务中实现了最先进的质量。

我们在整个工作中研究了超级插曲（Wang等，2019）基准，其中包括包括情感分析（SST-2），Word Sense Disampain（WIC），句子相似性（MRPC，STS-B，QQP），自然的句子差异（WIC），句子sense sission disamigation（WIC），自然语言推论（MNLI，QNLI，RTE，CB），Question-words-ing（Multirc，Record，Boolq），Coreference解决方案（WNLI，WSC）和句子完成（COPA）（COPA）和句子可接受性（COLA）。我们经常在许多NLP任务中（但不能保证）性能在Superglue Tocorrelate上观察到良好的性能。我们还包括一个潜水员的额外基准测试。 CNN-DM（Hermann等，2015）和BBC Xsum（Narayan等，2018）数据集用于衡量总结文章的能力。问题答案是根据小队数据集（Rajpurkar等，2016）以及级别的科学问题进行探讨的，这是简单和弧形推理挑战（Clark等，2018）。就像罗伯茨等人一样。（2020年），通过对三个闭幕问题答案数据集进行详细说明：自然问题（Kwiatkowski等，2019），Web问题（Berant等，2013）和Trivia QA（Joshi et et al。 Al。，2017）。封闭式手册只是指没有任何补充参考orcontept材料提出的问题。为了衡量模型的常识推理，我们将其评估在Winograndeschema Challenge（Sakaguchi等，2020）中。最后，我们在对抗性NLI基准上测试了模型的自然语言推理能力（Nie等，2019）。

6.1 ST-MoE-L

为了简单起见并容易涵盖数十个任务，我们训练了列出的任务的混合物，而不是在每个任务上逐步调整模型。但是，由于任务的大小差异很大，因此示例数量的同样采样将过多样本的大型任务和样本下的小酮。因此，与Raffel等人一样，我们将每个任务与示例的数量成比例地与示例的数量（UP TORMOM MAX NUM示例= 65536）混合在一起。（2019）。这意味着包含超过65536个培训示例的任务是加权的，就好像它们仅包含Max NUM示例一样。

表10总结了密集的T5-LARGE（L）模型和稀疏模型的质量，该模型的质量大约是500K步骤的预先训练数量相同数量的FLOP，该步骤在C4 DataSet上具有1M批次尺寸（524B令牌）（Raffel等人）（Raffel等人）。，2019年）。编码器的序列长度为512和114的解码器。我们观察到跨越各种任务的验证（DEV）的改进，以实现自然语言理解，问答和摘要。如Feduset Al所示。（2021年），在封闭的书本回答中观察到了惊人的收益（Roberts等，2020）。

同样，为了支持第4.1节中提出的过度介绍假设，我们观察到两个小任务CB和WSC（分别为250和259个培训示例），是唯一的Sparsemodel不会在其密集的对应物上产生增长的唯一一个。这再次表明，改进的稀疏模型的规范化形式可能会释放更大的性能。

6.2 ST-MoE-32B

随着质量在T5-Large的规模上验证，我们试图通过ST-MOE-32B的稀疏模型来推动稀疏模型的能力。在设计此问题时，我们寻求在拖鞋和选举者之间的平衡。在Fedus等人中，高流动稀疏模型以前不稳定。（2021）在我们的设置（即cododer-decoder模型，afafactor Optimizer）中，但是路由器z-loss使我们能够继续进行。为了征收效率，我们扩大了专家的隐藏大小（下面表11中的D F F）11。最后，我们将D KV提高到128，以在硬件上更好地性能。最显着的变化是Fewer的总体参数，而每个令牌相对于Switch-C和Switch-XXL，每个令牌的差额更多。我们的

11通过模型并行性引入的Allreduce激活通信独立于命名的大小，而不是模型维度，这使其成为增加的好选择。 he

Name	Metric	Split	Dense-L (↑)	ST-MoE-L (↑)	Gain (%)
SQuADv2	F1	dev	94.0	**94.5**	+1%
SQuADv2	acc	dev	87.6	**88.1**	+1%
SuperGLUE	avg	dev	85.1	**87.4**	+3%
BoolQ	acc	dev	87.1	**88.6**	+2%
Copa	acc	dev	83.0	**91.0**	+10%
RTE	acc	dev	91.0	**92.1**	+1%
WiC	acc	dev	70.4	**74.0**	+5%
MultiRC	F1	dev	83.9	**86.0**	+3%
WSC	acc	dev	**95.2**	93.3	−2%
ReCoRD	acc	dev	85.7	**88.9**	+4%
CB	acc	dev	**100**	98.2	−2%
XSum	ROUGE-2	dev	19.9	**21.8**	+10%
CNN-DM	ROUGE-2	dev	20.3	**20.7**	+2%
WinoGrande (XL)	acc	dev	75.4	**81.7**	+8%
ANLI (R3)	acc	dev	54.3	**57.3**	+6%
ARC-Easy	acc	dev	63.5	**75.4**	+19%
ARC-Challenge	acc	dev	50.2	**56.9**	+13%
Closed Book TriviaQA	acc	dev	28.1	**33.8**	+20%
Closed Book NatQA	acc	dev	27.2	**29.5**	+8%
Closed Book WebQA	acc	dev	30.5	**33.2**	+9%

表10：翻牌匹配密度和稀疏模型的微调性能。比较密度基线和稀疏的flop匹配版本（更好的数字）。我们使用大约相同数量的计算观察到跨不同任务的持续分配。从稀疏模型中进行的两项任务是最小的两个任务：具有250个训练示例的CB和259的WSC。

ST-MOE-32B具有“仅” 269b参数，并且大约在具有32B参数的密集反式形式中触摸匹配。从开关C和Switch-XXL较低的参数计数减少了服务和调整的负担。最后，我们使用了描述的InappendixC的稀疏密度堆叠。

我们将1.5t代币的训练预先在仅英语C4数据集的混合物（Raffel等，2019）和Glam（Du等人，2021年）的Thedataset的混合物中，总结在附录E中。 Afafactor优化器具有默认的超参数，以及通过反平方根衰减遵循的10K步骤的学习率热身。我们的模型遵循了拟议的Infedus等人的初始化方案。 （2021）。

表12使用仅使用Inference-（零射，一次射击）和精细调整的先前最新方法评估了我们的ST-MOE-32B模型。在Superglue上，我们的模型提高了先前的最新模型，在测试服务器上的平均得分为91.2（93.2Validation Accorsy），超过一个百分点，超过了估计的人类能力。 Forboth摘要数据集，XSUM和CNN-DM，我们的模型在没有对培训或辅导的广告方面变化而实现了最新（Raffel等，2019; Liang等，2021）。 ST-MOE-32BIMPROV在ARC Easy（92.7→94.8）和ARC挑战（81.4→86.5）的测试服务器提交中的当前最新提交中。在三本封闭的质量保证任务中的两个中，我们改进了先前的最新时间。封闭的书籍WebQA达到了47.4的精度（前42.8个Frouberts等人（2020年），超过了Ernie 3.0 Titan260B密度参数模型的零拍摄的结果（Wang等，2021））。封闭的书NATQA提高到41.9的精度（Karpukhin等人（2020年）的41.5先前最佳状态）。我们发现对对抗性构造的数据集（ANLI R3和Winogrande XL）的显着改进。 Anli R3（Nie等人，2019年）将thestate thestate提高到74.7（53.4的优先级）。

我们注意到模型中的一些弱点。 ST-MOE-32B在SmallSquad数据集上表现不佳，精确的匹配分数为90.8，距离T5-XXL设置为91.3的较旧基准。此外，在为总体胶水设置新的最先进时，

Model	Parameters	FLOPs/seq	d_{model}	$\text{FFN}_{\text{GEGLU}}$	d_{ff}	d_{kv}
Dense-L	0.8B	645B	1024	✓	2816	64
T5-XXL	11.1B	6.3T	4096	✓	10240	64
Switch-XXL	395B	6.3T	4096	✓	10240	64
Switch-C	1571B	890B	2080		6144	64
ST-MoE-L	4.1B	645B	1024	✓	2816	64
ST-MoE-32B	269B	20.2T	5120	✓	20480	128

Model	Num. Heads	Num. Layers	Num. Experts	Expert Layer Freq.	Sparse-Dense
Dense-L	16	27	–	–	
T5-XXL	64	24	–	–	
Switch-XXL	64	24	64	1/4	
Switch-C	32	15	2048	1/1	
ST-MoE-L	16	27	32	1/4	✓
ST-MoE-32B	64	27	64	1/4	✓

表11：模型比较。比较密集的L和T5-XXL，这是两个最大的SwitchTransformer变体（Switch-XXL和Switch-C），以及ST-MOE-L和ST-MOE-32B。 D模型归于模型隐藏状态大小，而D F F是FFN层的内部大小。 D KV是每个注意力头的尺寸。专家层Freq。是FFN层的一部分被稀疏的层所取代。SparseGonene是指附录C中描述的架构变体。

			Previous Best (↑)			Ours (↑)
Name	**Metric**	**Split**	Zero-Shot	One-Shot	Fine-Tune	Fine-Tune
SQuADv2	F1	dev	68.3[e]	70.0[e]	96.2[a]	**96.3**
SQuADv2	acc	dev	62.1[e]	64.6[e]	**91.3**[a]	90.8
SuperGLUE	avg	test	–	–	90.9	**91.2**
BoolQ	acc	dev/test	83.0[e]	82.8[e]	92.0	**92.4**
Copa	acc	dev/test	91.0[d]	92.0[e]	98.2	**99.2**
RTE	acc	dev/test	68.8[e]	71.5[e]	**94.1**	93.5
WiC	acc	dev/test	50.5[e]	52.7[e]	**77.9**	77.7
MultiRC	F1	dev/test	72.9[d]	72.9[d]	88.6	**89.6**
WSC	acc	dev/test	84.9[e]	83.9[e]	**97.3**	96.6
ReCoRD	acc	dev/test	90.3[e]	90.8[e]	**96.4**	95.1
CB	acc	dev/test	46.4[d]	73.2[e]	**99.2**	98.0
XSum	ROUGE-2	test	–	–	24.6[h]	**27.1**
CNN-DM	ROUGE-2	test	–	–	21.6[a]	**21.7**
WinoGrande XL	acc	dev	73.4[e]	73.2[d]	–	96.1
ANLI R3	acc	test	40.9[e]	40.8[e]	53.4	**74.7**
ARC-Easy	acc	test	71.9[e]	76.6[e]	92.7[g]	**95.2**
ARC-Challenge	acc	test	51.4	53.2	81.4[g]	**86.5**
CB TriviaQA	em	dev	68.0[e]	**74.8**[e]	61.6[b]	62.3
CB NatQA	em	test	21.5[e]	23.9[e]	41.5[c]	**41.9**
CB WebQA	em	test	38.0[f]	25.3	42.8[b]	**47.4**

表12：ST-MOE-32B与以前的仅推理技术和精细的Mod-els。“开发/测试”的拆分是指零射门和一击和测试拆分以进行微调。数据无法用“ - ”填写。上标表示结果：A：Raffel等人（2019）B：Roberts等。（2020）C：Karpukhin等。（2020），D：Brown等。（2020），E：Du等。（2021），F：Wang等。（2021），G：unifiqa + arc mc/da + ir，h：zhang等。（2020）。

某些任务，包括CB，WSC等小型任务，无法改进。最后，在封闭的书籍triviaqa上，我们的模型通过Roberts等人的SSM对精细基线进行了改进。（2020年），但Failsto在GPT-3和Glam上产生增长。

虽然不是本文的重点，但我们介绍了最新进步技术之间的质量差异，例如很少的学习和对这些任务进行了细微的调整（GPT-3（Brownet al。，2020），Glam（Du等。，2021）和Gopher（Rae等，2021））。正如预期的和观察到的，精细的调整优于零/一击学习，但对于每个任务都需要辅助培训和不同模型的缺点。

7通过模型跟踪令牌

到目前为止，我们已经提出了定量措施和绩效指标。我们通过可视化令牌如何在专家之间进行路由来更改定性功能。我们这样做，绕过一批令牌到模型，并在每一层手动检查令牌分配。 Weconsis在单语C4语料库（Raffel等，2019）或多语言MC4语料库（Xue等，2020）上进行了预训练的ST-MOE-L模型。在编码器和解码器上，Themodel都有六个稀疏的层，每个层都有32个专家。

Preliminaries

跨度损坏目标是恢复输入中已删除的可变长度连续段的跨度。格式为：

输入：我去了购买 _ _ _ _

目标：<额外ID 0>商店<额外ID 1>牛奶 _ _

在我们的编码器架构中，输入将传递给编码器并靶向解码器。

每组代币均由Shazeer等人提出的辅助助剂激励的专家之间的负载平衡共同路由。（2017）（有关详细信息，请参见附录A）。代币竞争他们小组中其他代币的专家竞争，而不是整个批次，而专家专业人士则受到每个组中令牌分布的影响。引入了小组的概念，以限制向正确的专家派遣和收集正确的令牌的成本。

7.1编码专家展示专业

我们的第一个观察结果是，在每一层中，至少一位专家专门从事哨兵令牌（表示代表空白到填充的蒙版to-kens）。此外，一些编码器专家表现出明确的专业化，一些专家主要在标点符号，动词，专有名称，计数等方面运作。表13PRESENTERS PRESSENTS在Encoder专家中有一些著名的专业示例。尽管我们发现了专业化的许多建筑，但已从许多示例中提取了这些专业化，而没有Aclear语义或句法专业化。

7.2解码器专家缺乏专业

相比之下，专家专业化在解码器中的意义要差得多。在解码器专家之间，不仅在统一的方向统一（见表14），而且我们也没有在解码器专家中观察到专业化（语义或语法）。

我们假设缺乏有意义的专业专业化是由跨腐败目标引起的标记令的分布引起的。特别是（a）由于编码器中较长的序列长度（例如，在我们的设置中的解码器中的编码器vs 456）和（b）较高比例的tokensare tokensare，（a）由于编码器中的较长序列长度而在解码器中共同列出的较少数量）和（b）tokensare的比例较高。解码器中的前哨令牌。结果，每个组中的目标令牌通常涵盖一个小型的空间（与编码器相比），也许可以解释thedecoder中缺乏专家专业化。建筑与培训目标之间的这种复杂的相互作用进一步邀请

Expert specialization	Expert position	Routed tokens
Sentinel tokens	Layer 1	been <extra_id_4><extra_id_7>floral to <extra_id_10><extra_id_12><extra_id_15> <extra_id_17><extra_id_18><extra_id_19>...
	Layer 4	<extra_id_0><extra_id_1><extra_id_2> <extra_id_4><extra_id_6><extra_id_7> <extra_id_12><extra_id_13><extra_id_14>...
	Layer 6	<extra_id_0><extra_id_4><extra_id_5> <extra_id_6><extra_id_7><extra_id_14> <extra_id_16><extra_id_17><extra_id_18>...
Punctuation	Layer 2	, , , , , , , , , - , , , , ,).)
	Layer 6	, , , , , : . : , & , & & ? & - , , ? , , , . <extra_id_27>
Conjunctions and articles	Layer 3	The the the the the the the the the The the the the the the The the the the
	Layer 6	a and and and and and and and or and a and . the the if ? a designed does been is not
Verbs	Layer 1	died falling identified fell closed left posted lost felt left said read miss place struggling falling signed died falling designed based disagree submitted develop
Visual descriptions *color, spatial position*	Layer 0	her over her know dark upper dark outer center upper blue inner yellow raw mama bright bright over open your dark blue
Proper names	Layer 1	A Mart Gr Mart Kent Med Cor Tri Ca Mart R Mart Lorraine Colin Ken Sam Ken Gr Angel A Dou Now Ga GT Q Ga C Ko C Ko Ga G
Counting and numbers *written and numerical forms*	Layer 1	after 37 19. 6. 27 I I Seven 25 4, 54 I two dead we Some 2012 who we few lower each

表13：编码专家专业的著名示例。我们找到专家，专家专家，标点符号，连词和文章，动词，视觉描述，专有名称，计数和数字。在所有层中（未显示），我们观察到主要在前哨令牌上运行的专家（标记为）。请注意，如果词汇中不存在，例如，句子模型（Kudo和Richardson，2018年）将使该令牌属于代币，例如肯尼斯可能成为肯·肯（Ken ne）。

研究更好地利用解码器的稀疏性和专业专业化的研究。另外，未来的工作可以研究简单地删除解码器层中的专家，这也赋予了收益的自回旋解码（Kudugunta等，2021a）。

	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6	Uniform (32-experts)
Encoder	2.2	1.8	1.6	1.7	1.7	1.2	3.5
Decoder	3.4	3.4	3.4	3.4	3.4	3.4	3.5

表14：跨编码器和解码器层的路由前哨令牌的熵。我们支持编码专家专业的资格观察，但是解码器专家并不能通过计算哨兵的哨兵路由来进行哨兵代币的路由。编码器路由熵较低，但解释器是高熵的，几乎等于均匀的路由。由于每一层具有32个专家，因此完全均匀的分布的熵为3.5。

7.3多语言专家专业，但不是语言

接下来，我们考虑了一种在不同语言和编码器中专业专业的混合物中预测的多语言稀疏模型。与单语案例一样，我们找到了专家专业的有力证据。表15列出了一些专业的专家示例，专门从事前哨令牌，数字，连词和文章以及专有名称。

Expert specialization	Routed tokens
Sentinel tokens	to <extra_id_6>to til <extra_id_9> <extra_id_10>to <extra_id_14><extra_id_17> <extra_id_19><extra_id_20><extra_id_21>...
Numbers	$50 comment .10.2016 ! 20 20 3 ! 5 1. ! 91 ? né ? 2 17 4 17 11 17 8 & 11 & 22:30 02 2016.) iOS
Conjunctions & Articles	of of of their their of any this this your your am von this of Do of of This these our 的的于的在的在的 le les Le la di la sur sur 136 sur ののするのというのし
Prepositions & Conjunctions	For for or for for or or for from because https during https 并与和par c Pour à a par trè pour pour pour pour pour c とやのに でででなので- and and + c between and and
Proper names	Life Apple iOS A IGT 众莫HB F HB A K A OPP OK HB A Gia C Gia C P Scand Wi G H Z PC G Z ハイ PC G Ti CPU PC PC A キットOS

Table 15: **Examples of specialization in multilingual experts (encoder)**. Multilingual experts also exhibit specialization, which sometimes spans across different languages (e.g. ”for” and ”pour”). Experts trained on multilingual mixtures do not exhibit language specialization.

人们可能会期望专家专门研究语言，这似乎是专家之间数据批次的自然标准。但是，我们发现没有语言专业化的证据（可见15）。相反，路由器从英语，日语，法语和中文杂乱无章的和专家似乎是多语言的。但是，当考虑到令牌路由和负载平衡的机制时，这种缺乏语言专业化并不令人惊讶。由于每组TokensMay仅包含一个，因此最多只有几种语言（一组通常由2-4个序列INOUR设置组成），因此鼓励所有专家来处理所有语言的代币。我们试验了全球负载余额损失，但是，这通常会导致较差的负载平衡和较差的模型性能，因此我们将进一步改进的多语言专家模型作为开放工作领域（第9节）。

我们的可视化揭示了在我们的模型（表13，15）中学到的明显专业化，该层是对末积层的。在Shazeer等人（2017）的附录中也观察到了其他专家专业。但是，这导致了一个有趣的问题，即如何消除了学习的Roller等人。（2021）; Zuo等。（2021）表现良好。对学识渊博的与随机路由的标准特性进行的广泛研究可能会证明是对未来工作的帮助，并帮助我们更好地理解路由行为。

8 相关工作

Experts（MOE）的混合物至少可以追溯到Jacobs等人的工作。（1991）; Jordan和Jacobs（1994）。在最初的概念中，MoE定义了整个神经网络类似的toensemble方法。但是后来的Eigen等。（2013）扩展了将MOE作为更深层网络的组成部分的想法。 Shazeer等。（2017年）然后将这个想法缩放到137b参数模型，以示意机器翻译中的最新时间。后来的大多数工作（包括我们的）遵循ThisMoe作为组件方法。

扩展自然语言处理。自然语言处理中规模的显着成功（Kaplan等，2020; Brown等，2020）振兴了MOE研究。

最近工作的激增（Lepikhin等，2020; Fedus等，2021; Yang等，2021; Kim等，2021; Du等，2021; Attetxe等，2021; Zuo et al。。，2021年； Clark等，2022）。提出了稀疏的专家模型作为实现大规模致密模型结果的一种方法，更有效地。（2021）在T5-XXL上表现出4倍的预训练加速（Raffel等，2019），而Du等人（2021）仅使用1个使用1匹配GPT-3的质量（Brown等，2020）/3能量。在过去的十二个月的thespan中，多个群体已经实现了有效训练数万亿个参数深神经网络工作的里程碑（Fedus等，2021; Yang等，2021; Du et al。，2021），，2021年），，，，最近，Lin等人。（2021）引入了训练10T参数模型的技术。一个旁注是，稀疏专家模型最近的显着成功通常是在具有大量数据的设置中，没有分发的变化 - 两个例子是语言建模/跨度腐败和机动翻译（Shazeer等，2017； Lepikhin等，2020年，2020年Kim等人，2021年；在Fedus等人中，Fedus等人，2021年）。（2021）; Narang等。（2021）; Artetxe等。（2021），但我们期望的正规化技术进步，以继续提高下游质量。

采用更好的路由算法。基础层（Lewis等，2021）将令牌路由重现为Alinear分配问题 - 消除了负载平衡辅助损失的需求。这项工作示出了单个专家层的效率。克拉克等。（2022）深入研究了几种不同的路由算法的缩放量，并提出了自己的基础层变体，可提供最佳的运输配方。杨等。（2021）引入了M6-T体系结构和Expert原型设计，该原型将专家划分为不同的组并应用K TOP-1路由程序（与常用的其他地方使用的TOP-K路由对比）。 Hazimeh等。（2021）提出了相关性的稀疏门，对香草Top-K门控的改进。更激进的版本完全消除了学习路由。哈希层（Roller等，2021）显示了随机固定的路由（每个哈希功能），导致了通过学习的曲线竞争性能。 Zuo等。（2021）还提出了一种算法，该算法在训练和推断期间随机选择专家，并发现在开关变压器上获得了2个BLEU点，并且与Kim等人的较大模型相比，竞争得分。（2021）。最后，Fan等。（2021）设计一种具有解释语言特异性的子层的体系结构（而不是允许在Lepikhin等人（2020）中进行任意路由，以产生+1 BLEU的增长。

其他模式的稀疏专家模型。除了语言外，萌和稀疏的专家模型还以模式为先进。Riquelme等。（2021）设计了15b参数V-Moeto匹配的最新成像网（Deng等，2009）模型，具有较少的计算资源。Lou等。（2021）同样，通过使用跨图像贴片和通道尺寸的MOE层来表现出比密集视力模型的好处。此外，Severmoe变体已经改善了自动语音识别（You等，2021a; B）。 Kumatani等。（2021）使用序列到序列变压器和变压器传感器的MOE模型降低了Worderror速率。

改善稀疏模型的部署。最初的专家设计（包括这项工作）将每个用语分别路由到该层的专家。一个问题是，这些类型的体系结构可能是负担，因为它需要足够的存储器来存储参数。蒸馏显示了Fedus等人。（2021）要中等有效，但最近的方法修改了到稳定的路线或任务的路由（Kudugunta等，2021b; Zuo等，2021），然后在服务时允许子网进行分解（例如，仅部署与NewTask关联的网络）。作为蒸馏的替代方法，Kim等人。（2021）直接考虑将专家缩减为感兴趣的任务。

MOE多任务学习。我们结束了近期的MOE研究之旅，并取得了成功的inmultitask设置。 Ma等。（2018年）建议使用单独的门控或路由器网络对每个任务，这一想法很快可能会用于变压器体系结构。最后，Gururangan等人（2021年）建议使用更大的语言模型模块，并有条件地激活基于域/任务标签或通过推断标签的专家。

9 讨论

尽管这项工作是在稀疏模型上

神经网络体系结构进步。因此，我们的讨论涵盖了这项研究期间的广泛主题。

预先进行多语言数据进行预测的动态。我们经常观察到，在多语言数据上预先培训的这些模型将产生较小的预训练加速，并且更加不明显。一个假设是，这是由于跨批处理的每组序列的差异引起的。提醒我们，我们鼓励一组中的令牌负载平衡。每组通常只有2-8个序列（更高的昂贵），每个序列都以单个局部命令写入。因此，即使用100种语言进行培训，最多必须在专家之间平衡2-8种语言。这导致群体和批次之间的差异很大，导致混乱且不可预测的路由。在后续实验（仅针对简洁而突出显示）中，我们预先训练了英语C4的混合物，以及一小部分填充任务的一小部分，这类似地导致了Anunstable模型。

稀疏模型的鲁棒性。尽管有一篇论文重点介绍了稀疏模型 - 典型的细节，但缩放我们发现它们对广泛的超级参数和建筑款式具有强大的态度。稀疏模型在多种路由算法，降低的代币和不同的超参数下的表现都获得了出色的性能。虽然我们确实指出了与Kaplan等人在线的批量规模和学习率的重要性，但我们的直觉。（2020），真正的赢家是规模。例如，表8显示了通过简单地增加容量因子（即失败）而不是通过更复杂的路由（即算法）来获得的较大收益。

自适应计算。稀疏模型是自适应计算模型的一个子类，因为每个输入都将其应用于其上的不同计算。在稀疏的模型中，令牌将令牌路由到其选择的专家。当容量因素少于一个时，该模型就会学会不将计算机应用于某些令牌。这表明了计算机视觉（Riquelme等，2021）和我们的语言实验（附录D）的希望。我们设想未来的模型通过杂物明尼斯专家（例如，每个专家都应用不同的计算）。直观地，不同的输入示例可能会根据困难而需要不同的处理。通过新兴的计算基础架构将有效地实现这一方面的未来模型（Dean，2021）。

从小到大规模将发现概括。我们在整个工作中面临的一个关键问题是识别反映更大规模实验的小型模型和培训设置。这是我们在第3节中的稳定研究中发生的，其中必须使用XL尺寸的ModelSto表面相关动力学进行实验。对于我们的体系结构和路由算法实验，当模型经过更长的时间或更大的培训时，我们经常发现插图消失甚至反向。作为一个例子，Fedus等人的顶级发现。（2021）在我们在这里提到的8倍大规模实验中反转，该实验揭示了顶部 -（n + 1）在Top-N路由上的较小提升（请参见表8）。

培训模型的精度更低。我们发现稳定模型的最佳方法是Router Z-loss。这是一个辅助损失，这鼓励模型逻辑在绝对幅度上具有较小的值。鉴于最大范围float32和bfloat16可以支持（～3e 38），这使我们相信不需要大部分范围，并且压缩它实际上可能会改善模型训练动力学。因此，未来的精度格式可能会考虑到更受压缩的指数范围来训练某些模型的阶级。

设计具有更多乘法交互的新操作。第3.1节表明，与添加相比，具有更多乘法交互作用的操作性，或者不积分过度数字的操作性提高模型性能。我们通过向专家层注入更多的乘法层次来对此进行进一步测试，这些层次将预先培训加速4%，而没有任何变化的步骤时间（附录C）。我们认为这暗示着有希望的模型建筑改进，并且可能是设计原则。最近，仅积累了3-5个元素的深度卷积，Havealso被证明可极大地改善变压器的性能（So等，2021）。这些操作特别令人兴奋，因为使用模型并行性时，通常不会引入任何通信头（这使得像Depthiswise Resloctions andour乘法相互作用非常有效）。虽然我们确实注意到了这些方法以增加3.1节中的模型构成，但使用路由器Z-loss在我们的模型中阻止了任何进一步的不稳定性。

限制激活以减轻其他不良模型缩放动力学。我们观察到其他训练不稳定性的来源。（1）编码器模型是仅使用模型（用于固定量的拖鞋）的更不稳定的模型。由于对解码器上每个FFN具有自我注意事项和ENCECTECTENTION层，编码器模型的注意力层的比例更高（例如，指数函数更大）。（2）对于固定数量的拖鞋而言，更深的模型比浅层模型更不稳定。更深层次的模型还通过额外的注意层引入了更多的指数功能。我们假设对这两个观察者的促成因素仅仅是网络中发现的指数函数的数量增加。未来的工作应该通过在注意力效果上增加Z-loss的惩罚来解决这些训练动态，以解决非SPARSE模型，尤其是因为我们观察到添加它们并没有改变模型质量。

密集和稀疏的模型在高参数方面的依赖性不同。第4.3节中，我们的细胞调整分析显示，在密集和帕斯斯模型之间，最佳的调整超参数差异很大。在某些设置中，对密集的模型效果很好的填充高票量可以从稀疏模型中进行任何改进（尽管进行了大量的训练速度）。对于NewModel课程，我们建议研究人员和从业人员在过早放弃方法之前广泛测试关键的超参数。

10 结论

我们在Fedus等人的规模过度降低了量表。（2021）通过显示1/5的模型如何，但可以更好地平衡计算（flops）与参数 - 是一个更有效的SparSearner。此外，这可以提高稀疏模型的可用性，因为它可以用不级数的开销来部署。使用我们稀疏的模型变体，我们可以在广泛的Themost竞争性公共基准中实现SOTA。我们希望这项工作显示出模型稀疏的力量，并能够采用这种模型。

ACKNOWLEDGEMENTS

我们要感谢Alex Passos，Ekin Cubuk，Margaret Li，Noah Constant，Oriol Vinyals，Basil Mustafa，Joan Puigcerver，Diego de Las Casas，Mike Lewis和Ryan Sepassi，以获取有关草稿早期版本的详细信息和反馈。在整个工作过程中，我们还要感谢Google Brain团队进行讨论。

REFERENCES

Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, Giri Anantharaman, Xian Li, Shuohui Chen, Halil Akin, Mandeep Baines, Louis Martin, Xing Zhou, Punit Singh Koura, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Mona Diab, Zornitsa Kozareva, and Ves Stoyanov. Efficient large scale language modeling with mixtures of experts, 2021.

吉米·雷巴、杰米·瑞恩·基罗斯和杰弗里·E·辛顿。层标准化。 arXiv 预印本arXiv：1607.06450，2016。

Emmanuel Bengio，Pierre-Luc Bacon，Joelle Pineau和Doina Precup。更快的模型的有条件计算在神经网络中，2016年。

乔纳森·贝兰特（Jonathan Berant），安德鲁（Andrew Chou），罗伊·弗罗斯特（Roy Frostig）和珀西·梁（Percy Liang）。在freebase fromQuestion-Asswer对上进行语义解析。在2013年自然语言处理中经验方法会议的会议记录中，第1533-1544页，2013年。

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Aidan Clark, Diego de las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, et al. Unified scaling laws for routed language models. *arXiv preprint arXiv:2202.01169*, 2022.

Peter Clark、Isaac Cowhey、Oren Etzioni、Tushar Khot、Ashish Sabharwal、Carissa Schoenick 和 Oyvind Tafjord。您认为您已经解决了问答问题吗？尝试 arc，ai2 推理挑战。arXiv 预印本 arXiv：1803.05457，2018。

Yann N Dauphin，Angela Fan，Michael Auli和David Grangier。使用卷积网络进行语言建模。在机器学习国际会议上，第933–941页。PMLR，2017年。

玛丽-凯瑟琳·德·玛内夫、曼迪·西蒙斯和朱迪思·汤豪瑟。承诺库：调查自然发生的话语中的投射。《Sinn und Bedeutung》论文集，第 23 卷，第 107-124 页，2019 年。

Jeff Dean. Introducing pathways: A next-generation ai architecture. *Google AI Blog*, 2021.

Jia Deng，Wei Dong，Richard Socher，Li-Jia Li，Kai Li和Li Fei-Fei。 ImageNet：一个大规模的HIERARCALICAL IMAGE数据库。在2009年IEEE计算机视觉和模式识别会议上，第248-255页。 IEEE，2009年。

Tim Dettmers，Mike Lewis，Sam Shleifer和Luke Zettlemoyer。 8位优化器通过Block-WiseQuantization，2021。

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathy Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. Glam: Efficient scaling of language models with mixture-of-experts, 2021.

大卫·艾根、马克·奥雷利奥·兰扎托和伊利亚·苏茨克韦尔。学习在专家的深度混合中分解表征。 arXiv 预印本 arXiv:1312.4314, 2013。

Angela Fan，Shruti Bhosale，Holger Schwenk，Zhiyi MA，Ahmed El-Kishky，Siddharth Goyal，Man-Deep Baines，Onur Celebi，Guillaume Wenzek，Vishrav Chaudhary等。超越英语中心的机器翻译。机器学习研究杂志，22（107）：1-48，2021。

William Fedus, Ian Goodfellow, and Andrew M Dai. Maskgan: Better text generation via filling in the_. *arXiv preprint arXiv:1801.07736*, 2018.

威廉·费杜斯、巴雷特·佐夫和诺姆·沙泽尔。开关变压器：通过简单高效的稀疏性扩展到万亿参数模型。 arXiv 预印本 arXiv:2101.03961, 2021。

Suchin Gururangan，Mike Lewis，Ari Holtzman，Noah A. Smith和Luke Zettlemoyer。Demixlayers：模块化语言建模的解开域，2021。

Hussein Hazimeh、赵哲、Aakanksha Chowdhery、Maheswaran Sathiamoorthy、Yihua Chen、Rahul Mazumder、Lichan Hong 和 Ed H. Chi。 Dselect-k：专家混合中的可微分选择及其在多任务学习中的应用，2021 年。

何凯明、张翔宇、任少清、孙健。用于图像识别的深度残差学习，2015。

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Karl Moritz Hermann，Tomas Kocisky，Edward Grefenstette，Lasse Espeholt，Will Kay，Mustafa Suleyman和Phil Blunsom。加内特（Garnett），《AdvanceS中的神经信息处理系统》的编辑，第28卷，第1693–1701页。 Curran Asso-Ciates，Inc.，2015年。URLhttps：//proceedings.neurips.cc/paper/2015/file/afdec7005cc9f14302cd047474fd0fd0f3c96-paper.pdf。

塞普·霍克赖特 (Sepp Hochreiter) 和于尔根·施米德胡贝尔 (Jürgen Schmidhuber)。长短期记忆。神经计算，9(8):1735–1780, 1997。

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019.

Sergey Ioffe和Christian Szegedy。分批归一化：通过回复内部协变量转移来加速深层网络训练。在机器学习国际会议上，第448-456页。 PMLR，2015年。

罗伯特·A·雅各布斯、迈克尔·乔丹、史蒂文·J·诺兰和杰弗里·E·辛顿。当地专家的适应性组合。神经计算，3(1):79–87, 1991。

迈克尔·乔丹和罗伯特·A·雅各布斯。专家的分层混合和 em 算法。神经计算，6(2):181–214, 1994。

曼达尔·乔希、恩索尔·崔、丹尼尔·S·韦尔德和卢克·泽特莫耶。 Triviaqa：用于阅读理解的大规模远程监督挑战数据集。 arXiv 预印本 arXiv:1705.03551, 2017。

贾里德·卡普兰、萨姆·麦坎德利什、汤姆·赫尼根、汤姆·B·布朗、本杰明·切斯、瑞旺·柴尔德、斯科特·格雷、亚历克·雷德福、杰弗里·吴和达里奥·阿莫迪。神经语言模型的缩放定律。 arXiv 预印本 arXiv:2001.08361, 2020。

Vladimir Karpukhin，Barlas Ouz，Sewon Min，Patrick Lewis，Ledell Wu，Sergey Edunov，Danqichen和Wen Tau Yih。四层通道检索，用于开放域的问题回答，2020年。

Young Jin Kim, Ammar Ahmad Awan, Alexandre Muzio, Andres Felipe Cruz Salinas, Liyang Lu, Amr Hendy, Samyam Rajbhandari, Yuxiong He, and Hany Hassan Awadalla. Scalable and efficient moe training for multitask multilingual models, 2021.

工藤卓和约翰·理查森。 Sentencepiece：用于神经文本处理的简单且独立于语言的子词标记器和去标记器。 arXiv 预印本 arXiv:1808.06226, 2018。

Sneha Kudugunta, Yanping Huang, Ankur Bapna, Maxim Krikun, Dmitry Lepikhin, Minh-Thang Luong, and Orhan Firat. Beyond distillation: Task-level mixture-of-experts for efficient inference. *arXiv preprint arXiv:2110.03742*, 2021a.

Sneha Kudugunta, Yanping Huang, Ankur Bapna, Maxim Krikun, Dmitry Lepikhin, Minh-Thang Luong, and Orhan Firat. Beyond distillation: Task-level mixture-of-experts for efficient inference. *arXiv preprint arXiv:2110.03742*, 2021b.

Kenichi Kumatani，Robert Gmyr，Felipe Cruz Salinas，Linquan Liu，Wei Zuo，Devang Patel，Ericsun和Yu Shi。建立一位伟大的多语言老师，并与稀疏门控的专家换句话说，2021年。

汤姆·科维亚特科斯基、珍妮玛丽亚·帕洛马基、奥利维亚·雷德菲尔德、迈克尔·柯林斯、安库尔·帕里克、克里斯·阿尔伯蒂、丹妮尔·爱泼斯坦、伊利亚·波洛苏欣、雅各布·德夫林、肯顿·李等。自然问题：问答研究的基准。计算语言学协会汇刊，7:453–466，2019。

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

布莱恩·莱斯特、拉米·艾尔富和诺亚·康斯坦特。参数高效提示调整的规模力量。 arXiv 预印本 arXiv:2104.08691, 2021。

Mike Lewis，Shruti Bhosale，Tim Dettmers，Naman Goyal和Luke Zettlemoyer。基层：简化大型稀疏模型的培训。 Arxiv预印型ARXIV：2103.16716，2021。

李翔丽莎和梁珀西。前缀调优：优化生成的连续提示。 arXiv 预印本 arXiv:2101.00190, 2021。

Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. R-drop: Regularized dropout for neural networks, 2021.

Junyang Lin, An Yang, Jinze Bai, Chang Zhou, Le Jiang, Xianyan Jia, Ang Wang, Jie Zhang, Yong Li, Wei Lin, Jingren Zhou, and Hongxia Yang. M6-10t: A sharing-delinking paradigm for efficient multi-trillion parameter pretraining, 2021.

Yuxuan Lou，Fuzhao Xue，Zangwei Zheng和Yang您。稀疏-MLP：全MLP架构以及条件计算，2021。

Jiaqi MA，Zhe Zhao，Xinyang Yi，Jilin Chen，Lichan Hong和Ed H Chi。使用多门的混合物在多任务学习中建模任务关系。在第24届ACMSIGKDD国际知识发现与数据挖掘会议论文集，第1930- 1939年，2018年。

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017.

维诺德·奈尔和杰弗里·E·辛顿。整流线性单元改进了受限玻尔兹曼机。In Icml，2010。

Sharan Narang, Hyung Won Chung, Yi Tay, William Fedus, Thibault Fevry, Michael Matena, Karishma Malkan, Noah Fiedel, Noam Shazeer, Zhenzhong Lan, et al. Do transformer modifications transfer across implementations and applications? *arXiv preprint arXiv:2102.11972*, 2021.

沙希·纳拉扬 (Shashi Narayan)、谢伊·B·科恩 (Shay B Cohen) 和米雷拉·拉帕塔 (Mirella Lapata)。不要给我细节，只给我概要！用于极端概括的主题感知卷积神经网络。 arXiv 预印本arXiv：1808.08745，2018。

Arvind Neelakantan、Luke Vilnis、Quoc V Le、Ilya Sutskever、Lukasz Kaiser、Karol Kurach 和 James Martens。添加梯度噪声可以改善非常深的网络的学习。 arXiv 预印本arXiv：1511.06807，2015。

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*, 2019.

Long Ouyang，Jeff Wu，Xu Jiang，Diogo Almeida，Carroll L Wainwright，Pamela Mishkin，Chongzhang，Sandhini Agarwal，Katarina Slama，Alex Ray等。培训语言模型以跟随人类反馈。2022。

Razvan Pascanu，Tomas Mikolov和Yoshua Bengio。关于训练复发性神经网络的困难。在机器学习国际会议上，第1310-1318页。PMLR，2013年。

大卫·帕特森、约瑟夫·冈萨雷斯、Quoc Le、陈亮、路易斯-米克尔·蒙吉亚、丹尼尔·罗斯柴尔德、大卫·苏、莫德·特克希尔和杰夫·迪恩。碳排放和大型神经网络训练。arXiv 预印本 arXiv：2104.10350，2021。

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. Scaling language models: Methods, analysis & insights from training gopher, 2021.

Colin Raffel、Noam Shazeer、Adam Roberts、Katherine Lee、Sharan Narang、Michael Matena、YanqiZhou、Wei Li 和 Peter J Liu。使用统一的文本到文本转换器探索迁移学习的局限性。arXiv 预印本 arXiv：1910.10683，2019。

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

Carlos Riquelme，Joan Puigcerver，Basil Mustafa，Maxim Neumann，Rodolphe Jenatton，AndréSu-Sano Pinto，Daniel Keysers和Neil Houlsby。与专家的稀疏混合物缩放视觉。ARXIV预印型 ARXIV：2106.05974，2021。

亚当·罗伯茨、科林·拉斐尔和诺姆·沙泽尔。您可以将多少知识装入语言模型的参数中？ arXiv 预印本 arXiv:2002.08910, 2020。

Stephen Roller, Sainbayar Sukhbaatar, Arthur Szlam, and Jason Weston. Hash layers for large sparse models. *arXiv preprint arXiv:2106.04426*, 2021.

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. A simple recipe for multilingual grammatical error correction. *arXiv preprint arXiv:2106.03830*, 2021.

Keisuke Sakaguchi，Ronan Le Bras，Chandra Bhagavatula和Yejin Choi。Winogrande：广告中的Winograd模式挑战赛。在AAAI人工智能会议论文集，第34卷，第8732–8740页，2020年。

蒂姆·萨利曼斯（Tim Salimans）和Durk P Kingma。重量归一化：简单的重新聚集到深度神经网络的加速。神经信息处理系统的进步，29：901–909，2016。

诺姆·沙泽尔。快速变压器解码：您只需要一个写入头。 arXiv preprintarXiv:1911.02150, 2019。

Noam Shazeer. Glu variants improve transformer, 2020.

Noam Shazeer和Mitchell Stern。 Afafactor：具有子宫内记忆成本的自适应学习率。在机器学习国际会议上，第4596–4604页。 PMLR，2018年。

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Noam Shazeer，Youlong Cheng，Niki Parmar，Dustin Tran，Ashish Vaswani，Penporn Koanantakool，Peter Hawkins，Hyoukjoong Lee，Mingsheng Hong Hong，Cliff Young等。网格味流量：超级计算机的深度学习。在神经信息处理系统的进步中，第10414-10423页，2018年。

Sam Shleifer, Jason Weston, and Myle Ott. Normformer: Improved transformer pretraining with extra normalization. *arXiv preprint arXiv:2110.09456*, 2021.

David R So，WojciechMańke，Hanxiao Liu，Zihang Dai，Noam Shazeer和Quoc V le。底漆：寻找有效的变压器进行语言建模。 ARXIV预印型ARXIV：2109.08668,2021。

Nitish Srivastava，Geoffrey E. Hinton，Alex Krizhevsky，Ilya Sutskever和Ruslan Salakhutdinov.Dropout。机器学习研究杂志，15（1）：1929–1958，2014。URL http：//www.cs.toronto.edu/～rsalakhu/papers/srivastava14a.pdf。

纳西姆·尼古拉斯·塔勒布（Nassim Nicholas Taleb）。抗碎片：从疾病中获得的事物，第3卷。随机居住，2012年。

Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. Scale efficiently: Insights from pre-training and fine-tuning transformers. *arXiv preprint arXiv:2109.10686*, 2021.

Ashish Vaswani，Noam Shazeer，Niki Parmar，Jakob Uszkoreit，Llion Jones，Aidan N Gomez，olukasz Kaiser和Illia Polosukhin。注意就是您所需要的。在神经信息处理系统的进步中，第5998-6008页，2017年。

Alex Wang，Yada Pruksachatkun，Nikita Nangia，Amanpreet Singh，Julian Michael，Felix Hill，Omerlevy和Samuel Bowman。 Superglue：通用语言知识系统的粘性基准。在神经信息处理系统的进展中，第3266-3280页，2019年。

Shuohuan Wang，Yu Sun，Yang Xiang，Zhihua Wu，Siyu Ding，Weibao Gong，Shikun Feng，Jun-Yuan Shang，Yanbin Zhao，Chao Pang等。 Ernie 3.0 Titan：探索大规模的知识进行培训，以了解语言理解和产生。 ARXIV预印型ARXIV：2112.12731,2021。

塞缪尔·威廉姆斯，安德鲁·沃特曼和戴维·帕特森。 Roo FiNE：一种用于多核心体系结构的有见识的视觉性能模型。 ACM的通信，52（4）：65–76，2009。

吴永辉、Mike Schuster、陈志峰、Quoc V Le、Mohammad Norouzi、Wolfgang Macherey、Maxim Krikun、曹元、高琴、Klaus Macherey 等。谷歌的神经机器翻译系统：弥合人类和机器翻译之间的差距。 arXiv 预印本arXiv：1609.08144，2016。

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*, 2020.

An Yang，Junyang Lin，Rui Men，Chang Zhou，Le Jiang，Xianyan Jia，Ang Wang，Jie Zhang，Jia-Mang Wang，Yong Li，Di Zhang，Di Zhang，Wei Lin，Lin Qu，Jingren Zhou和Hongxia Yang。 M6-T：探索2021年稀疏专家模型及以后。

Zhao You，Shulin Feng，Dan Su和Dong Yu。 SpeechMoe：将专家的动力路由混合物缩放到大型声学模型，2021a。

Zhao You，Shulin Feng，Dan Su和Dong Yu。 SpeechMoe2：Experts模型与IM冠军路由的混合物，2021b。

张彪和 Rico Sennrich。均方根层归一化。 arXiv 预印本arXiv：1910.07467，2019。

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, 2020.

张盛、刘晓东、刘晶晶、高剑峰、Kevin Duh 和 Benjamin Van Durme。记录：弥合人类和机器常识阅读理解之间的差距。arXiv 预印本 arXiv：1810.12885，2018。

Simiao Zuo, Xiaodong Liu, Jian Jiao, Young Jin Kim, Hany Hassan, Ruofei Zhang, Tuo Zhao, and Jianfeng Gao. Taming sparsely activated transformer with stochastic experts, 2021.

令牌负载余额描述

Shazeer等人的辅助负载平衡损失。（2017年）也可以在这里来平衡Tokensacross专家。假设我们的n专家由i = 1至n索引，而t batch b则具有t令牌。计算文工损失是作为向量F和p之间的缩放点产物计算的，

$$\text{loss} = \alpha \cdot N \cdot \sum_{i=1}^{N} f_i \cdot P_i \tag{7}$$

f i是派遣给专家i的代币的比例

$$f_i = \frac{1}{T} \sum_{x \in \mathcal{B}} \mathbb{1}\{\text{argmax } p(x), i\} \tag{8}$$

P I是为专家i分配的路由器概率的比例

$$P_i = \frac{1}{T} \sum_{x \in \mathcal{B}} p_i(x) \tag{9}$$

由于我们寻求跨N专家的一批令牌的统一路由，因此我们希望两个矢量的值为1/n。方程7的辅助损失鼓励统一路由，因为它在均匀分布下进行了限制。该目标也可以区分为p -vector差异化，但f -vector却没有。最终的损失乘以专家计数n，以保持损失范围，因为专家的数量在统一路由下变化
$\frac{1}{n}$。最后，超参数α是这些辅助损失的多重系数。在整个工作中，我们都使用α= 10 -2，它足够大，以确保负载平衡，同时又不压倒主要的跨透镜物镜。

B路由器Z-loss训练动力学

图7绘制了方程5的路由器Z-loss跨系数扫描，其中最佳值OFC z = 0.001绘制为编码器和解码器的绿色绘制。

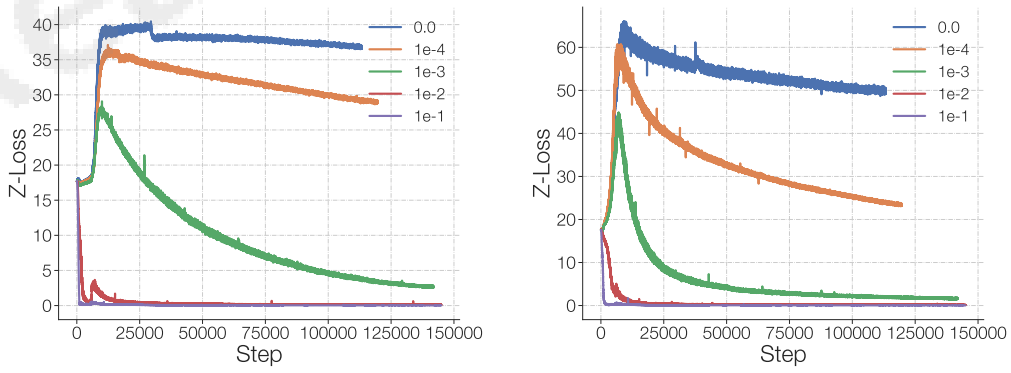


图7：路由器Z-loss的清除损失系数（C Z）。我们在没有路由器Z-loss（蓝色）的预训练的情况下绘制了路由器Z-lesses，并随着C Z值的增加（我们选择与绿色曲线相关的coefientient for line以后的实验）。Z-loss的值为1E-2或更大，接近零的Z-loss收缩。左图显示一个编码器层，右图显示了adecoder层。

[2]A potential source of confusion: $p_i(x)$ is the probability of routing token x to expert i. P_i is the probability fraction to expert i across *all tokens* in the batch $\mathcal{B}$.

c改进的建筑修改

我们在这里考虑一些小型体系结构。第一次修改是在每个莫伊层或之后（称为稀疏致密）之前或之后添加额外的FFFN层（馈送网络，有关更多详细信息，请参见表1）。表16揭示了FFN层在每个稀疏层上的有效性，并且这些额外的FFN层在网络中的其他位置时会帮助更少。确保所有令牌在每个注意力层之间至少都应用了一个FFN似乎有用。

Model	**Neg. Log Perp.** (↑)	Δ
Dense model (baseline)	-1.474	-
Dense model w/ extra FFN layers	-1.452	0.022
Sparse model (baseline)	-1.383	-
Sparse model w/ extra FFN layer *after* each sparse layer	**-1.369**	0.014
Sparse model w/ extra FFN layer *before* each sparse layer	**-1.369**	0.014
Sparse model w/ extra FNN layers placed randomly in the network	-1.376	0.007

表16：在每个稀疏层之前或之后，密集的FFN可提高质量。在每个稀疏层之前或之后，插入一个额外的密度FFN，可以提高质量的2倍，因为将密集层（随机）放置在网络中的其他位置。所有非基线模型剃须刀都添加了相同数量的FFN层，以进行公平比较。请注意，随着模型变得更好，更难改善困境。

其次，我们在专家层中引入了额外的偏见。我们所有的模型都使用gelu-linearffn（Shazeer，2020），而不是relu ffn：ffn relu（x）=（relu（xw 1））w 2ffn geglu（x）=（gelu（xw 11） 2

添加偏置是在FFN层形状的第一个矩阵乘法之后添加的学习重量（B）[批处理，D F F]。乘法偏差（也称为比例参数）是相同形状的学习压力，但要进行元素乘法。我们初始化了添加偏置tozeros和对and的乘法偏置。 11）xW 12）b] w 2

表17显示了我们不同方法的结果。添加剂和乘法偏见都是完全免费的：廉价计算，添加了很少的新参数，并且与模型和专家并行性没有其他额外的沟通成本。从第3.1节使用我们的路由器Z-loss时，Weobserve no Multiverative Bias的不稳定性。我们确实看到，乘法性相互作用造成了绩效，在我们强大的稀疏基线上实现了4%的融合时间加速。这暗示着未来建筑研究的有希望的途径是找到将更多的多重互动添加到网络中的新方法。

Model	**Neg. Log. Perp.** (↑)	Δ
Dense Baseline	-1.474	-
Sparse Baseline	-1.369	-
Sparse + Additive Bias	-1.371	-0.002
Sparse + Multiplicative Bias	**-1.361**	0.008

表17：更多的乘法相互作用提高了稀疏模型质量。添加剂和乘法偏置几乎没有添加参数或计算。

最后，受Roller等人的工作的动机。（2021），我们探索了类似的方法，但在我们的环境中没有发现。我们尝试使用专门使用嵌入一词的路由，以及

嵌入层嵌入路由决策的附加输入。我们切换了停止嵌入单词或允许其从路由器传播的梯度的梯度。仅使用单词嵌入伤害质量，而在使用正常层隐藏激活之外的同时，在预先培训50b+令牌的模型上是1B+密集参数的模型，它具有中性效应。附录J有有关实验的进一步详细信息，结果负面结果。

D批量优先考虑较低容量因素的路由

出乎意料的是，尽管序列上的左至右顺序以象征性的路由为top-1和TOP-2路由在CF小于1.0的情况下很好地奏效。如果将n代币发送给只有m spacesthen n> mbens的专家，则会下降。掉落的顺序很重要：我们将left的令牌置于右（例如，句子早期的令牌将在末端的末端进行路由）。这是Doneto避免模型作弊。如果我们将令牌放在另一个顺序中，则该模型会根据是否删除令牌，以后在序列后来发生什么令牌。

Riquelme等人的批次优先路由（BPR）。（2021）在视觉变压器（Dosovitskiy等，2020）中引入了图像分类。我们的工作通过语言建模的thecontext中的TOP-1路由探索BPR。 BPR的目标是对所有代币进行全球视图，以确定应该删除哪个tokens而不是从左到右的订购。该算法是通过查找Atall n代币将其发送给专家I，然后仅将M概率从路由器带来最高概率的M。表18显示，BPR TOP-1路由在TOP-2路由上提高了性能，尤其是当容量因素小于1.0时。我们将其留在未来的工作中尝试Top-N Bprrouting，这将有望为更高的容量因素带来更大的进步。

重要的是，只能在编码器模型的编码器侧进行BPR路由。在编码器方面，没有自回归预测，所有令牌都可以彼此看到。如果您在解码器上使用bpr，它将学会通过使用未来的令牌信息来提高当前的标记为作弊。

Algorithm	Train CF	Eval CF	Neg. Log. Perp. (↑)
Dense	—	—	-1.474
Dense-L	—	—	-1.384
BPR Top-1	0.5	0.5	-1.433
BPR Top-1	0.5	2.0	-1.416
Top-1	0.75	0.75	-1.428
Top-1	0.75	2.0	-1.404
Top-2	0.75	0.75	-1.424
Top-2	0.75	2.0	-1.402
BPR Top-1	0.75	0.75	-1.409
BPR Top-1	0.75	2.0	-1.397
Top-1	1.0	1.0	-1.397
Top-1	1.0	2.0	-1.384
Top-2	1.0	1.0	-1.392
Top-2	1.0	2.0	-1.378
BPR Top-1	1.0	1.0	-1.386
BPR Top-1	1.0	2.0	-1.379
Top-1	1.25	1.25	-1.378
Top-1	1.25	2.0	-1.373
Top-2	1.25	1.25	-1.375
Top-2	1.25	2.0	-1.369
BPR Top-1	1.25	1.25	-1.376
BPR Top-1	1.25	2.0	-1.375

表18：优先考虑TOP-1路由（BPR）性能。当容量因子≤1时，BPR TOP-1路由的提高。但是，一旦容量因子达到1.25，改进率就大大降低，并且表现不佳。未来的工作可以尝试使用Top-2横幅的BPR，这应该有望进一步提高性能。

e预培训数据集详细信息

用于训练我们稀疏的32B模型的训练前数据集是C4的混合物（Raffel等，2019）和Glam中引入的数据集（Du等，2021）。

Dataset	Tokens (B)	Weight in Mixture
Filtered C4	183	0.17
Filtered Webpages	143	0.34
Wikipedia	3	0.05
Conversations	174	0.23
Forums	247	0.02
Books	390	0.17
News	650	0.02

表19：训练集中的数据和混合物重量。我们来自不同数据集的概率与“混合物中的重量”成正比采样。列出的代币数量为数十亿（b）。有关C4语料库的更多详细信息，请参见Raffel等。（2019年），有关其他数据集，请参见Du等人（2021）。

f完整的微调灵敏度数据

表20包含图6的原始数据，测量了调节协议灵敏度。密集稀疏是与T5-LARGE相匹配的编码器模型flop，该模型已预先训练500kSteps，其批量大小为C4语料库的批量大小为1M令牌。

Model	Learning Rate	Batch Size	Reset Optimizer Slot Vars	SuperGLUE (↑)
Dense	1e-3	1M		84.8
Dense	1e-3	1M	✓	84.3
Dense	5e-4	1M		84.8
Dense	5e-4	1M	✓	84.2
Dense	1e-4	1M		84.0
Dense	1e-4	1M	✓	84.8
Dense	1e-3	262k		84.9
Dense	1e-3	262k	✓	83.7
Dense	5e-4	262k		84.9
Dense	5e-4	262k	✓	84.0
Dense	1e-4	262k		**85.1**
Dense	1e-4	262k	✓	85.0
Dense	1e-3	65k		83.7
Dense	1e-3	65k	✓	82.5
Dense	5e-4	65k		84.4
Dense	5e-4	65k	✓	84.1
Dense	1e-4	65k		84.9
Dense	1e-4	65k	✓	84.6
Sparse	1e-3	1M		**86.9**
Sparse	1e-3	1M	✓	85.9
Sparse	5e-4	1M		86.1
Sparse	5e-4	1M	✓	83.5
Sparse	1e-4	1M		84.3
Sparse	1e-4	1M	✓	84.3
Sparse	1e-3	262k		86.2
Sparse	1e-3	262k	✓	85.2
Sparse	5e-4	262k		85.5
Sparse	5e-4	262k	✓	84.8
Sparse	1e-4	262k		85.1
Sparse	1e-4	262k	✓	85.5
Sparse	1e-3	65k		85.8
Sparse	1e-3	65k	✓	85.5
Sparse	5e-4	65k		86.5
Sparse	5e-4	65k	✓	85.1
Sparse	1e-4	65k		85.6
Sparse	1e-4	65k	✓	84.5

表20：微调协议灵敏度。我们改变了批处理的大小，学习率以及是否适用于密度和稀疏模型的优化器插槽变量。重置优化器陈述细胞调整会损害性能。我们观察到最佳批处理大小和稀疏模型的学习量差异。某些高参数填充设置使稀疏和底性模型的性能几乎完全相同，表明正确调整了流囊的重要性。

g最佳设置路由阈值

TOP-N路由算法

1。将每个令牌X路由至具有最高路由器概率的专家（门1（x））。

2。将每个令牌X的顶级N专家路由器得分标准化，所以GATE i = $\frac{\text{gate}_i(x)}{\sum_n^{(x)}}$.
i = 1门I（x）.3。将令牌路由到其他N-1专家（由i索引），概率最低（1.0, $\frac{\text{gate}_i(x)}{\text{threshold}}$）.
阈值）。阈值是一个预先设置为0.2的预定的超参数。

我们描述了MOE超参数，以及它们应该如何作为路由算法变化。 MOE TOP-2路由算法（Shazeer等，2017; 2018; 2018; Lepikhin et al。，2020）WorksAs works Aws Awstans：首先找到了路由器的专家，该专家被分配了较高的路由器分数（GATE 1），并始终对此表示感谢专家。令牌还发送给其概率明显的第二高专家（1.0，门2 /阈值）。阈值是通常设置为0.2的超参数，而GATE 2是代币的第二高专家路由器概率。请注意，门1和门2通过其两个分数的总和进行标准化，因此它们总和为一个。

我们在此处将TOP-2算法扩展到在此处为顶N路由的工作。以每个令牌为单位的最高nexperts的得分，然后根据该总和将每个专家路由器分数重新归一致。如果第三项授权重新归一化的专家得分的值高于阈值（例如0.2），则令牌将被路由，否则将以概率Scorethreshold进行路由。在很高的水平上，如果他们的分数不比得分最高的专家低得多，这仅将其与下一个N-1专家相连。

对于TOP-3路由与TOP-2，专家得分归一化的总和更大，因此我们通过降低阈值进行了实验。我们的实验结果显示在表21中。从较低的阈值中，我们确实观察到了前3个路由，而theopposite对于TOP-2路由是正确的。

我们还尝试了绝对的阈值策略，而不是相对的策略。仅当其路由器分数大于某些预先确定的值（例如0.2）时，才能将其thenext n-1令牌进行路由。我们发现，如果调谐阈值，它可以达到性能。

Algorithm	Train CF	Threshold	Neg. Log. Perp. (↑)
Dense	—	—	-1.474
Dense-L	—	—	-1.384
Top-2	3.0	0.2	-1.354
Top-2	3.0	0.05	-1.356
Top-3	3.0	0.2	**-1.351**
Top-3	3.0	0.05	**-1.349**

表21：具有不同阈值的TOP-2和TOP-3路由的性能。与TOP-2路由相比，TOP-3路由的阈值较低。

H网格布局用于数据，模型和专家的专家专家

我们使用网格味流的数据和模型并行性分区（Shazeer等，2018）。分区策略是通过第一组形成大小d x m的逻辑2D网格的作用，而行串联到数据维度（d）和列作为模型维度（M）和核心总数等于核心的总数，n = d x x m。该网格只是抽象。每个逻辑固定都映射到物理核心，该核心通过性能调整进行了优化。

作为进修，网格中的每一行都将具有自己独特的数据切片，每列都将拥有模型权重的唯一切片。最终的梯度艾尔德鲁斯通信发生每个单独的列。模型并行性Allreduce通信发生在网格中的each行。这种方法的一个约束是行的数量必须均匀

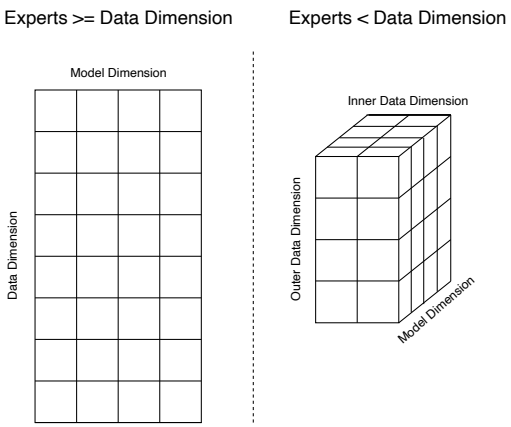


图8：用于分发模型的数据和模型并行网格。在此示例中，总共有32个处理器（例如n = 32）。（左）有效的2D网格，如果专家的数量更大或等于数据并行性维度。数据维度有8行（D），而模量有4列（M）。（右）有效的3D网格当我们拥有的专家少于数据核心尺寸时。批处理维度分为两个新的维度：内部数据和图数据维度。现在，我们每个内部数据维度（i）有1个专家。在外部批处理（O）中，在外部批处理（i）中将8个数据行变成4个，而不是2个专家而不是8个。

将数据序列的数量和列的数量划分，必须均匀地划分为分区的模量。

但是，如果我们的专家少于D，那么这种布局将不起作用。为了使较少的专家在我们的网格中thandata并行性行，我们将数据维度分为两个新的维度：内（i）和外部（o），其中i x o = d，专家的数量等于i。这将形状d x m的逻辑2Dmesh变成了形状o x i x m的3D网格。可视化均值12。

我注意到分布式模型的通信成本

通信操作（Alleduce和All2All）可以显着影响稀疏的模型传递吞吐量（有关通信操作的描述，请参见表1）。 AllreduceCalls沿模型和批处理维度执行，通常由模型尺寸尺寸调用主导，该调用的总和来自工人的部分矩阵乘法操作的总和。当矩阵乘法跨多个核心分配时，需要调用这些调用（例如，模型Parlarallelism）。梯度总和可以通过训练模型施加较大的批次尺寸来摊销，因为梯度积累了alleduce沟通成本是批处理大小的依据。为了减轻较大批量尺寸的内存问题，Microbatches可以使用。微匹配是通过将批次分配到每个均匀分裂的块和每个均匀分裂的块和计算量的方法，然后求和。

为了增加Alleduce吞吐量，可能需要将更多的工人分配给模型DIMension（而不是批处理维度）。但是，增加工人的数量可能会减少每个工人的计算，从而导致较高的通信开销，从而取消了来自Allreduce的较高通信吞吐量。对于本文的结果，首先是对各种模型分配策略进行了筛选。接下来，基于性能基准测试，将前训练作业的形状分配给了AL，该基准显示出Allreduce和All2All中最低的累积通信头。

12有关内部和外部批处理的更多详细信息，请参见网格张量流量：https：//github.com/tensorflow/mesh/blob/ / blob/master/mesh_tensorflow/transflow/transformer/moe.py.py.py.py.py.py.py.py.py.py

J负面结果

我们以一些想法在我们的环境中产生了负面结果。

如果将令牌放在路由器上，则添加信息。我们试验使专家层具有以前的专家仪中是否被路由还是丢弃的信息。我们通过计算在所有审视前的专家层中列出令牌的次数，对每个可能的价值嵌入，然后将其添加到赌场中。我们发现这对性能没有影响。

添加明确的专家位置信息。我们试验将明确的位置信息添加到专家层的输出中。我们想看看在训练开始时，当专家层急剧变化时，它是否改善了性能者加速融合。我们通过添加与每个令牌发送的专家（包括删除令牌的嵌入式）相对应的嵌入来做到这一点，但这并不能提高性能。

将预训练的噪声添加到固定的预训练和调节差异中。为了帮助固定训练的困惑和调节差距，我们尝试了使用各种不同类型的噪声来预先训练稀疏模型。目的是帮助预训练匹配wheredropout的细胞调整条件，并且可以删除更多的令牌。我们尝试在Pre-Training期间尝试添加的一些噪声类型是辍学，辍学了一批令牌，并在路由器中添加了熵合并辅助损失。不幸的是，所有方法要么损害了预训练质量过多，要么最终并没有帮助您进行调整。

在较低的N-1专家中，在顶部N路线中的负载平衡。在标准的Top-N Moe Formoniza-Tion中，只有在A Doken的顶级专家上都有负载平衡。我们用辅助负载量平衡术语对TOP-N路由的其他N-1专家进行了实验，但发现了这种最小的最小值。

混合训练和填充数据以防止过度拟合。为了帮助对在填充过程中的稀疏模型进行过度拟合，我们尝试在培训前的训练跨度损坏数据中混合（例如1%，5%，25%，...）。最终没有帮助精细的表现，但确实增加了培训损失。