神经元组局部学习

Adeetya Patel[1,2] **Michael Eickenberg**[3] **Eugene Belilovsky**[1,2]
[1] 康考迪亚大学 [2] Mila – 魁北克人工智能研究所 [3] Flatiron 研究所{
adeetya.patel, eugene.belilovsky}@concordia.ca,
meickenberg@flatironinstitute.org

arXiv:2301.07635v1 [cs.LG] 18 Jan 2023

摘要

传统的深度网络训练方法会针对所有组件共同优化一个整体目标函数。这可能导致并行化方面的各种低效问题。局部学习是一种模型并行方法，它摒弃了标准的端到端学习设置，并利用局部目标函数，使深度网络中的模型组件能够并行学习。近期研究表明，局部学习的变体能够实现现代深度网络的高效训练。然而，就可分配的计算量而言，这些方法通常受限于网络中的层数。在这项工作中，我们提出研究如何将局部学习应用于层或模块的拆分，将其划分为子组件，从而在局部学习现有的深度级模块化之外，增添宽度级模块化的概念。我们研究了能够使此类模型高效训练的局部学习惩罚项。在 CIFAR-10、CIFAR-100 和 Imagenet32 数据集上的实验表明，引入宽度级模块化能够相较于现有的局部学习方法带来计算优势，并为改进模型并行分布式训练开辟新的机会。代码见：
https://github.com/adeetyapatel12/GN-DGL。

1 引言

神经网络通常采用随机梯度下降并结合反向传播算法进行训练（Krizhevsky 等人，2012；Huang 等人，2019；Simonyan 和 Zisserman，2014；Szegedy 等人，2015）。这种学习算法能够基于全局目标函数，对网络中的所有层和神经元连接进行联合调整。人们通常认为，这种联合调整对于在大规模数据集上取得高性能至关重要，因为围绕总体目标进行联合调整，能够使各层和神经元高效地调整其功能。另一方面，近期有多项研究探讨了使用纯局部损失函数的思路，在这种方法中，各个独立层之间不存在反馈。这些方法以经典的序贯贪心学习过程为基础（Ivakhnenko & Lapa，1965年；Bengio 等人，2007年），但允许不同的模型组件同时学习各自的模型参数。令人惊讶的是，仅依靠这种前向通信，也能够构建高性能模型（Belilovsky 等人，2020年；Nøkland 和 Eidnes，2019年）。这些发现不仅对高性能深度网络的功能特性有影响，也对神经网络的分布式训练具有实际意义。如果逐层学习仍能带来高性能，那么问题就来了：神经网络组件在多大程度上需要进行联合训练?

我们可以从生物神经系统中获得一些启发。生物神经系统依赖极为局部化的突触更新，这通常被建模为赫布学习（Hebb，1949）或其变体。生物神经系统将全局信息纳入学习的主要方式，是通过反馈/循环神经连接，将后期计算得到的信息传回较早阶段的神经元，从而使其能够通过局部学习纳入学习过程。生物系统中的局部学习包括视网膜拓扑图中的空间局部处理，这启发了卷积网络等局部连接架构。此外，生物系统中往往存在多条并行通路，它们执行不同但部分重叠的计算（例如大细胞和小细胞视觉通路）。我们从这一理念中汲取灵感，并将其转化为

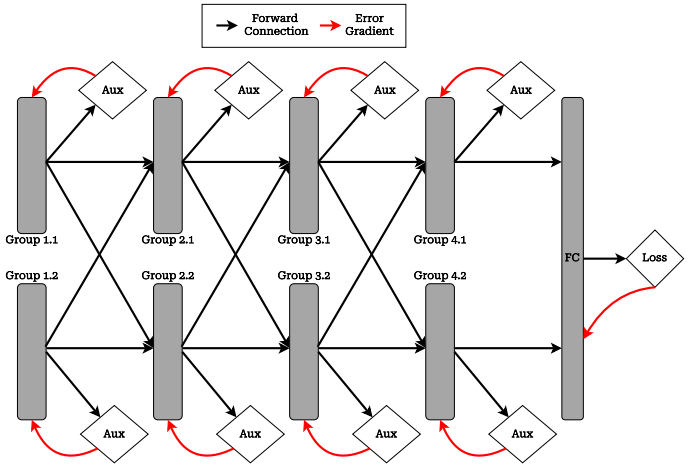

图 1：分组神经元 DGL，每一层和每个神经元组都有一个局部目标。所有局部组均并行学习，并将其输出直接传递给下一层的组。

按通道拆分计算，使不同的组件能够高效地部署在不同的处理单元上。

在这项工作中，我们探究监督式局部损失函数的应用边界，并将其从层级扩展到“神经元组”级别。具体而言，我们考虑这样一种情况：网络层中的互不重叠神经元组各自拥有局部目标函数，并独立优化其参数。研究发现，即使进行这种改动，网络仍能展现出高性能。我们还探究如何仅依靠前向通信，促使这些解耦网络学习多样化的行为（即彼此学习不同的表征）。为展示该方法的实际应用，我们在总训练时间、推理时间和模型性能方面，将所提出的方法与现有局部学习方法进行了比较。

2 相关工作

序贯局部学习通过贪心地逐层添加单独的层，并求解逐层局部优化问题来构建网络；这一方法已在多项经典研究中得到探讨（Ivakhnenko & Lapa，1965年；Fahlman & Lebiere，1989年）。这类方法通常用于简单数据集，尤其是无监督模型（Bengio 等人，2007年；Vincent 等人，2010年）。采用这些方法的主要动机，是深度网络联合优化所面临的困难，例如梯度消失问题。然而，随着用于解决深度网络联合训练问题的现代技术问世，例如改进的初始化、残差连接（He 等人，2016年）和归一化（Ioffe & Szegedy，2015年），这类方法逐渐失宠。Belilovsky 等人（2019b）近期重新审视了这些序贯局部学习技术，并表明该方法能够在大型数据集和各类架构上实现高性能。Belilovsky 等人（2020）；Nøkland 和 Eidnes（2019）进一步证明，这种方法也可用于并行局部学习设置，即同时学习各层，从而改进深度网络的分布式训练。如果允许每层保留内存，甚至可以异步方式进行。Wang 等人（2021）进一步拓展了这一思路，提出了一种新颖的正则化项，以防止表征向监督任务的目标塌缩。Laskin 等人（2021）对多种局部学习技术进行了研究，并在若干常见的大规模任务上将其与互锁反向传播进行了比较。

与我们的工作相关，Veness 等人（2019）研究了一种局部目标，其中每个神经元都要解决一个二分类问题。研究结果在在线学习设置中展示，但尚未扩展到标准的离线学习设置，也未应用于 ImageNet 等复杂数据集。Choromanska 等人（2019）；Lee 等人（2015）；Belilovsky 等人（2020）考虑了目标由全局损失决定的局部目标函数。然而，这些方法需要反馈

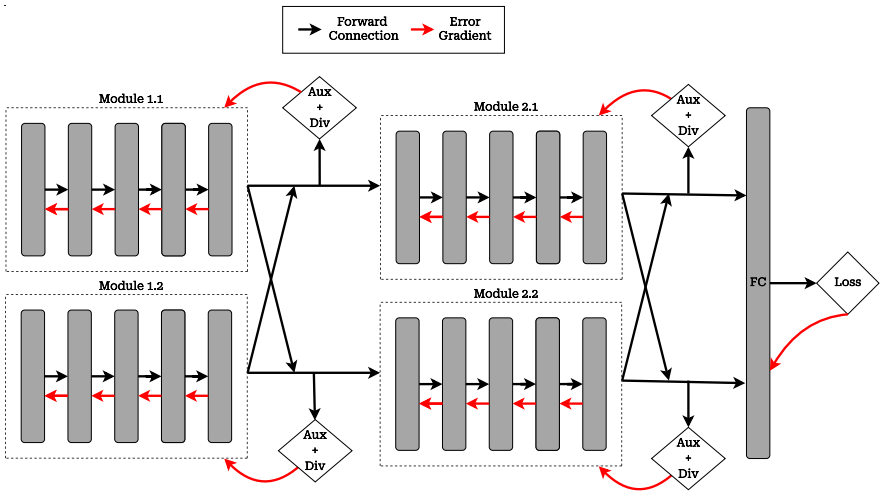

图2：我们方法的示意图。结合使用多层模块、梯度解耦输出和促进多样性的惩罚项。

在各个层和模型组件之间进行通信。另一方面，我们的工作侧重于局部学习目标及其扩展。Jaderberg 等人（2017）还提出使用辅助网络在局部近似梯度，从而实现神经网络层和模块的并行学习。

流水线（Huang 等人，2018）是一种系统级解决方案，可以缓解反向传播引发的问题，这些问题会阻碍不同模块之间的并行训练。然而，这些技术并未消除根本限制（Jaderberg 等人（2017）通常将其描述为锁），因此无法实现完全并行化。

3 方法

3.1 DGL 背景

现在，我们介绍 Belilovsky 等人（2020年）以及 Nøkland 和 Eidnes（2019年）的局部学习框架，并引入相关记号。考虑神经网络的一个输入 x_0，我们将第 j 层的运算记为 $f_{\theta_j}(x_{j-1})$，其中 θ_j 对应网络的参数，而 $\mathcal{L}(y,\ x_j;\ \gamma_j,\ \theta_j)$ 对应于作用于表征 x_j 的局部损失函数，其中 γ_j 是辅助网络的参数。Belilovsky 等人（2020 年）以及 Nøkland 和 Eidnes （2019 年）提出联合并行地学习参数 θ_j。在下一节中，我们将进一步划分目标函数，引入组级局部损失 $\mathcal{L}(y,\ x_j^i;\ \gamma_j^i,\ \theta_j^i)$，其中 x_j^i 是 x_j 的一个子集。

3.2 分组神经元 DGL（GN-DGL）

基于逐层局部学习，我们考虑将每一层进一步拆分为多个组，并为各组设置独立损失。我们提出在线并行学习每个模块（如 Belilovsky 等人（2020年）以及 Nøkland 和 Eidnes（2019年）所采用的方法），而非像 Belilovsky 等（2019b年）那样按顺序学习。具体而言，我们尝试为每个神经元组优化一个贪心目标，如算法 1 所示。这里，每个神经元组包含的神经元数量大致相同。需要注意的是，在前向传播过程中，第 j 层的每个神经元组 i 都会接收来自前一层 $j-1$ 的所有组的输出表示 x_{j-1}^k 作为输入（即第 $j-1$ 层中的所有 k）。图 1 展示了这一过程。我们将这种方法称为分组神经元 DGL（GN-DGL）。请注意，在我们的工作中，最后一个全连接层也与前一层的神经元组解耦。

在 Belilovsky 等人（2019a）中，通过对逐层预测进行加权集成，提升了逐层模型的性能。同样，我们也可以对层和子模块进行集成，以获得最终预测。

3.3 停止梯度分组神经元DGL

我们可以放宽层组之间的通信限制，允许它们将输出发送给彼此的辅助网络。在算法 2 所示的方法中，每个神经元组的辅助网络会收集同一层其他神经元组的梯度解耦输出表示，并将其用作最终预测的一部分。我们注意到，在将输出发送给辅助网络之前进行空间池化，可以降低通信成本。重要的是，这只是一个前向连接：不会向源节点传回任何信息。

算法 1： 分组神经元DGL

输入： 小批次 $\mathcal{S} \triangleq \{(x_0^t,\ y^t)\}_{t \le T}$

初始化 参数 $\{\theta_j^i, \gamma_j^i\}_{j \le J,\ i \le G}$。
对于 $(x_0,\ y) \in \mathcal{S}$ **执行**
  对于 $j \in 1,\ldots,J$，**执行**
    对于 $i \in 1,\ldots,G$，**执行**
      $x_j^i \leftarrow f_{\theta_j^i}(x_{j-1})$。
      计算 $\nabla_{(\gamma^i \quad _j\ \theta_j^i)}\hat{}\mathcal{L}(y, x_j^i;\ \gamma_j^i, \theta_j^i)$.
      (θ_j^i, γ_j^i) ←更新参数 (θ_j^i, γ_j^i)。
    end
    $x_j \leftarrow \|_{i=1}^{G} x_j^i$。
  end
结束

算法 2： 停止梯度 GN-DGL

输入：小批次

$$\mathcal{S} \triangleq \{\ x_0^t\ \ y^t\ \}_{t \le T}$$

(,) 初始

化参数 $\{\theta_j^i, \gamma_j^i\}_{j \le J, i \le G}$**. 2 for (**$x_0$**,** y**)** $\in \mathcal{S}$ **do 3 for** $j \in 1$**, ...,** J **do 4 for** $i \in 1$**, ...,** G **do**
  $x_j^i \leftarrow f_{\theta_j^i}(x_{j-1})$**.**
  $z_j^i = sg[\|_{k \ne i} x_j^k]$**7 计算**$\nabla_{(\gamma_j^i, \theta_j^i)}\hat{}\mathcal{L}$**(**$y$**,** x_j^i**,** z_j^i**;** γ_j^i**,** θ_j^i**). 8 (**θ_j^i**,** γ_j^i**)** ←**更新参数 (**θ_j^i**,** γ_j^i**). 9 end**10
  $x_j \leftarrow \|_{i=1}^{G} x_j^i$**. 11 end 12 end**

3.4 带有促进多样性惩罚项的分组神经元 DGL

局部分组损失的一个预期弱点是，不同组学习到的表征彼此相关且冗余，因而无法最优地聚合预测能力。我们尝试通过引入促进多样性的惩罚项来解决这一问题，该惩罚项几乎不会增加模块间通信的开销。具体而言，我们允许辅助模块彼此传递其 softmax 输出分布。损失函数中的促进多样性项仅基于辅助模块之间 softmax 层输出的通信，在局部鼓励多样性。

对于促进多样性的惩罚项，我们采用了 Dvornik 等人（2019年）研究的一种惩罚项变体。网络中的每个神经元组及其对应的辅助模块分别由 θ_j^i 和 γ_j^i 参数化，其中 i 和 j 分别表示组索引和层索引。每个组

得到类别概率 $p_j^i = \text{softmax}(f_{(\theta_j^i, \gamma_j^i)}(x_{j-1}))$。现在考虑 $\hat{}p_j^i = \frac{p_j^i \odot m_y}{\|p_j^i \odot m_y\|_1}$，其中_____

m_y 是一个向量，在位置 y 取值为 0，其他位置取值为 1。现在，对于层 j中的一组 a，我们按照公式 1 所示，计算其与同一层中每个其余组 b 之间的促进多样性的惩罚项，

$$\phi(\hat{p}_j^a, \hat{p}_j^b) = \texttt{sim}(\hat{p}_j^a, sg[\hat{p}_j^b]) \tag{1}$$

这里， sg 表示停止梯度操作，sim 表示余弦相似度。需要注意的是，为了估计第 a 组与同一层中所有其余组 b 之间的促进多样性的惩罚项，我们使用了$\hat{p}_j^b$的梯度解耦版本。这意味着无需传回梯度信号。该惩罚项通过推动不同组的非目标标签 Softmax 激活值彼此远离来促进多样性。

逐层集成评估 通常，神经网络的预测由网络最后一层的 Softmax 输出给出。由于在这种局部学习范式中，我们使用独立的局部损失函数单独训练每个组，因此也可以聚合不同深度以及不同神经元组的多个预测器。遵循 Belilovsky 等人（2019a）的方法，我们研究每个组在任务上的表现，并评估对这些神经元组进行集成后网络的性能。

考虑每个神经元组生成的 Softmax 层 $\boldsymbol{l}_{ij}$ 输出向量，其中 i 和 j 分别表示神经元组在宽度和深度方向上的索引。为获得最终输出，我们对所有向量取加权平均：沿深度方向逐渐增加权重，并为同一深度的各组分配相近的权重。

$$l_e = \frac{1}{K \times G} \sum_{i=1}^{G} \sum_{j=1}^{K} \alpha_{ij} \boldsymbol{l}_{ij}$$

对于给定层中的各组，我们保持 α_{ij} 不变。对于不同的层，我们根据 Belilovsky 等人（2019a）给出的权重进行调整 j。

4 实验与结果

我们的实验结果首先介绍一组初步的消融实验，随后重点评估模型并行学习设置下的性能权衡。首先，我们研究一个简单的 VGG6a 模型，采用逐层训练以及神经元组级别的局部损失函数。通过这种方式，我们可以确定该方法相较于逐层训练的基本性能特征。在第二组实验中，我们聚焦于模型并行训练的应用，将我们的方法在多层模块情况下与近期方法进行比较，例如解耦贪心学习（DGL）Belilovsky 等人（2020）以及采用信息传播（InfoPro）的局部学习 Wang 等人（2021）。

数据集 我们使用许多先前工作中常用的 CIFAR-10 数据集（Belilovsky 等人，2020；Huo 等人，2018）进行消融实验，以深入了解各个组件的性能。我们在研究模型并行标准权衡的实验中使用 CIFAR-10、CIFAR-100，以及 ImageNet 数据集的一个降采样版本（记作 Imagenet32，Chrabaszcz 等人（2017））。

训练超参数 对于第4.1节中的实验，网络使用 Adam 优化器（Kingma 和 Ba，2014）进行训练，批量大小为 128，共训练 150 轮次。我们将初始学习率设为 0.001，并在第 50 和第 100 轮次分别衰减至 0.0005 和 0.0001，dropout 率为 0.01

对于第4.2节中的实验，我们采用 Wang 等人（2021）中的超参数。对于所有数据集，我们使用带有 0.9 Nesterov 动量的 SGD 优化器训练网络，共训练 160 轮次。我们将初始学习率设为 0.8，并采用余弦学习率退火。批量大小设为 1024，且采用 ℓ_2 权重衰减率 1e-4。

4.1 逐层与分组神经元 DGL

实验中使用的 VGGNet 记为 VGG6a，由六层组成（四个卷积层和两个全连接层），取自 Nøkland 和 Eidnes（2019）。卷积层具有 128 个，

方法	DGL	2组	4组	6组	8组	10组
测试准确率	92.25	91.63	90.55	89.8	89.07	88.91

表 1：VGG6a 各层中的神经元分组。我们观察到，随着组数增加，性能会下降；然而，即使神经元组之间缺乏通信，整体准确率仍能维持在令人惊讶的高水平。需要指出的是，所展示的结果未采用停止梯度技术，也未使用局部多样性促进惩罚项。

方法	DGL	6组 GN-DGL	6组 GN-DGL 多样性	6组 GN-DGL 停止梯度 +多样性
测试准确率	92.2	89.8	90.6	91.9

表 2：消融实验分析局部多样性促进惩罚项的作用以及在 GN-DGL 中使用停止梯度通信的效果。

最后两个全连接层分别有 256、512、512 个通道，以及 8192、1024 个特征。为了报告 DGL 结果，我们按照 Belilovsky 等人（2020）的描述，为每一层使用各自的辅助网络进行训练。对于 GN-DGL 实验，我们进一步将各层沿宽度方向拆分为 2 至 10 个神经元组。每层中的神经元组及其辅助网络均通过辅助损失进行局部训练。关于宽度方向拆分的详细说明将在下文第 4.1 节末尾进一步展开。除最后一个全连接层外，每个神经元组都有自己的辅助损失，如图 1 所示。我们采用与 Nøkland 和 Eidnes（2019）相同的辅助网络设计和损失（称为 *predsim*）。

我们首先考察朴素拆分（不使用停止梯度和多样性的 GN-DGL）。VGG6a 上朴素拆分实验的结果列于表 1。这里，组数的增加以 G 表示，G 是任意一层中的最小组数。我们还报告了 VGG6a 的 DGL（Belilovsky 等人，2020）测试准确率；在该设置中，网络仅沿深度方向划分，每一层本身都是一个局部模块，并配有各自的辅助网络和损失。我们观察到，随着 G 增加，测试准确率略有下降（这是预期之中的）。我们认为，准确率下降主要是由于特征之间存在相关性，这源于各神经元组彼此独立地进行训练。

	VGG6a	**2xVGG6a**	**3xVGG6a**
DGL	92.25	93.27	93.27
6组 GN-DGL	89.8	92.6	93.28

表 3：消融实验探究了宽度对 GN-DGL 的影响。我们观察到，随着宽度增加，GN-DGL 的性能开始接近 DGL，同时并行化程度也有所提高。

特征的多样性与停止梯度 为了提升整体模型的性能，我们允许层内模块之间进行前向通信。具体而言，我们允许在局部模块之间传递 Softmax 激活值和梯度解耦的辅助模块前特征。这对应于第 3.3 节和第 3.4 节讨论的停止梯度和基于多样性的方法。在表 2 中，我们展示了消融实验，结果表明，这些改动持续提升模型性能，使其几乎恢复到 DGL 的性能。

增加宽度 我们还在表 3 中研究了增加宽度的影响。观察发现，增加 6 组 GN-DGL 的宽度可以使其性能更接近 DGL。

VGG6a 的宽度方向拆分 每层中的神经元组所含的神经元数量大致相同。对于 2组配置，我们分别按照 (2,4,2) 的模式拆分前三个卷积层，这意味着第一个卷积层被分为 2 个神经元组，第二个分为 4 个，第三个分为 2 个。此外，我们针对$G \in \{4, 6, 8, 10\}$引入 G-Group 配置，其中各层按照 (2,4,2) 乘以 $G/2$ 的模式进行拆分。也就是说，对于 6组配置，拆分模式为 (6, 12, 6)。

4.2 多层分组神经元 DGL

正如 Belilovsky 等人（2020年）和 Wang 等人（2021年）的研究所发现的那样，完全逐层训练可能会导致性能下降。因此，本节将我们的拆分技术扩展到模块，即由多个层组成的集合，以及子模块：可以组合成模块的子网络。

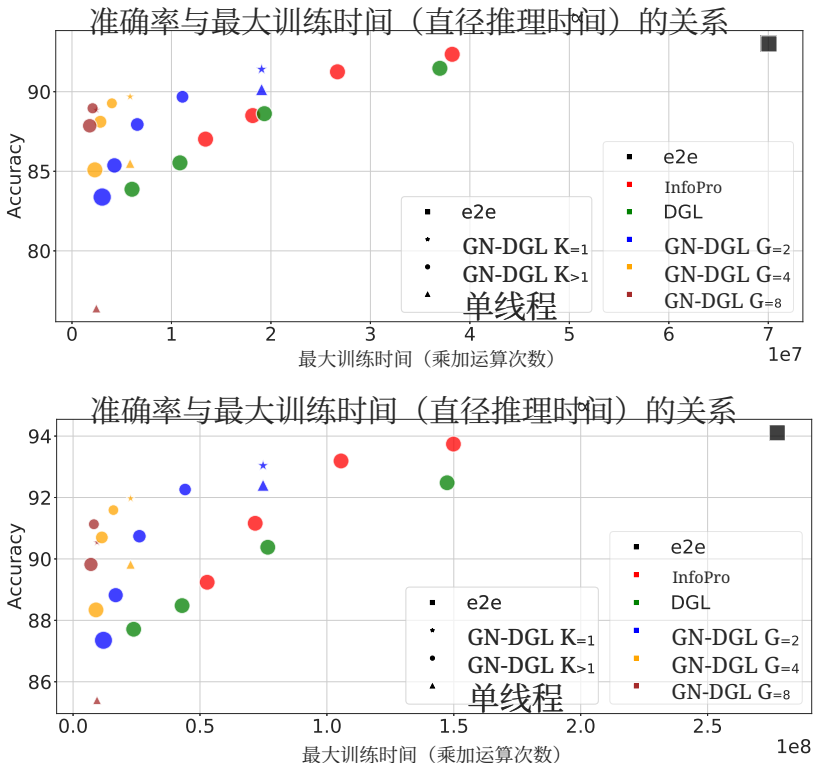

图 3：CIFAR-10 结果。在理论上的分布式场景中，针对各种局部学习方法，给定节点的准确率与最大训练时间（以乘加运算次数计）的关系。我们观察到，对于 ResNet-32（上）和更宽的 ResNet-32x2（下），GN-DGL 在训练时间与准确率之间实现了更好的权衡。气泡大小与模型的推理时间成正比。

与上一节不同，对于多层版本的 GN-DGL，我们将网络划分为 K 个深度方向的本地模块和 G 个宽度方向的本地模块，共计 $K \times G$ 个本地模块。这里， K 对应于 Wang 等人（2021年）使用的深度方向拆分；我们的这组实验基于他们的实验设置。每个本地模块包含的层数以及每层的通道数大致相同。如图2所示，各模块彼此独立地训练。值得注意的是，在卷积神经网络中进行多层拆分时，整个网络的推理时间会缩短（交叉连接减少，计算量也随之降低）。这为应用 GN-DGL 带来了额外的优势。

基线方法讨论 在此，我们将所提出的方法与其他局部学习方法进行比较，例如解耦贪心学习（*DGL*）Belilovsky 等人（2020）提出的方法，以及采用信息传播进行局部学习的 InfoPro（*InfoPro*）Wang 等人（2021）提出的方法。其他局部学习方法仅沿深度方向拆分网络，而我们的方法同时沿深度和宽度方向拆分网络。沿这两个方向拆分可在性能和计算速度方面带来多项优势。为便于比较，我们将其他局部学习方法视为 G = 1 的特例。值得注意的是，InfoPro 方法存在一项低效之处：其解码器模型会尝试在每一层重建网络输入，导致较深层的计算成本较高（因为需要对大量通道进行空间上采样）。此外，我们还提供一个参考结果：将模型按宽度拆分为 G 并进行端到端训练。我们将此基线称为 单线程。我们还报告了对应于 G 和 K=1 的模型性能，作为一种基线：独立训练各个子网络，再通过全连接层将其重新组合。最后，我们将其与端到端训练（$e2e$）进行比较。

指标 在我们的设置中，我们考虑三个比较因素：总训练时间、推理时间和最终性能，因为这些因素之间存在多种权衡。为了公平地考量

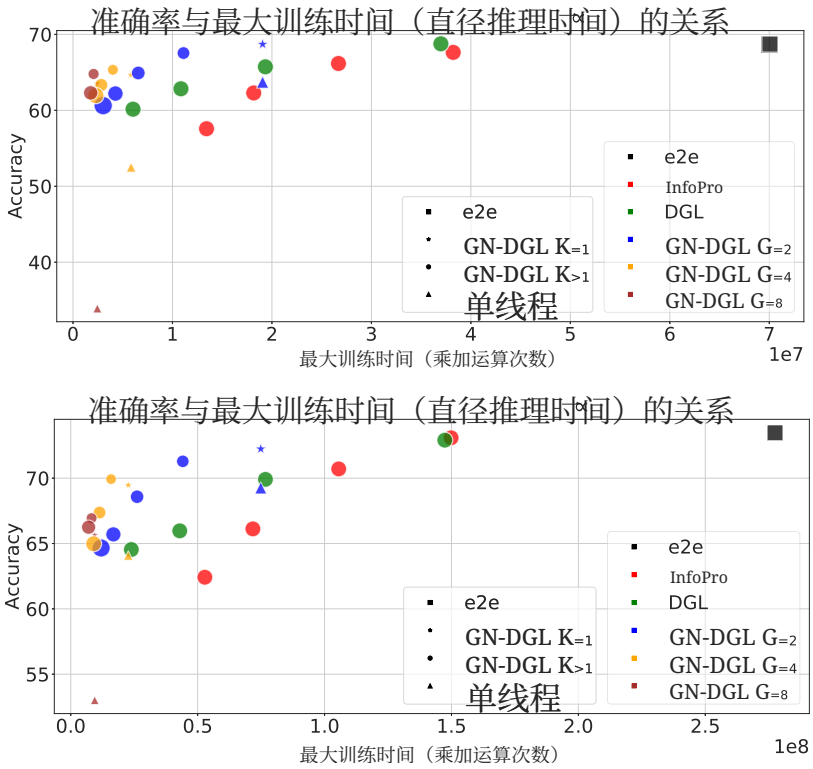

图 4：CIFAR-100 结果。在分布式场景中，对于各种局部学习方法，给定节点的准确率与最大训练时间（以乘加运算次数计）的关系。我们观察到，对于 ResNet-32（上图）和更宽的 ResNet-32x2（下图），GN-DGL 在训练时间与准确率之间实现了更优的权衡，其趋势相较于 DGL 和 InfoPro 有了显著改善。

训练时间，我们以乘加运算次数（MACs）衡量某种方法中任一子组件的最大计算量。

结果讨论 图 3、图 4 和图 5 分别展示了 CIFAR-10、CIFAR-100 和 Imagenet-32 数据集上的结果。我们考察了 ResNet-32 和更宽的 ResNet-32x2 的结果。可以观察到，与采用 ResNet-32 的 DGL 和 InfoPro 相比，我们的 GN-DGL 方法在多种配置下、许多情况下，在性能、总训练时间和推理速度方面实现了更优的权衡。对于更复杂的 CIFAR-100 和 Imagenet32 数据集，我们观察到，采用 G>1 的情况下，这些权衡得到了更显著的改善，其趋势明显优于 DGL 或 InfoPro。

单线程的朴素基线在训练时间与准确率的权衡方面表现很差。我们注意到，简单的 K=1方法在较宽的模型上表现尚可，但随着 G 增大，性能会下降；如图 4 所示，在 CIFAR-100 上尤为明显。

InfoPro 在 CIFAR-10 上的表现优于 DGL，但在 CIFAR-100 上并未带来显著提升。此外，我们未能在 Imagenet32 上通过 InfoPro 获得令人满意的性能：要么运行速度过慢，要么准确率较低。我们推测这是因为它需要一个深层卷积解码器模型，从而导致速度大幅下降。我们还观察到，与 DGL 和 GN-DGL 方法相比，InfoPro 对超参数更加敏感。

4.3 逐层集成的消融实验

不同于传统方法仅使用最后一层的输出进行最终预测，我们按照 Belilovsky 等人（2019a）的描述，对辅助模块的输出进行集成。这种方法称为逐层集成，将在第 3.4 节中详细讨论。表 4 展示了比较逐层集成与最后一层评估有效性的消融研究。我们的

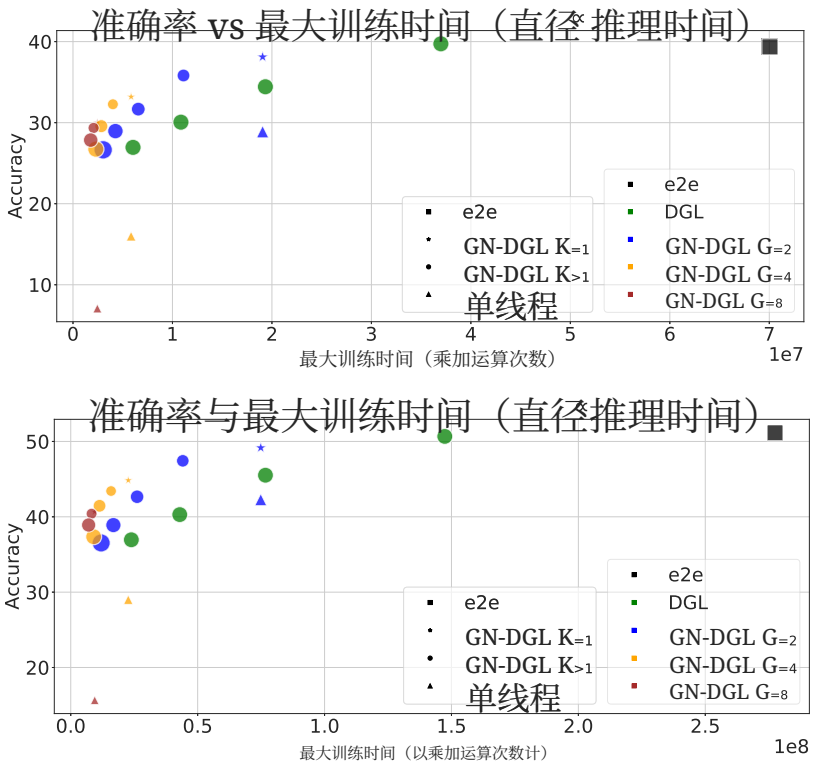

图 5：Imagenet32 结果。在理论分布式场景中，对于各种局部学习方法，给定节点的准确率与最大训练时间（以乘加运算次数计）的关系。我们观察到，对于 ResNet-32（上）和更宽的 ResNet-32x2（下），GN-DGL 在这个复杂数据集上都实现了更优的权衡。由于 InfoPro 在该数据集上的表现很差，因此未予展示。

评估方法	G=2				G=4			G=8	
	K=2	K=4	K=8	K=16	K=2	K=4	K=8	K=2	K=4
最后一层	89.45	87.52	84.73	82.84	89.01	88.25	84.64	88.67	87.11
逐层集成	89.68	87.94	85.37	83.38	89.27	88.11	85.09	88.96	87.86

表 4：在使用 ResNet-32 的 CIFAR-10 上，消融 GN-DGL 中逐层集成的影响。我们观察到，与最后一层评估相比，GN-DGL 的逐层集成能够带来更好的性能。对于给定的 G，随着 K 的增加，性能增益逐步提高。

结果表明，逐层集成始终能够提升性能，且不会增加额外成本。例如，在 CIFAR-10 上对 GN-DGL (K=4, G=8) 进行测试时，采用逐层集成的测试准确率为 87.86%，而采用最后一层评估时为 87.11%。

5 结论

我们研究了在神经网络的多个层中训练相互独立的局部神经元组。具体而言，我们研究了一项常见的图像分类任务，重点关注卷积神经网络。结果表明，这种方法能够提升局部学习方法的并行化程度，从而在给定训练和推理时间的情况下，大幅提高准确率。未来的研究可以探索更高效的辅助模型，并将该方法扩展到更多任务。

致谢

本研究由 Concordia 的内部经费资助。我们感谢 ComputeCanada 和 Calcul Quebec 提供资源。

参考文献

Belilovsky, E., Eickenberg, M. 和 Oyallon, E. 贪心逐层学习可扩展至 ImageNet。载于国际机器学习会议，第 583–593 页。PMLR，2019a。Belilovsky, E., Eickenberg, M. 和 Oyallon, E. 贪心逐层学习可扩展至 ImageNet。第 *36* 届国际机器学习会议论文集（*ICML*），2019b年。Belilovsky, E., Eickenberg, M. 和 Oyallon, E. 卷积神经网络的解耦贪心学习。载于 III, H. D. 和 Singh, A.（编），第 *37* 届国际机器学习会议论文集，第 119 卷，机器学习研究论文集，第 736–745 页。PMLR，2020年7月13–18日。网址：https://proceedings.mlr.press/v119/belilovsky20a.html。Bengio, Y., Lamblin, P., Popovici, D. 和 Larochelle, H. 深度网络的贪心逐层训练。载于神经信息处理系统进展，第 153–160 页，2007年。Choromanska, A., Cowen, B., Kumaravel, S., Luss, R., Rigotti, M., Rish, I., Diachille, P., Gurev, V., Kingsbury, B., Tejwani, R. 等。超越反向传播：使用辅助变量进行在线交替最小化。载于国际机器学习会议，第 1193–1202 页。PMLR，2019年。Chrabaszcz, P., Loshchilov, I. 和 Hutter, F. 作为 CIFAR 数据集替代方案的下采样版 ImageNet。*arXiv* 预印本 *arXiv:1707.08819*，2017年。Dvornik, N., Schmid, C. 和 Mairal, J. 多样性与协作：用于小样本分类的集成方法。载于*IEEE/CVF* 国际计算机视觉会议论文集，第 3723–3731 页，2019年。Fahlman, S. 和 Lebiere, C. 级联相关学习架构。神经信息处理系统进展，第 2 卷，1989年。He, K., Zhang, X., Ren, S. 和 Sun, J. 用于图像识别的深度残差学习。载于*IEEE* 计算机视觉与模式识别会议论文集，第 770–778 页，2016年。Hebb, D.《行为的组织》。emph纽约，1949年。Huang, G., Liu, Z., Pleiss, G., Van Der Maaten, L. 和 Weinberger, K. 具有密集连接的卷积网络。*IEEE* 模式分析与机器智能汇刊，2019年。Huang, Y., Cheng, Y., Chen, D., Lee, H., Ngiam, J., Le, Q. V. 和 Chen, Z. GPipe：利用流水线并行高效训练巨型神经网络。*arXiv* 预印本 *arXiv:1811.06965*，2018年。Huo, Z., Gu, B. 和 Huang, H. 使用特征重放训练神经网络。神经信息处理系统进展，2018年。Ioffe, S. 和 Szegedy, C. 批归一化：通过减少内部协变量偏移加速深度网络训练。载于国际机器学习会议，第 448–456 页。PMLR，2015年。Ivakhnenko, A. G. 和 Lapa, V. G. 控制论预测装置。*CCM Information Corporation*。，1965年。Jaderberg, M., Czarnecki, W. M., Osindero, S., Vinyals, O., Graves, A., Silver, D. 和 Kavukcuoglu, K. 使用合成梯度的解耦神经接口。国际机器学习会议，2017年。Kingma, D. P. 和 Ba, J. Adam：一种随机优化方法。*arXiv* 预印本 *arXiv:1412.6980*，2014年。Krizhevsky, A., Sutskever, I. 和 Hinton, G. E. 使用深度卷积神经网络进行 ImageNet 分类。神经信息处理系统进展，第 25 卷，2012年。Laskin, M., Metz, L., Nabarro, S., Saroufim, M., Noune, B., Luschi, C., Sohl-Dickstein, J. 和 Abbeel, P. 使用局部更新并行训练深度网络，2021年。网址：https: //openreview.net/forum?id=ufS1zWbRCEa。

Lee, D.-H., Zhang, S., Fischer, A. 和 Bengio, Y. 差分目标传播。载于机器学习与数据库知识发现联合欧洲会议，第 498–515 页。Springer，2015年。Nøkland, A. 和 Eidnes, L. H. 使用局部误差信号训练神经网络。*arXiv* 预印本 *arXiv:1901.06656*，2019年。Simonyan, K. 和 Zisserman, A. 用于大规模图像识别的超深卷积网络。*arXiv* 预印本 *arXiv:1409.1556*，2014年。Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. 和 Rabinovich, A. 通过卷积更深入。载于*IEEE* 计算机视觉与模式识别会议论文集，第 1–9 页，2015年。Veness, J., Lattimore, T., Budden, D., Bhoopchand, A., Mattern, C., Grabska-Barwinska, A., Sezener, E., Wang, J., Toth, P., Schmitt, S. 等。门控线性网络。*arXiv* 预印本 *arXiv:1910.01526*，2019年。Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A. 和 Bottou, L. 堆叠去噪自编码器：在具有局部去噪准则的深度网络中学习有用的表征。机器学习研究期刊，11(12)，2010年。Wang, Y., Ni, Z., Song, S., Yang, L. 和 Huang, G. 重新审视局部监督学习：端到端训练的另一种选择。载于国际学习表征会议，2021年。URL https://openreview.net/forum?id=fAbkE6ant2。