神经语言模型的缩放定律

摘要

我们研究了语言模型在交叉熵损失上的经验缩放定律。损失随着模型规模、数据集规模以及用于训练的计算量呈幂律缩放，这种趋势跨越了七个数量级以上。其他架构细节（如网络的宽度或深度）在较大范围内对结果影响很小。我们还用简单的公式描述了过拟合与模型/数据集规模的关系，以及训练速度与模型规模的关系。这些关系使我们能够确定在固定计算预算下的最优分配方式。更大的模型在样本利用效率上显著更高，因此最优的计算效率训练方式是：用相对适中的数据量训练非常大的模型，并且在收敛前就提前停止训练。

目录

1 引言

语言为人工智能研究提供了一个天然的领域，因为绝大多数推理任务都可以通过语言高效地表达和评估，而世界上的文本数据也为通过生成式建模进行无监督学习提供了丰富的资源。近年来，深度学习在语言建模方面取得了快速进展，最先进的模型（RNSS18, DCLT18, YDY19, LOG19, RSR19）在许多具体任务上已接近人类水平（WPN19），包括生成连贯的多段落提示文本样本（RWC19）。

人们可能会认为，语言建模的性能取决于模型架构、神经网络的规模、用于训练的计算能力以及可用于训练的数据量。在本研究中，我们将通过实证方法考察语言建模损失与这些因素之间的关系，重点关注 Transformer 架构（VSP17, LSP18）。语言任务的性能上限很高、下限很低，使我们能够研究跨越七个数量级以上的规模趋势。

在整个研究过程中，我们都观察到：模型性能作为训练时间、上下文长度、数据集规模、模型规模和计算预算的函数，呈现出精确的幂律缩放关系。

1.1 总结

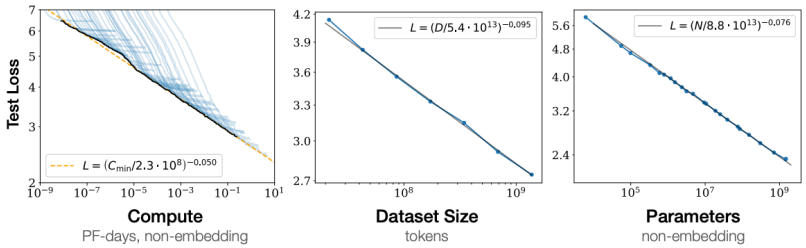

图 1: 随着模型规模、数据集规模和用于训练的计算量的增加，语言建模性能会平滑提升。要获得最优性能，三者必须同步扩大。实证结果显示，当另外两个因素不成为瓶颈时，性能与每一个单独因素都呈幂律关系。

我们对 Transformer 语言模型的主要发现如下：

- **性能主要依赖于规模，弱依赖于模型形状：**模型性能最主要取决于规模，包括三个因素：模型参数数量 N（不含嵌入层）、数据集大小 D、以及用于训练的计算量 C。在合理范围内，性能对其他结构超参数（如深度与宽度的比例）依赖很弱。（第 3 节）
- **平滑的幂律关系：**当不受另外两个因素限制时，性能与 N、D、C 这三个规模因子分别呈现幂律关系，这一趋势跨越了六个数量级以上（见图 1）。在高端没有观察到偏离这一趋势的迹象，尽管在损失趋近于零前，性能最终会趋于平稳。（第 3 节）
- **过拟合的普适性：**只要同步扩大 N 和 D，性能会可预测地提升；但如果 N 或 D 其中之一保持不变，另一个增加，则会进入收益递减区间。性能损失可由 $N^{0.74}/D$ 的比值预测，也就是说，每当模型规模增加 8 倍，只需将数据量增加约 5 倍即可避免性能损失。（第 4 节）
- **训练过程的普适性：**训练曲线遵循可预测的幂律，其参数与模型规模基本无关。通过外推训练曲线的前半段，可以大致预测如果训练更久会达到的损失水平。（第 5 节）
- **迁移能力随测试性能提升：**当我们在与训练分布不同的文本上评估模型时，结果与训练验证集上的表现高度相关，损失有一个大致恒定的偏移量——也就是说，迁移到新分布会带来一个固定的损失惩罚，但整体提升趋势与训练集一致。（第 3.2.2 节）
- **样本效率：**大模型比小模型更具样本效率，能用更少的优化步数（见图 2）和更少的数据点（见图 4）达到同等性能。
- **收敛效率低下：**在固定计算预算 C 下，如果对模型规模 N 和可用数据 D 没有限制，最优性能是在训练非常大的模型并在远未收敛时提前停止（见图 3）。最优的计算效率训练比用小模型训练到收敛要高得多，数据需求随 $D \sim C^{0.27}$ 增长非常缓慢。（第 6 节）
- **最优批量大小:**这些模型的理想批量大小大致只与损失的幂有关,并可通过测量梯度噪声尺度(MKAT18)确定；对于我们能训练的最大模型，收敛时的批量大小约为 100 万到 200 万个 token。（第 5.1 节）

综上，这些结果表明，只要合理扩大模型规模、数据和计算量，语言建模性能会平滑且可预测地提升。我们预计更大的语言模型将比当前模型表现更好且更具样本效率。

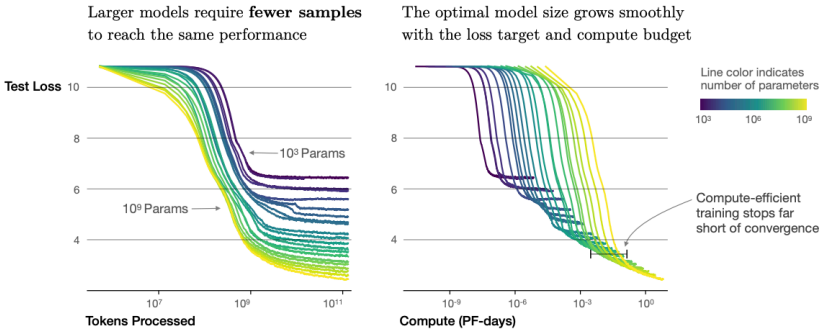

图 2: 展示了一系列语言模型的训练过程，模型规模从 10^3 到 10^9 个参数（不包括嵌入层）。

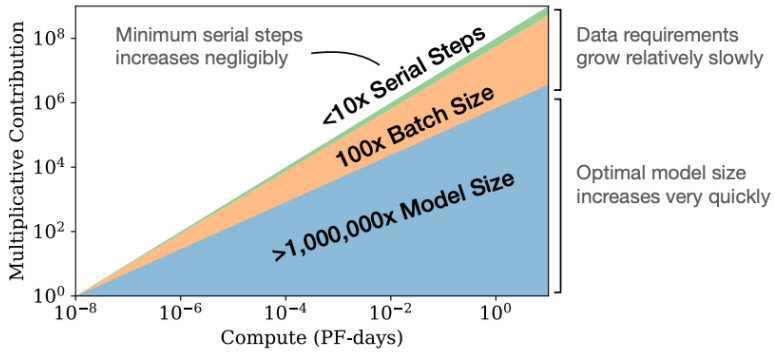

图 3: 随着可用计算量的增加，我们可以选择将多少计算资源分配给更大的模型、更大的批量和更长的训练步数。此图以计算量增加十亿倍为例进行说明。对于最优的计算效率训练，大部分计算资源应用于增加模型规模。只需相对较小的数据增量即可避免数据重复。在数据增量中，大部分可用于通过更大的批量提升并行度，只需极小的串行训练时间增长。

1.2 缩放定律总结

经过自回归语言建模训练的 Transformer，其测试损失在仅受非嵌入参数数量 N、数据集大小 D 或最优分配的计算预算 $C_{\min}$ 限制时，可以用幂律关系进行预测（见图 1）：

1. 对于参数数量有限、在足够大数据集上训练至收敛的模型：

$$L(N) = \left(\frac{N_c}{N}\right)^{\alpha_N}, \quad \alpha_N \sim 0.076, \quad N_c \sim 8.8 \times 10^{13} \text{ (non-embedding parameters)} \tag{1.1}$$

2. 对于使用有限数据集并提前停止训练的大型模型：

$$L(D) = \left(\frac{D_c}{D}\right)^{\alpha_D}, \quad \alpha_D \sim 0.095, \quad D_c \sim 5.4 \times 10^{13} \text{ (tokens)} \tag{1.2}$$

3. 当在有限计算量、足够大的数据集、最优规模的模型和足够小的批量下（即最优利用计算资源）进行训练时：

$$L(C_{\min}) = \left(\frac{C_c^{\min}}{C_{\min}}\right)^{\alpha_C^{\min}}, \quad \alpha_C^{\min} \sim 0.050, \quad C_c^{\min} \sim 3.1 \times 10^8 \text{ (PF-days)} \tag{1.3}$$

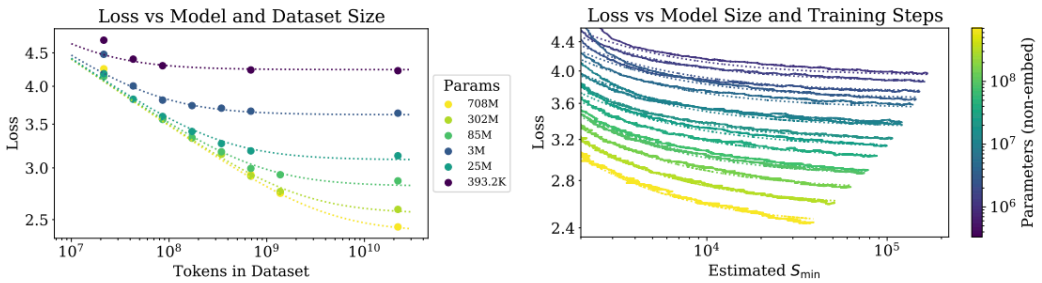

图 4: 左：提前停止时的测试损失 $L(N, D)$ 随数据集规模 D 和模型规模 N 的变化可由公式 (1.5) 预测。右：经过初始过渡期后，所有模型规模 N 的学习曲线都可以用公式 (1.6) 拟合，该公式以 $S_{\min}$（在大 batch size 下训练时的步数）为参数（详见第 5.1 节）。

我们还观察到，在固定批量大小下训练时，训练计算量 C 也呈现经验幂律趋势（见图 1），但用于预测应采用 $C_{\min}$ 的趋势，两者由公式 (5.5) 相关联。这些关系在 $C_{\min}$ 上跨越八个数量级，在 N 上跨越六个数量级，在 D 上跨越两个数量级以上。它们对模型形状和其他 Transformer 超参数（如深度、宽度、自注意力头数）的依赖性极弱，具体数值与 Webtext2 训练集相关。这些幂律指数 $\alpha_N, \alpha_D, \alpha_C^{\min}$ 指定了随着 N、D 或 $C_{\min}$ 增大时性能提升的幅度。例如，参数数量翻倍，损失会减少到 $2^{-\alpha_N} = 0.95$。$N_c, C_c^{\min}, D_c$ 的具体数值依赖于词表大小和分词方式，因此没有本质意义。

临界批量大小 B_{crit}（决定数据并行时速度/效率权衡，见 MKAT18）也大致服从关于损失 L 的幂律关系：

$$B_{\text{crit}}(L) = B^* L^{1/\alpha_B}, \quad B^* \sim 2 \times 10^8 \text{ (tokens)}, \quad \alpha_B \sim 0.21 \tag{1.4}$$

公式 (1.1) 和 (1.2) 共同表明，随着模型规模的增加，数据集规模应亚线性增长，即 $D \propto N^{\alpha_N/\alpha_D} \sim N^{0.74}$。实际上，我们发现有一个公式可以同时描述 N 和 D 的依赖关系，并刻画过拟合程度：

$$L(N, D) = \left[\left(\frac{N_c}{N} \right)^{\alpha_N/\alpha_D} + \frac{D_c}{D} \right]^{\alpha_D} \tag{1.5}$$

其拟合结果见图 4 左。我们猜测这种函数形式也适用于其他生成建模任务的 log-likelihood。

当在无限数据下，对给定模型训练有限步数 S 时，经过初始过渡期后，学习曲线可准确拟合为（见图 4 右）：

$$L(N, S) = \left(\frac{N_c}{N} \right)^{\alpha_N} + \left(\frac{S_c}{S_{\min}(S)} \right)^{\alpha_S} \tag{1.6}$$

其中 $S_c \approx 2.1 \times 10^3$，$\alpha_S \approx 0.76$，$S_{\min}(S)$ 是用公式 (5.4) 估算的达到给定损失的最小优化步数。

在固定计算预算 C 下训练时，公式 (1.7) 预测最优模型规模 N、最优批量大小 B、最优步数 S 和数据集规模 D 应该按如下方式增长：

$$N \propto C^{\alpha_C^{\min}/\alpha_N}, \quad B \propto C^{\alpha_C^{\min}/\alpha_B}, \quad S \propto C^{\alpha_C^{\min}/\alpha_S}, \quad D = B \cdot S \tag{1.7}$$

其中

$$\alpha_C^{\min} = 1/\left(1/\alpha_S + 1/\alpha_B + 1/\alpha_N\right) \tag{1.8}$$

这与经验最优结果 $N \propto C_{\min}^{0.73}$，$B \propto C_{\min}^{0.24}$，$S \propto C_{\min}^{0.03}$ 非常吻合。随着计算预算 C 的增加，主要应投入到更大的模型上，而不需要大幅增加训练时间或数据集规模（见图 3）。这也意味着，随着模型变大，其样本效率会不断提升。实际上，由于硬件限制，研究者通常会用小模型训练更久，而不是采用最优的计算效率。最优性能依然遵循关于总计算量的幂律关系（见公式 (1.3)）。

我们还为公式 (1.6) 提供了基本理论动机，分析了学习曲线拟合及其对训练时间的影响，并对每个 token 的结果进行了细致分解。同时，我们也简要对比了 LSTM 和循环 Transformer（DGV18）。

1.3 符号说明

我们使用如下符号：

- L ——以 nats 为单位的 cross entropy loss。通常对上下文中的 token 取平均，但有时也报告特定 token 的损失。
- N ——模型参数数量，不包括所有 vocabulary 和 positional embeddings。
- $C \approx 6NBS$ ——训练总计算量的估算（不含 embedding），其中 B 为 batch size，S 为训练步数（即参数更新次数）。数值以 PF-days 为单位（1 PF-day $= 10^{15} \times 24 \times 3600 = 8.64 \times 10^{19}$ 浮点运算）。
- D ——数据集规模（tokens 数）。
- B_{crit} ——临界 batch size（MKAT18），定义和讨论见第 5.1 节。在临界 batch size 下训练可在时间和计算效率间取得大致最优平衡。
- C_{min} ——达到给定损失所需的最小训练计算量（不含 embedding）。即在 batch size 远小于临界 batch size 时的训练计算量。
- S_{min} ——达到给定损失所需的最小训练步数。即在 batch size 远大于临界 batch size 时的训练步数。
- α_X ——损失关于 X 的幂律指数，$L(X) \propto 1/X^{\alpha_X}$，其中 X 可以是 $N, D, C, S, B, C_{\text{min}}$ 等。

2 背景与方法

我们在 WebText2 上训练语言模型，WebText2 是 WebText（RWC+19）数据集的扩展版本，使用 byte-pair encoding（SHB15）分词，词表大小 $n_{vocab} = 50257$。我们优化自回归 log-likelihood（即 cross-entropy loss），在 1024-token 的上下文上取平均，这也是我们的主要性能指标。我们记录了模型在 WebText2 测试集上的损失，以及在其他若干文本分布上的损失。我们主要训练 decoder-only（LSP+18, RNSS18）Transformer（VSP+17）模型，同时也训练了 LSTM 和 Universal Transformer（DGV+18）模型用于对比。

2.1 Transformer 的参数与计算量缩放

我们用超参数 n_{layer}（层数）、d_{model}（残差流维度）、d_{ff}（前馈层中间维度）、d_{attn}（注意力输出维度）、n_{heads}（每层注意力头数）来参数化 Transformer 架构。输入上下文包含 n_{ctx} 个 token，除特殊说明外 $n_{ctx} = 1024$。

我们用 N 表示模型规模，定义为非嵌入参数的数量：

$$N \approx 2d_{model}n_{layer}(2d_{attn} + d_{ff}) = 12n_{layer}d_{model}^2 \tag{2.1}$$

其中采用了标准设定 $d_{attn} = d_{ff}/4 = d_{model}$，并省略了 bias 和其他次要项。我们的模型还包含 $n_{vocab}d_{model}$ 个参数的 embedding 矩阵，以及 $n_{ctx}d_{model}$ 个参数的 positional embedding，但在讨论“模型规模” N 时不计入这些参数；这样可以得到更清晰的缩放定律。

Transformer 的一次前向传播大约涉及

$$C_{forward} \approx 2N + 2n_{layer}n_{ctx}d_{model} \tag{2.2}$$

次 add-multiply 运算，其中系数 2 来自矩阵乘法中的 multiply-accumulate 操作。更详细的每步参数和计算量见表 1。

Operation	Parameters	FLOPs per Token
Embed	$(n_{vocab} + n_{ctx})d_{model}$	$4d_{model}$
Attention: QKV	$n_{layer}d^3_{model}d_{attn}$	$2n_{layer}d^3_{model}d_{attn}$
Attention: Mask	—	$2n_{layer}n_{ctx}d_{attn}$
Attention: Project	$n_{layer}d_{attn}d_{model}$	$2n_{layer}d_{attn}d_{embd}$
Feedforward	$n_{layer}2d_{model}d_{ff}$	$2n_{layer}2d_{model}d_{ff}$
De-embed	—	$2d_{model}n_{vocab}$
Total (Non-Embedding)	$N = 2d_{model}n_{layer}(2d_{attn} + d_{ff})$	$C_{forward} = 2N + 2n_{layer}n_{ctx}d_{attn}$

表 1: Transformer 模型的参数量和前向传播计算量估算。省略了非线性、bias 和 layer normalization 等次要项。

对于 $d_{model} > n_{ctx}/12$ 的上下文和模型，每个 token 的 context-dependent 计算量在总计算中占比较小。由于我们主要研究 $d_{model} \gg n_{ctx}/12$ 的模型，因此在训练计算量估算中不计入 context-dependent 项。考虑到反向传播大约是前向传播计算量的两倍，我们将非嵌入计算量估算为 $C \approx 6N$ 浮点运算每训练 token。

2.2 训练过程

除非另有说明，我们使用 Adam 优化器（KB14）训练模型，训练步数为 2.5×10^5，batch size 为 512 个长度为 1024 的序列。由于内存限制，参数量超过 1B 的最大模型采用 Adafactor（SS18）优化器。我们尝试了多种学习率和调度方式，详见附录 D.6。我们发现收敛时的结果基本不受学习率调度影响。除非另有说明，所有训练均采用 3000 步线性 warmup，随后余弦衰减至零的学习率调度。

2.3 数据集

我们在 [RWC+19] 描述的 WebText 数据集扩展版上训练模型。原始 WebText 数据集是对 Reddit 截至 2017 年 12 月所有获得至少 3 karma 的外链进行网页抓取。WebText2 在此基础上，增加了 2018 年 1 月至 10 月期间同样满足 3 karma 的 Reddit 外链。karma 阈值作为判断链接是否有趣或有用的启发式标准。新链接的文本内容通过 Newspaper3k Python 库提取。最终数据集包含 2030 万个文档，96 GB 文本，1.62×10^{10} 个单词（按 wc 定义）。我们随后应用 [RWC+19] 描述的可逆分词器，得到 2.29×10^{10} 个 tokens。我们保留 6.6×10^8 个 tokens 作为测试集，并在 Books Corpus（ZKZ+15）、Common Crawl（Fou）、英文 Wikipedia 及一组公开互联网书籍的样本上进行测试。

3 实证结果与基本幂律

为了刻画语言模型的缩放特性，我们训练了多种模型，主要变化因素包括：

- 模型规模（non-embedding parameters 从 768 到 15 亿不等）
- 数据集规模（从 2200 万到 230 亿 tokens）
- 结构（包括深度、宽度、注意力头数和前馈层维度）
- 上下文长度（大多数实验为 1024，也有部分实验采用更短的上下文）
- 批量大小（大多数实验为 2^{19}，但我们也对其进行调整以测量临界批量大小）

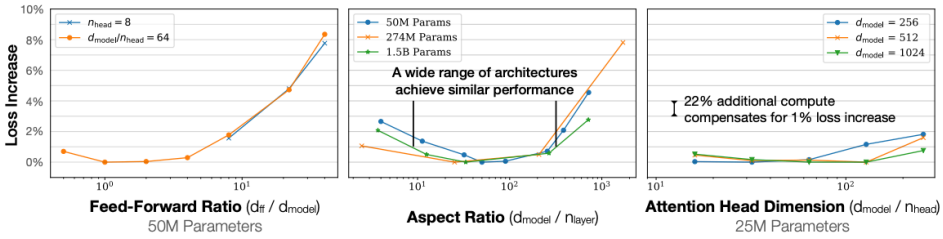

图 5: 当 non-embedding 参数总数 N 固定时，性能对模型结构的依赖非常弱。损失在较大范围的结构变化下仅有百分之几的波动。参数数量的微小差异通过以 $L(N)$ 拟合为基线进行补偿。尤其是纵横比可以变化 40 倍，性能仅有轻微影响；例如 $(n_{layer}, d_{model}) = (6, 4288)$ 的模型损失仅比 [RWC+19] 中 $(48, 1600)$ 的模型高 3%。

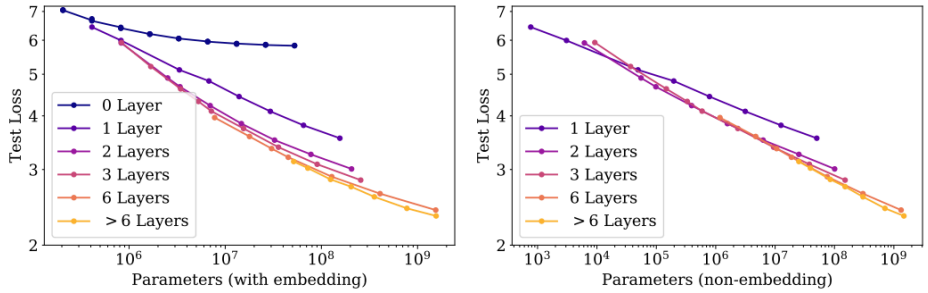

图 6: 左：当包含 embedding 参数时，性能除了依赖参数数量外，还表现出对层数的强依赖。右：当不计入 embedding 参数时，不同深度模型的性能收敛到同一趋势线上。只有层数少于 2 或深宽比极端的模型才会显著偏离该趋势。

3.1 近似的 Transformer 结构与超参数无关性

当我们固定 non-embedding parameter 数量 N 时，Transformer 的性能对 n_{layer}、n_{heads} 和 d_{ff} 等结构参数的依赖非常弱。为验证这一点，我们在固定模型规模的情况下，单独调整某一超参数。对于 n_{heads}，操作最为简单。调整 n_{layer} 时，我们同时调整 d_{model}，以保持 $N \approx 12 n_{layer} d_{model}^2$ 不变。类似地，调整 d_{ff} 时也需同步调整 d_{model}，具体参数关系见表 1。如果更深的 Transformer 能有效表现为浅层模型的集成（如 ResNet 所示，VWB16），则 n_{layer} 的无关性也可成立。相关结果见图 5。

3.2 与 Non-Embedding Parameter 数量 N 的性能关系

图 6 展示了多种模型的性能，规模从 $(n_{layer}, d_{model}) = (2, 128)$ 的小模型到 $(6, 4288)$、$(207, 768)$ 的十亿参数级大模型。所有模型均在完整 WebText2 数据集上训练至接近收敛，几乎未见过拟合（仅极大模型可能例外）。

如图 1 所示，non-embedding parameter 数量 N 与性能之间存在稳定趋势，可用公式 (1.5) 的第一项拟合：

$$L(N) \approx \left(\frac{N_c}{N}\right)^{\alpha_N} \tag{3.1}$$

要观察到这些趋势，必须将性能作为 N 的函数来研究；如果用总参数量（包括 embedding 参数），趋势会被部分掩盖（见图 6）。这说明 embedding 矩阵可以缩小而不影响性能，正如近期研究所示（LCG+19）。

虽然这些模型是在 WebText2 上训练的，但在多种其他数据集上的测试损失同样与 N 呈幂律关系，且幂指数几乎一致（见图 8）。

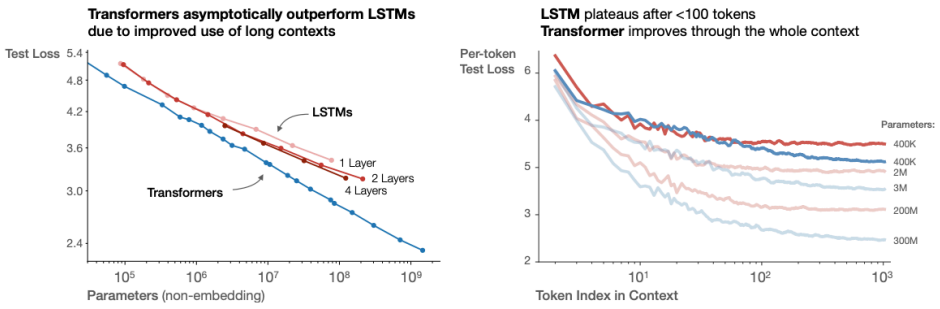

Figure 7

图 7: LSTM 与 Transformer 在不同模型规模下的性能对比。LSTM 在上下文前部 token 上表现与 Transformer 相当，但在后部 token 上无法匹敌 Transformer。

3.2.1 与 LSTM 和 Universal Transformer 的对比

图 7 对比了 LSTM 和 Transformer 在 non-embedding parameter 数量 N 下的性能。LSTM 使用相同的数据集和上下文长度。从图中可以看出，LSTM 在上下文前部 token 上表现与 Transformer 相当，但在后部 token 上无法匹敌 Transformer。我们在附录 D.5 给出了性能与上下文位置的幂律关系，模型越大，幂指数越大，表明其更快识别模式的能力更强。

我们还在附录图 17 中对比了标准 Transformer 与 recurrent Transformer（DGV+18）的性能。后者参数复用，因此在 N 相同时性能略优，但每参数计算量更大。

3.3 不同数据分布上的泛化能力

我们还在一组额外的文本数据分布上测试了模型。图 8 展示了这些数据集上测试损失与模型规模的关系；所有模型均仅在 WebText2 上训练。可以看到，这些分布上的损失随模型规模平滑下降，与 WebText2 上的提升趋势一致。我们发现，泛化能力几乎完全取决于 in-distribution 验证损失，与训练时长或收敛程度无关。我们也未观察到对模型深度的依赖（见附录 D.8）。

3.4 与数据集规模和计算量的性能关系

我们在图 1 中展示了测试损失随数据集规模 D（以 tokens 计）和训练计算量 C 的经验趋势。

关于 D 的趋势，我们用 $(n_{layer}, n_{embd}) = (36, 1280)$ 的模型在 WebText2 的不同子集上训练，训练至测试损失不再下降。结果表明，测试损失可用简单幂律拟合：

$$L(D) \approx \left(\frac{D_c}{D}\right)^{\alpha_D} \tag{3.2}$$

数据和拟合见图 1。

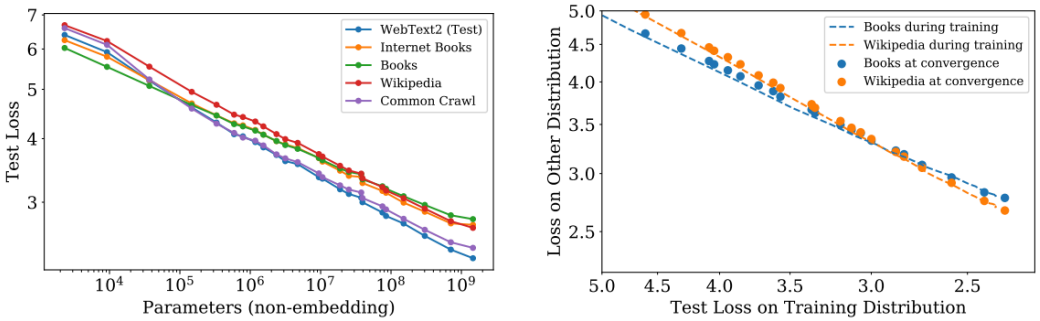

图 8: 左：对其他数据分布的泛化性能随模型规模平滑提升，与 WebText2 训练分布相比仅有很小且增长极慢的偏移。右：泛化性能仅依赖于训练分布上的性能，与训练阶段无关。我们将收敛模型的泛化（点）与单个大模型在训练过程中的泛化（虚线）进行了对比。

训练总计算量 C 可估算为 $C = 6NBS$，其中 B 为 batch size，S 为参数更新步数，系数 6 考虑了前向和反向传播。因此，对于给定 C，我们可以遍历不同 N 的模型，找到在步数 $S = C/(6BS)$ 下性能最优的模型。注意，这些实验中 B 固定，因此结果并非真正最优。后续章节将用调整后的 $C_{\min}$ 得到更清晰的趋势。

结果见图 1 左侧的粗黑线。可用下式拟合：

$$L(C) \approx \left(\frac{C_c}{C}\right)^{\alpha_C} \tag{3.3}$$

图中还包含了单个学习曲线的示意，以说明何时单个模型最优。我们将在后文更详细地研究计算资源的最优分配。数据强烈表明，样本效率随模型规模提升而提升，相关直观展示见附录图 19。

4 无限数据极限与过拟合的刻画

在第 3 节中，我们发现了语言建模性能的一些基本缩放定律。本节将研究在同时变化模型规模 N 和数据集规模 D 时，模型在 D 个 tokens 上训练的性能。我们将通过实证展示，最优训练下的测试损失与公式 (1.5) 的缩放定律一致。这为在控制过拟合的前提下，训练更大模型所需的数据量提供了指导。

4.1 L(N, D) 公式的提出

我们选择了如下参数化（公式 (1.5)，为方便起见在此重复）：

$$L(N, D) = \left[\left(\frac{N_c}{N}\right)^{\alpha_N/\alpha_D} + \frac{D_c}{D}\right]^{\alpha_D} \tag{4.1}$$

该公式基于三条原则：

1. 词表大小或分词方式的变化会整体缩放损失，因此 $L(N, D)$（以及所有损失模型）必须自然支持这种缩放。
2. 固定 D，令 $N \to \infty$，损失应趋近于 $L(D)$；反之，固定 N，令 $D \to \infty$，损失应趋近于 $L(N)$。
3. $L(N, D)$ 在 $D = \infty$ 处应为解析函数，即可按 $1/D$ 的整数幂展开。该原则的理论支持不如前两条强。

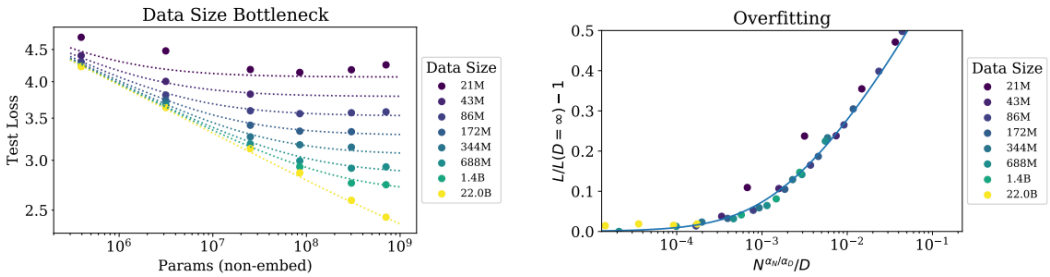

图 9: 提前停止时的测试损失 $L(N, D)$ 随数据集规模 D 和模型规模 N 的变化可由公式 (1.5) 预测。左：当 D 很大时，性能随 N 呈现幂律关系；当 D 较小时，随着 N 增大，性能提升停止，模型开始过拟合（反之亦然，见图 4）。右：过拟合的程度主要取决于 $N^{\alpha_N/\alpha_D}/D$ 的比值，正如公式 (4.3) 所预测。图中直线为该公式的拟合结果。

我们选择的 $L(N, D)$ 满足第一条，因为 N_c, D_c 可随词表变化而缩放，因此它们的具体数值没有本质意义。由于我们在测试损失不再下降时提前停止训练，并对所有模型采用相同优化方式，因此预期更大模型总能优于小模型。但在有限 D 下，任何模型都无法逼近最佳损失（即文本熵）；同理，固定模型规模时也会受限于容量。这些考虑动机了第二条原则。注意，已知 $D \to \infty$ 时的 $L(N)$ 和 $N \to \infty$ 时的 $L(D)$，即可完全确定 $L(N, D)$ 的所有参数。

第三条原则更具猜测性。理论上，过拟合在大 D 时应 $\propto 1/D$，因为过拟合与数据集的方差或信噪比相关（AS17），而这正比于 $1/D$。只要损失函数平滑，这一预期应成立，因为我们可以围绕 $D \to \infty$ 展开损失。但该论证假设 $1/D$ 修正项主导于其他方差来源（如有限 batch size 及优化极限）。没有实证支持时，这一原则的适用性有限。

第三条原则也解释了公式 (1.5) 中 N 和 D 角色的不对称。实际上，也可以写出对称表达式，但那样就没有 $1/D$ 的整数幂展开，并需引入额外参数。

无论如何，我们将看到 $L(N, D)$ 公式与数据拟合良好，这也是我们采用该 ansatz 的最重要理由。

4.2 结果

我们对所有模型使用 10% dropout，并在测试损失不再下降时停止训练。结果见图 9，并对公式 (1.5) 中的四个参数 $\alpha_N, \alpha_D, N_c, D_c$ 进行了拟合：

Parameter	α_N	α_D	N_c	D_c
Value	0.076	0.103	6.4×10^{13}	1.8×10^{13}

表 2: L(N, D) 的拟合参数

拟合效果极佳，唯一例外是数据集被缩小 1024 倍（约 2×10^7 tokens）时的实验。如此小的数据集下，一个 epoch 仅有 40 次参数更新，可能代表了语言建模的不同范式，因为过拟合在训练早期就发生（见图 16）。另外，这里的参数与第 3 节略有不同，因为本节拟合的是完整的 $L(N, D)$，而非 $L(N, \infty)$ 或 $L(\infty, D)$。

为了描绘无限数据极限的边界，我们可以直接研究过拟合的程度。除最大模型外，所有模型在 220 亿 token 的 WebText2 上训练时均未见过拟合，因此可视为 $D = \infty$ 的代表。于是我们可以通过

$$\delta L(N, D) \equiv \frac{L(N, D)}{L(N, \infty)} - 1 \tag{4.2}$$

比较有限 D 与无限数据极限，并将其作为 N, D 的函数进行研究。实证发现，δL 仅依赖于 N 和 D 的特定组合（见图 16），这由公式 (1.5) 的缩放定律推出：

$$\delta L \approx \left[1 + \left(\frac{N}{N_c}\right)^{\alpha_N/\alpha_D} \frac{D_c}{D}\right]^{\alpha_D} - 1 \tag{4.3}$$

注意，在大 D 时该公式也可按 $1/D$ 展开。

我们估算，不同随机种子的损失变化约为 0.02，这意味着若要在收敛阈值内避免过拟合，需要满足

$$D \gtrsim (5 \times 10^3) N^{0.74} \tag{4.4}$$

据此，参数量小于 10^9 的模型可在 220 亿 token 的 WebText2 上训练而几乎无过拟合，但最大模型会出现轻微过拟合。更一般地，这一关系表明，只要数据集规模亚线性增长即可避免过拟合。但这通常并不代表最优的计算效率训练。需要强调的是，我们在调整数据集和模型规模时，并未对正则化（如 dropout 概率）进行优化。

5 模型规模与训练时间的缩放定律

本节将展示，一个简单的缩放定律可以很好地描述损失关于模型规模 N 和训练时间的关系。首先，我们将说明如何利用 [MKAT18] 的结果定义一个通用的训练步数 $S_{\min}$，以修正大多数模型未在最优批量下训练的情况。随后，我们将展示，损失关于模型规模和训练时间的依赖可以用公式 (1.6) 拟合。之后我们将利用这些结果预测训练计算资源在模型规模和训练时间之间的最优分配，并验证该预测。

5.1 Bcrit(L) 批量调整

[MKAT18] 提出了一个关于训练批量依赖的简单经验理论（见 [SLA+18, ZLN+19]）。其核心观点是，存在一个临界批量 B_{crit}，当 $B \leq B_{\text{crit}}$ 时，增大批量几乎不会降低计算效率；而 $B > B_{\text{crit}}$ 时，增大批量带来的收益递减。梯度噪声尺度可用于预测 B_{crit}，且二者都不直接依赖模型规模，只与已达到的损失值有关。这些结果可用于预测训练时间和计算量随批量的变化。为了最有效地利用训练时间和计算资源，最佳做法是采用 $B \approx B_{\text{crit}}$ 训练。$B \gg B_{\text{crit}}$ 可最小化训练步数，而 $B \ll B_{\text{crit}}$ 可最小化计算量。

更具体地说，对于多种神经网络任务，训练步数 S 和处理样本数 $E = BS$ 满足如下关系：

$$\left(\frac{S}{S_{\min}} - 1\right)\left(\frac{E}{E_{\min}} - 1\right) = 1 \tag{5.1}$$

当训练到任意固定损失 L 时成立。这里 $S_{\min}$ 是达到 L 所需的最小步数，$E_{\min}$ 是所需处理的最小样本数。

我们在附录图 18 中对 Transformer 验证了关系 (5.1)。该关系定义了临界批量：

$$B_{\text{crit}}(L) \equiv \frac{E_{\min}}{S_{\min}} \tag{5.2}$$

它是目标损失 L 的函数。在临界批量下训练可实现时间/计算的近似最优权衡，需要 $2S_{\min}$ 步训练，处理 $E = 2E_{\min}$ 个样本。

图 10 展示了两个不同模型的临界批量和梯度噪声尺度随训练损失的变化。可以看到，$B_{\text{crit}}(L)$ 与模型规模无关，仅依赖于损失 L。因此 [MKAT18] 的结论同样适用于 Transformer 语言模型。

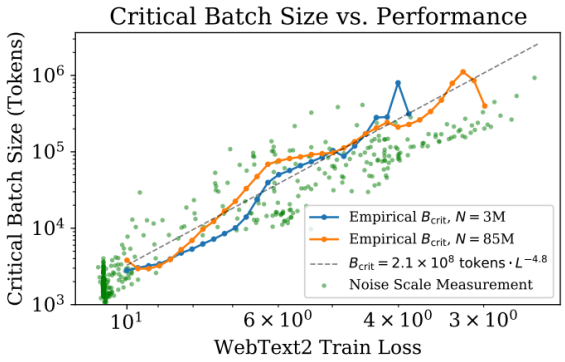

图 10: 临界批量大小 B_{crit} 随损失的降低呈幂律增长，并且不直接依赖于模型规模。我们发现，每当损失下降约 13% 时，临界批量大小大约翻倍。B_{crit} 由图 18 中的数据实测得到，同时也可由梯度噪声尺度近似预测（见 MKAT18）。

临界批量可用损失的幂律拟合：

$$B_{\text{crit}}(L) \approx B^* L^{1/\alpha_B} \tag{5.3}$$

其中 $B^* \approx 2 \times 10^8$，$\alpha_B \approx 0.21$。

我们选择这种参数化，是因为当损失趋近于最小值 L_{min} 时，梯度噪声尺度预期会发散，B_{crit} 也会随之发散。我们并不知道 L_{min}，因为模型尚未接近该值，但 $L_{\text{min}} > 0$，因为自然语言的熵非零。由于 L_{min} 远小于我们目前达到的 L，我们采用 B_{crit} 在 $L \to 0$ 时发散的参数化。

我们将用 $B_{\text{crit}}(L)$ 估算在 $B = 2^{19}$ tokens 下训练步数 S 与在 $B \gg B_{\text{crit}}$ 下训练步数的关系：

$$S_{\text{min}}(S) \equiv \frac{S}{1 + B_{\text{crit}}(L)/B} \quad \text{(minimum steps, at } B \gg B_{\text{crit}}\text{)} \tag{5.4}$$

对于任意目标损失 L。这也定义了在 $B \ll B_{\text{crit}}(L)$ 下训练到 L 所需的最小计算量：

$$C_{\text{min}}(C) \equiv \frac{C}{1 + B/B_{\text{crit}}(L)} \quad \text{(minimum compute, at } B \ll B_{\text{crit}}\text{)} \tag{5.5}$$

其中 $C = 6NBS$ 估算为 batch size B 下的（非嵌入）训练计算量。

5.2 L(N, Smin) 及模型规模与计算量的性能

现在我们用公式 (5.4) 定义的 S_{min}，得到损失关于模型规模和训练时间的简单通用拟合（无限数据极限下）。我们用 Adam 优化的稳定训练过程，采用公式 (1.6)（为方便起见在此重复）：

$$L(N, S_{\text{min}}) = \left(\frac{N_c}{N}\right)^{\alpha_N} + \left(\frac{S_c}{S_{\text{min}}}\right)^{\alpha_S} \tag{5.6}$$

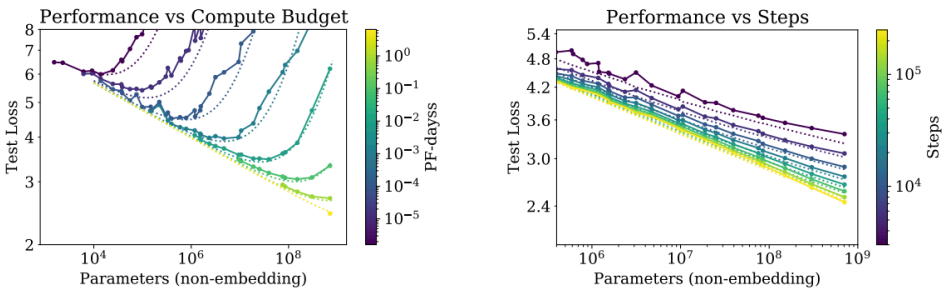

图 11: 当我们固定总计算量或训练步数时，性能遵循公式 (5.6) 的 $L(N, S)$。每个计算预算对应一个最优模型规模，使性能最大化。在小 S 区间的拟合效果一般，这并不意外，因为学习曲线的幂律公式在训练初期并不成立。

我们在学习率 warmup 之后的所有训练步数上进行拟合，得到如下参数：

Parameter	α_N	α_S	N_c	S_c
Value	0.077	0.76	6.5×10^{13}	2.1×10^3

表 3: L(N, S) 的拟合参数

用这些参数，我们得到了图 4 中的学习曲线拟合。虽然拟合并不完美，但考虑到公式 (5.6) 的简洁性，我们认为结果非常有说服力。

数据和拟合还可以用另一种更有趣的方式可视化，如图 11 所示。我们研究了在固定训练总计算量 C 或固定步数 S 时，测试损失随模型规模的变化。拟合时用到了公式 (5.5)、(5.4) 及上述参数和公式 (5.6)。

损失对 $S_{\min}$ 的幂律依赖反映了优化器动力学与损失景观的相互作用。由于拟合在训练后期效果最佳，此时损失近似为二次型，幂律应反映损失 Hessian 的谱分布。其普适性表明 Hessian 特征值密度大致与模型规模无关。

5.3 提前停止步数的下界

$L(N, S_{\min})$ 的结果可用于推导数据受限训练时提前停止步数的下界（及粗略估计）。其动机是，对于给定模型，有限和无限 D 的学习曲线在 $S_{\min} \approx S_{\text{stop}}$ 前非常相似。因此，过拟合应与在 S_{stop} 处简单终止训练的修正项成正比。实际上，这会低估 S_{stop}，因为有限 D 时测试损失下降更慢，因此需要更多训练步数才能达到最优测试损失。由此推导出不等式：

$$S_{\text{stop}}(N, D) \gtrsim \frac{S_c}{[L(N, D) - L(N, \infty)]^{1/\alpha_S}} \tag{5.7}$$

其中 $L(N, \infty)$ 为收敛损失，在无限数据下评估。该不等式及与实证数据的对比见附录图 16。图中 S_{stop} 和 $L(N, D)$ 为实测值（S_{stop} 已调整以模拟 $B \gg B_{\text{crit}}$ 下训练），而 $L(N, \infty)$ 由 $L(N, D)$ 在 $D = \infty$ 处的拟合计算。

6 计算预算的最优分配

我们在图 1 右上展示了性能随训练计算量变化的经验趋势。然而，该结果是在固定 batch size B 下训练得到的，而实际上如果采用第 5.1 节讨论的临界 batch size B_{crit} 训练，可以更高效地利用计算资源。较大或较小的损失值本可以通过更少的样本或更少的步数实现，对这种低效进行修正、统一到临界 batch size 后，趋势会更加清晰和可预测。

本节将对上述问题进行修正。更重要的是，我们将利用第 5 节的结果，确定计算资源在模型规模 N 和训练过程中处理的数据量（即 $2B_{\text{crit}}S_{\min}$）之间的最优分配。我们将通过实证和理论两种方式（利用 $L(N, S_{\min})$ 公式）确定这一分配，并展示两者结果一致。

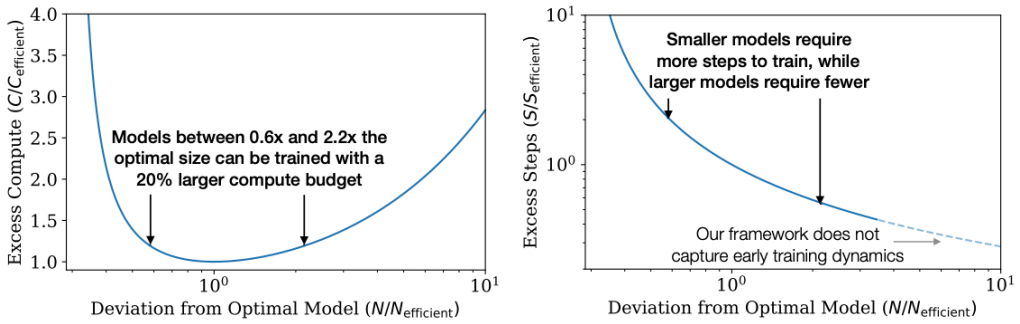

图 12: 左：在固定计算预算下，存在一个最优模型规模，但略大或略小的模型也只需极少的额外计算资源。右：大于计算效率最优规模的模型所需训练步数更少，如果能实现足够的并行，训练速度可能更快。需要注意的是，该公式在非常大的模型规模下并不适用，仅在学习曲线的幂律区间（初始过渡阶段之后）有效。

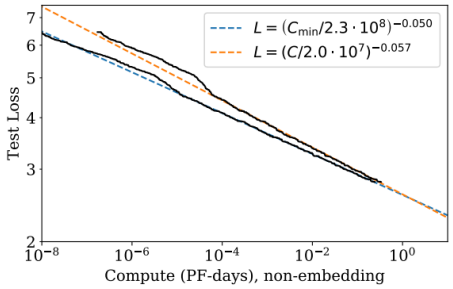

图 13: 当将性能调整为模拟远低于临界批量训练时，$L(C_{\min})$ 呈现出与完全实证结果略有不同的幂律关系。10^{-5} PF-days 处明显的突起标志着从 1 层网络到 2 层网络的转变；在幂律拟合中我们排除了 1 层网络。我们预计 $L(C_{\min})$ 的趋势能为更大计算量的外推提供可靠依据。

6.1 最优性能与分配

首先研究损失关于最优分配计算量（公式 (5.5)）的关系。结果见图 13，并给出了幂律拟合。与图 1 中的计算量曲线相比，采用 $C_{\min}$ 后的拟合有所提升。

已知 $L(C_{\min})$，自然会问：在给定训练计算量下，最优模型规模 $N(C_{\min})$ 是多少？最优模型规模见图 14。可以看到，$N(C_{\min})$ 可用幂律很好地拟合：

$$N(C_{\min}) \propto (C_{\min})^{0.73} \tag{6.1}$$

在图 12 中，我们展示了训练非最优规模模型的影响（见附录 B.4）。

根据定义 $C_{\min} \equiv 6NB_{\text{crit}}S$，因此可用 $N(C_{\min})$ 推导更多结果。特别地，已有拟合表明 $B \propto L^{-4.8}$ 且 $L \propto C_{\min}^{-0.05}$，可得 $B_{\text{crit}} \propto C_{\min}^{0.24}$。由此可知，最优步数随计算量增长极慢：

$$S_{\min} \propto (C_{\min})^{0.03} \tag{6.2}$$

与图 14 中的实证结果一致。实际上，测得的指数足够小，甚至可能与零一致。

因此我们得出结论：在最优分配计算资源进行语言建模时，应主要增加模型规模 N，同时通过 $B \propto B_{\text{crit}}$ 增大 batch size，而串行步数几乎无需增加。由于高效训练所需优化步数较少，进一步加速训练初期动态可能很有价值。

6.2 L(N, Smin) 的预测

$L(C_{\min})$ 及分配结果可由第 5 节得到的 $L(N, S_{\min})$ 公式预测。已知 $L(N, S_{\min})$，可代入 $S_{\min} = C_{\min}/(6NB)$，然后在固定训练计算量下对 N 求最小损失。我们在附录 B 中详细给出了这一过程及更多预测。

对于损失关于训练计算量的关系，预测为

$$L(C_{\min}) = \left(\frac{C_c^{\min}}{C_{\min}}\right)^{\alpha_C^{\min}} \tag{6.3}$$

其中

$$\alpha_C^{\min} \equiv \frac{1}{1/\alpha_S + 1/\alpha_B + 1/\alpha_N} \approx 0.054 \tag{6.4}$$

与图 13 中的指数高度一致。还可预测

$$N(C_{\min}) \propto (C_{\min})^{\alpha_C^{\min}/\alpha_N} \approx (C_{\min})^{0.71} \tag{6.5}$$

与图 14 中的缩放关系相差仅几个百分点。我们的缩放定律为语言建模性能提供了可预测的理论框架。

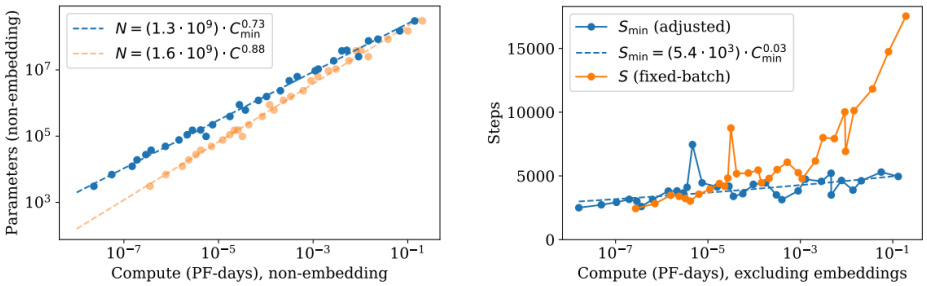

图 14: 左：每个计算预算 $C_{\min}$ 都对应一个最优模型规模 N。最优模型规模随 $C_{\min}$ 急剧增长，每增加 10 倍计算，模型规模增加约 5 倍。其余的增长由处理的数据样本数补足，数据量仅增加约 2 倍。右：批量调整后的优化步数增长极慢甚至几乎不变，这意味着大部分数据量的增长都可用于增大 batch size。

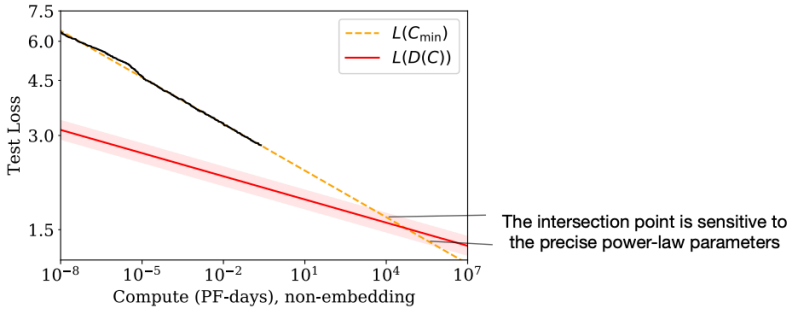

图 15: 在远超我们实证研究的模型规模下，由于高效训练所需数据增长缓慢，$L(C_{\min})$ 和 $L(D)$ 的公式出现矛盾。两者的交点标志着我们预测失效的临界点。该点的位置对幂律拟合所得指数极为敏感。

6.3 矛盾与猜想

我们未观察到在大计算量、数据量或模型规模下偏离幂律趋势的迹象。但这些趋势最终必然趋于平稳，因为自然语言的熵非零。

事实上，本节描述的高效训练趋势已经包含一个明显的矛盾。在远超本文记录范围的规模下，$L(C_{\min})$ 的缩放定律预测的性能会低于数据随计算量缓慢增长时的极限。这意味着缩放定律在此之前必然失效，但我们猜测这个交点有更深层的意义：它给出了 Transformer 语言模型达到极限性能的估计点。

由于高效训练的数据量随计算预算增长很慢，$L(C_{\min})$ 预测的性能最终会被 $L(D)$ 幂律设定的下界限制（见图 15）。具体分析如下：

为控制过拟合，第 4 节结果表明应按如下方式扩展数据集规模：

$$D \propto N^{0.74} \propto C_{\min}^{0.54} \tag{6.6}$$

其中 $N(C_{\min})$ 取自图 14 的高效训练。

与高效训练的数据需求对比：若始终在临界 batch size 下训练（即 $C = 2C_{\min}$），且训练过程中从不重复使用数据，则数据用量随计算增长为

$$D(C_{\min}) = \frac{2C_{\min}}{6N(C_{\min})} \approx 4 \times 10^{10} \text{ tokens} \cdot (C_{\min}/\text{PF-Day})^{0.26} \tag{6.7}$$

这代表数据集规模增长的最大速率（即只训练一个 epoch），但比公式 (6.6) 增长得慢得多。

这意味着即使训练过程中从不重复数据，高效训练最终也会遇到过拟合问题！

根据图 1，受限于数据集规模（即过拟合）时，损失应按 $L(D) \propto D^{-0.095}$ 缩放。这意味着损失随计算量缩放为 $L(D(C_{\min})) \propto C_{\min}^{-0.03}$，而我们在图 13 中得到 $L(C_{\min}) \propto C_{\min}^{-0.050}$，两者最终会相交。

$L(D(C_{\min}))$ 与 $L(C_{\min})$ 的交点为

$$C^* \sim 10^4 \text{ PF-Days}, \quad N^* \sim 10^{12} \text{ parameters}, \quad D^* \sim 10^{12} \text{ tokens}, \quad L^* \sim 1.7 \text{ nats/token} \tag{6.8}$$

但这些数值高度不确定，随幂律拟合指数的不同可有一个数量级的变化。最直接的解释是，缩放定律在或早于该点失效，而该点距离当前计算和模型规模仍有多个数量级。

也可以猜测，这个交点有更深的含义。如果模型规模无法超过 N^* 而不需要质变的数据需求，或许意味着当达到 $C^*_{\min}$ 和 N^* 时，已从自然语言数据中提取了所有可靠信息。在这种解释下，L^* 可视为自然语言的每 token 熵的粗略估计。在这种情形下，损失趋势会在 L^* 附近趋于平稳。

我们可以通过考虑带噪声的训练集，猜测 $L(C_{\min})$ 趋于平稳时的函数形式。例如，可以在每个上下文后附加一串随机 token，人为提升损失一个常数项。此时，$L - L_{\text{noise}}$ 距离更有意义，即使该距离的微小下降也可能带来显著的性能提升。由于人工噪声会等比例影响所有趋势，6.8 节的临界点不会改变（除了 L^* 的绝对值），即使发生在平稳点之后也可能有意义。

7 相关工作

幂律现象可以由多种机制产生（THK18）。在密度估计（Was06）和随机森林模型（Bia12）中，模型和数据集规模的幂律缩放可能与我们的结果有关。这些模型表明，幂律指数可以粗略地理解为数据中相关特征数量的倒数。

早期的一些工作（BB01, Goo01）发现了性能与数据集规模之间的幂律关系。近期的研究（HNA+17, HAD19）也探讨了模型规模与数据规模之间的缩放关系；他们的工作可能是与我们最接近的文献 8。需要注意的是，HNA+17 发现数据集规模与模型规模呈超线性缩放，而我们发现是亚线性缩放。我们的关于计算资源最优分配的发现与 Kom19 有一定相似之处，包括幂律学习曲线。EfficientNets（TL19）同样表现出准确率与模型规模之间的近似幂律关系。最新的研究（RRBS19b）在多种数据集上研究了数据集规模和模型规模的缩放，并拟合了与我们类似的 ansatz。

EfficientNet（TL19）主张为获得图像模型的最优性能，应以不同系数指数级地扩展深度和宽度，从而使宽度作为深度的幂律函数缩放。我们发现，对于语言模型，这个幂应当大致为 1（即扩展时宽深比应保持不变）。更重要的是，我们发现，与语言模型的整体规模相比，具体的结构超参数并不重要。在 VWB16 中，作者认为深层模型可以作为浅层模型的集成，这可能解释了我们的发现。早期工作（ZK16）比较了宽度和深度，发现宽 ResNet 在图像分类上可优于深 ResNet。一些研究固定每个数据样本的计算量，这通常与模型参数数量成正比，而我们则同时研究了模型规模和训练计算量的缩放。

多项工作（AS17, BHMM18）研究了高度过参数化模型的泛化，发现当模型规模达到数据集规模时会出现 “jamming transition”（GJS+19），但这通常需要远超实际应用的训练规模，且未采用提前停止。我们未观察到这种转变，发现所需训练数据随模型规模亚线性增长。模型规模的扩展，尤其是在大宽度下（JGH18, LXS+19），可能为理解我们的缩放关系提供有用框架。我们关于优化的结果，如学习曲线的形状，很可能可以用带噪声的二次模型解释，该模型在实际场景下可给出相当准确的预测（ZLN+19）。要将这种联系定量化，还需刻画 Hessian 的谱分布（Pap18, GKX19, GARD18）。

8 讨论

我们观察到语言模型的 log-likelihood 损失与 non-embedding parameter 数量 N、数据集规模 D 以及优化后的训练计算量 $C_{\min}$ 存在一致的缩放关系，这些关系体现在公式 (1.5) 和 (1.6) 中。相反，我们发现模型对许多结构和优化超参数的依赖非常弱。由于 N、D、$C_{\min}$ 的缩放关系是幂律的，随着规模的增加，收益会递减。

我们能够精确建模损失对 N 和 D，以及对 N 和 S 的依赖（当这些参数同时变化时）。我们利用这些关系推导了计算缩放、过拟合幅度、提前停止步数以及训练大语言模型时的数据需求。因此，我们的缩

放关系不仅仅是观察结果，更提供了一个可预测的理论框架。可以将这些关系类比为理想气体定律，它以普适的方式联系了气体的宏观属性，而与其微观组成的细节无关。

很自然地猜想，这些缩放关系也适用于其他最大似然损失的生成建模任务，甚至可能适用于其他领域。为此，在图像、音频、视频模型等其他领域，以及随机网络蒸馏等任务上测试这些关系将很有意义。目前我们还不清楚哪些结果依赖于自然语言数据的结构，哪些是普适的。如果能找到一个理论框架来推导这些缩放关系——即我们观察到的“热力学”背后的“统计力学”——那将非常令人兴奋。这样的理论可能使我们能够推导出更精确的预测，并系统性地理解缩放定律的局限性。

在自然语言领域，还需进一步研究损失的持续改进是否能转化为相关语言任务的性能提升。平滑的量变可能掩盖了重大的质变：“多即不同”。例如，经济的平滑增长无法反映其背后的具体技术进步。同样，语言模型损失的平滑提升也可能隐藏了能力上的质变。

我们的结果强烈表明，更大的模型将持续表现更好，并且其样本效率也远高于以往的认知。大模型可能比大数据更为重要。在此背景下，进一步研究模型并行化是有意义的。深层模型可以通过流水线（HCC+18）进行训练，将参数按深度在设备间分配，但随着设备数量增加，最终需要更大的 batch size。另一方面，宽网络更易于并行化（SCP+18），因为大层可以在多个工作节点间分割，串行依赖更少。稀疏性（CGRS19, GRK17）或分支结构（如 KSH12）可能通过提升模型并行性，实现更快的大模型训练。利用 WRH17, WYL19 等方法，在训练过程中动态扩展网络，或许可以在整个训练过程中始终保持在计算效率前沿。

A 幂律总结

为便于查阅，现将全文描述的主要趋势总结如下：

Parameters	Data	Compute	Batch Size	Equation
N	∞	∞	Fixed	$L(N) = (N_c/N)^{\alpha_N}$
∞	D	Early Stop	Fixed	$L(D) = (D_c/D)^{\alpha_D}$
Optimal	∞	C	Fixed	$L(C) = (C_c/C)^{\alpha_C}$ (naive)
N_{opt}	D_{opt}	$C_{\min}$	$B \ll B_{\text{crit}}$	$L(C_{\min}) = (C_c^{\min}/C_{\min})^{\alpha_C^{\min}}$
N	D	Early Stop	Fixed	$L(N, D) = \left[(N_c/N)^{\alpha_N/\alpha_D} + D_c/D\right]^{\alpha_D}$
N	∞	S steps	B	$L(N, S) = (N_c/N)^{\alpha_N} + (S_c/S_{\min}(S, B))^{\alpha_S}$

表 4: 主要幂律关系公式

这些趋势的经验拟合参数如下：

Power Law	Scale (tokenization-dependent)	Value
α_N	N_c (non-embed)	0.076, 8.8×10^{13} params
α_D	D_c (tokens)	0.095, 5.4×10^{13} tokens
α_C	C_c (PF-days)	0.057, 1.6×10^{7} PF-days
$\alpha_C^{\min}$	$C_c^{\min}$ (PF-days)	0.050, 3.1×10^{8} PF-days
α_B	B^* (tokens)	0.21, 2.1×10^{8} tokens
α_S	S_c (steps)	0.76, 2.1×10^{3} steps

表 5: 幂律拟合参数

计算效率最优训练的参数为：

Compute-Efficient Value	Power Law	Scale
$N_{\text{opt}} = N_e \cdot C_{\min}^{p_N}$	$p_N = 0.73$	$N_e = 1.3 \times 10^{9}$ params
$B \ll B_{\text{crit}} = B^* L^{1/\alpha_B} = B_e C_{\min}^{p_B}$	$p_B = 0.24$	$B_e = 2.0 \times 10^{6}$ tokens
$S_{\min} = S_e C_{\min}^{p_S}$ (lower bound)	$p_S = 0.03$	$S_e = 5.4 \times 10^{3}$ steps
$D_{\text{opt}} = D_e C_{\min}^{p_D}$ (1 epoch)	$p_D = 0.27$	$D_e = 2 \times 10^{10}$ tokens

表 6: 计算效率最优参数

B B 计算效率前沿的经验模型

本附录中，所有 C、S 和 α_C 的数值均已针对在临界批量 B_{crit} 下训练进行了调整。为简化记号，省略了 “adj” 标签。

B.1 B.1 定义方程

对学习曲线的幂律拟合意味着可以为计算效率最优训练给出简单的方案。本附录将推导最优性能、模型规模和训练步数关于计算预算的函数关系。我们从公式 (1.6) 出发，便于查阅，重写如下：

$$L(N, S) = \left(\frac{N_c}{N}\right)^{\alpha_N} + \left(\frac{S_c}{S}\right)^{\alpha_S} \tag{B.1}$$

其中 S 表示在临界批量 [MKAT18] 下的参数更新步数，定义见公式 (5.2)：

$$B(L) = B^* L^{1/\alpha_B} \tag{B.2}$$

我们希望在固定计算预算下确定最优训练参数，因此用 $S = C/(6NB(L))$ 替换 S，其中 C 是训练所用的 FLOPs 数：

$$L(N, C) = \left(\frac{N_c}{N}\right)^{\alpha_N} + \left(\frac{6B^* S_c}{NL^{1/\alpha_B} C}\right)^{\alpha_S} \tag{B.3}$$

现在，对 N 求偏导并令其为零以获得最优条件：

$$0 = \frac{\partial L}{\partial N}\Big|_C = -\frac{\alpha_N}{N}\left(\frac{N_c}{N}\right)^{\alpha_N} + \frac{\alpha_S}{N}\left(\frac{6B^* S_c}{NL^{1/\alpha_B} C}\right)^{\alpha_S}\left[1 - \frac{N}{L}\frac{\partial L}{\partial N}\Big|_C\right] \tag{B.4}$$

由此可得

$$\frac{\alpha_N}{\alpha_S}\left(\frac{N_c}{N}\right)^{\alpha_N} = \left(\frac{6B^* S_c}{NL^{1/\alpha_B} C}\right)^{\alpha_S} \tag{B.5}$$

公式 (B.3) 和 (B.5) 共同确定了计算效率前沿。

B.2 B.2 高效训练

将 (B.5) 代入 (B.3) 可得

$$L(N_{\text{eff}}(C), C) = \left[1 + \frac{\alpha_N}{\alpha_S}\right] L(N_{\text{eff}}, \infty) \tag{B.6}$$

这意味着对于计算效率最优训练，应训练到收敛损失上方约 $\alpha_N/\alpha_S \approx 10\%$ 的固定百分比。进一步，消去 N 可得性能关于计算的幂律关系：

$$L(C) = \left(\frac{C_c}{C}\right)^{\alpha_C} \tag{B.7}$$

其中

$$\alpha_C = \frac{1}{1/\alpha_S + 1/\alpha_B + 1/\alpha_N} \approx 0.052 \tag{B.8}$$

$$C_c = 6N_c B^* S_c \left[1 + \frac{\alpha_N}{\alpha_S}\right]^{1/\alpha_S + 1/\alpha_N} \left(\frac{\alpha_S}{\alpha_N}\right)^{1/\alpha_S} \tag{B.9}$$

同理，可消去 L 得到 $N(C)$：

$$\frac{N(C)}{N_c} = \left(\frac{C}{C_c}\right)^{\alpha_C/\alpha_N}\left[1 + \frac{\alpha_N}{\alpha_S}\right]^{1/\alpha_N} \tag{B.10}$$

以及

$$S(C) = \frac{C_c}{6N_c B^*}\left[1 + \frac{\alpha_N}{\alpha_S}\right]^{-1/\alpha_N}\left(\frac{C}{C_c}\right)^{\alpha_C/\alpha_S} \tag{B.11}$$

B.3 B.3 与非高效训练的对比

通常，研究者会将模型训练到接近收敛。此处我们将上述高效训练过程与更常见的做法进行对比。定义收敛因子 f 为损失相对收敛值的百分比偏差：

$$L(N, C) = (1 + f)L(N, \infty) \tag{B.12}$$

对于高效训练，$f = \alpha_N/\alpha_S \approx 10\%$，而实际中研究者常用更小的 f。这里取 $f_0 = 2\%$ 作为估算。对于固定损失，预测有：

$$\frac{N_f}{N_{f_0}} = \left(\frac{1+f}{1+f_0}\right)^{1/\alpha_N} \approx 2.7 \tag{B.13}$$

$$\frac{S_f}{S_{f_0}} = \left(\frac{1+1/f}{1+1/f_0}\right)^{1/\alpha_S} \approx 0.13 \tag{B.14}$$

$$\frac{C_f}{C_{f_0}} = \frac{N_f}{N_{f_0}}\frac{S_f}{S_{f_0}} \approx 0.35 \tag{B.15}$$

即高效训练所需参数更新步数减少 7.7 倍，参数量增加 2.7 倍，计算量减少 65% 即可达到同样损失。

B.4 B.4 非最优模型规模

可解 A.1 得到达到给定损失 L 所需的计算量 C 关于模型规模 N 的表达式：

$$C(N, L) = \left(\frac{6B^* S_c}{NL^{1/\alpha_B}}\right)\left[L - \left(\frac{N_c}{N}\right)^{\alpha_N}\right]^{-1/\alpha_S} \tag{B.16}$$

利用 (B.6) 和 (B.10)，可用 $N_{\text{eff}}(L)$（最有效达到 L 的模型规模）消去 L，进而得到使用非最优模型规模所需的额外计算量表达式：

$$\frac{C(N, N_{\text{eff}})}{C(N_{\text{eff}}, N_{\text{eff}})} = \frac{N}{N_{\text{eff}}}\left[1 + \frac{\alpha_S}{\alpha_N}\left(1 - \left(\frac{N_{\text{eff}}}{N}\right)^{\alpha_N}\right)\right]^{-1/\alpha_S} \tag{B.17}$$

结果如图 12 所示。使用 0.6× 到 2.2× 最优规模的模型，仅需增加 20% 的计算预算。若考虑推理成本，使用小模型有优势；而大模型可用更少步数达到同等性能，若硬件允许并行，可加速训练（见图 12 右）：

$$\frac{S(N, N_{\text{eff}})}{S(N_{\text{eff}}, N_{\text{eff}})} = \left[1 + \frac{\alpha_S}{\alpha_N}\left(1 - \left(\frac{N_{\text{eff}}}{N}\right)^{\alpha_N}\right)\right]^{-1/\alpha_S} \tag{B.18}$$

2.2× 大的模型只需 45% 的步数，计算量仅多 20%。注意，该公式仅在学习曲线幂律区间（初始过渡后）有效，极大模型下不适用。

C 注意事项

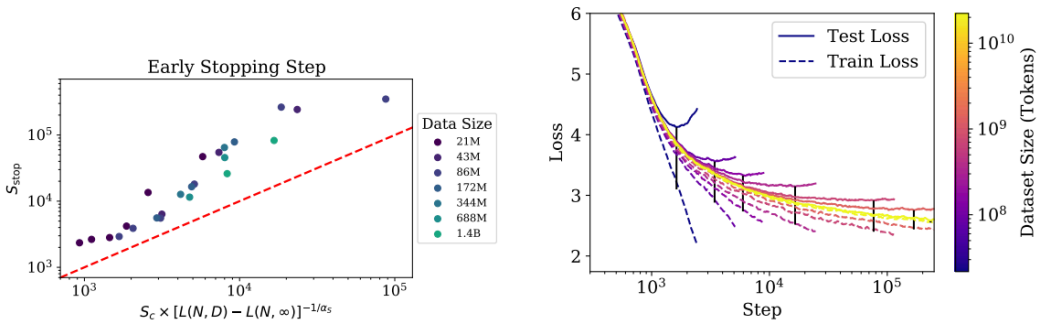

图 16: 左：我们将提前停止发生的步数刻画为过拟合程度的函数。红线表示第 5.3 节推导的提前停止下界。右：展示了一系列 3 亿参数模型在不同规模数据子集上训练的训练损失和测试损失。测试损失通常会跟随无限数据训练的趋势，直到出现偏离。需要注意的是，用 $L_{\text{test}} - L_{\text{train}}$（每次训练用黑色横线标注）来衡量过拟合程度时，往往会显著高估与无限数据极限的实际差距。

本节列出我们分析中可能存在的一些注意事项：

- 目前我们对所提出的任何缩放定律都没有坚实的理论理解。模型规模和计算量的缩放关系尤其神秘。或许可以通过用带噪声的二次函数建模损失，理解在大 D、固定模型规模下的缩放关系（AS17），以及训练后期学习曲线的形状。但在极大模型规模下 D 的缩放仍然难以解释。如果没有理论或对缩放定律修正项的系统理解，就很难判断这些定律在何种情况下是可靠的。

- 对于远超我们实验范围的损失值，$B_{\text{crit}}(L)$ 的预测并不十分可靠。B_{crit} 的变化可能会显著影响数据并行与串行训练步数之间的权衡，从而对训练时间产生重大影响。
- 我们没有系统研究小数据集情形，对于最小 D（一个 epoch 仅对应 40 步）下 $L(N, D)$ 的拟合效果较差。此外，我们没有尝试正则化和数据增强，这些改进可能会定量或定性地改变我们的结果。
- 我们采用了 $C \approx 6NBS$ 估算训练计算量，未包含与 n_{ctx} 成正比的项（见 2.1 节）。因此，在极大 n_{ctx}（尤其 $n_{ctx} \gtrsim 12d_{model}$）下，我们的计算缩放关系在实际中可能会受到影响。
- 我们调优了学习率，并尝试了不同的学习率调度，但可能忽略了某些对缩放有重要影响的超参数（如初始化尺度或动量）。
- 最优学习率的选择对目标损失较为敏感。接近收敛时，可能需要使用更小的学习率以避免发散；而在短训练（如受限于计算资源）时，可能可以用更大的学习率。我们没有在未训练到收敛的情况下尝试更高的学习率。

D　D 补充图表

D.1　D.1 提前停止与测试/训练损失

在第 5.3 节中，我们描述了图 16 所示的结果，该图给出了提前停止步数的下界预测。我们还展示了在不同规模数据集上训练同一模型规模时的训练损失和测试损失。

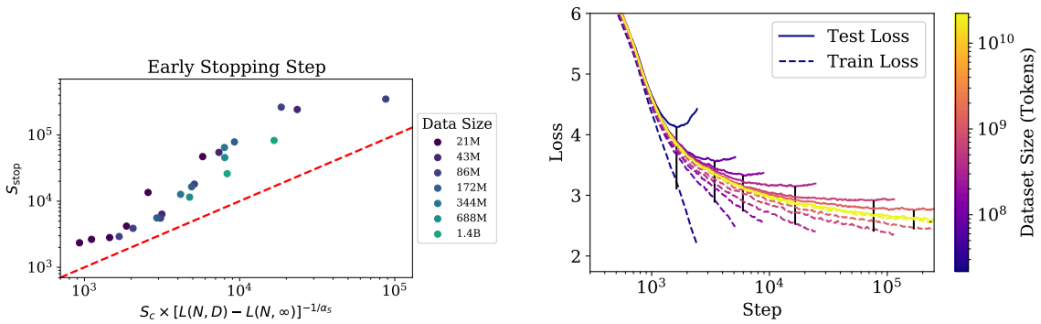

图 17: 左：我们将提前停止发生的步数刻画为过拟合程度的函数。红线表示第 5.3 节推导的提前停止下界。右：展示了一系列 3 亿参数模型在不同规模数据子集上训练的训练损失和测试损失。测试损失通常会跟随无限数据训练的趋势，直到出现偏离。需要注意的是，用 $L_{\text{test}} - L_{\text{train}}$（每次训练用黑色横线标注）来衡量过拟合程度时，往往会显著高估与无限数据极限的实际差距。

D.2　D.2 Universal Transformers

我们在图 17 中对比了标准 Transformer 与 recurrent Transformer（DGV+18）的性能。这些模型参数复用，因此在 N 相同时性能略优，但在计算量 C 相同时略逊。我们还展示了多种参数复用的可能性。

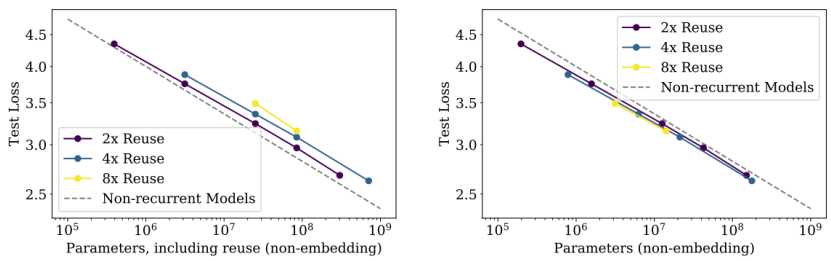

图 18: 我们对比了参数复用的 recurrent Transformer（DGV+18）与标准 Transformer。若以参数量为基准，recurrent Transformer 性能略优；但若考虑参数复用、以每 FLOP 为基准，则略逊。

D.3 D.3 批量大小

我们用图 18 中的数据测量了临界批量大小，从而可以在图 10 中估算 $B_{\text{crit}}(L)$。

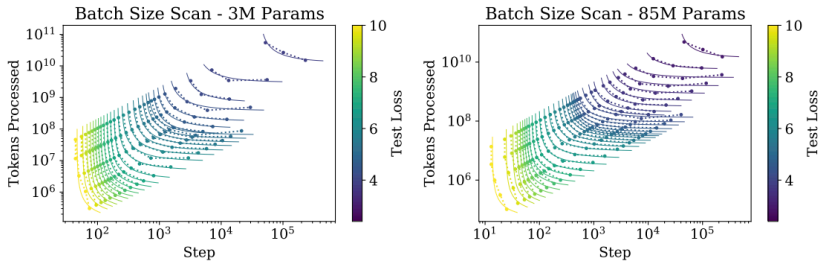

图 19: 这些图展示了对大量损失 L 取值和两种 Transformer 规模下，公式 (5.1) 的拟合结果。这些拟合用于测量图 10 中的 $B_{\text{crit}}(L)$。

D.4 D.4 样本效率与模型规模

从图 2 可以直观看出，大模型训练更快，因此样本效率更高。图 19 用另一种方式展示了这一现象，显示不同模型达到固定损失值所需的训练步数。

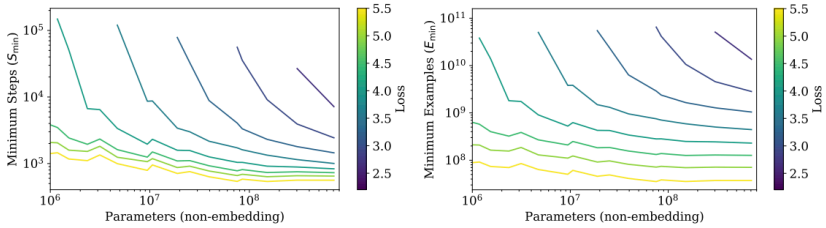

图 20: 达到任意固定测试损失所需的最小串行步数随模型规模急剧下降。样本效率（此处为远低于临界批量训练）也大幅提升，从最小模型到超大模型提升近 100 倍。

D.5 D.5 上下文依赖性

图 21 展示了不同上下文位置 token 的损失随模型规模的变化趋势。可以看到，$n_{ctx} = 1024$ 训练的模型在除第一个 token 外的所有位置上，性能随模型规模稳步提升。

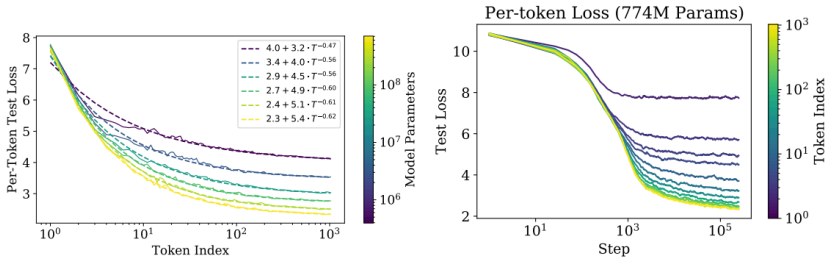

图 21: 本图展示了每 token 性能随模型规模和训练时间的变化。左：每 token 损失随其在 1024-token 上下文中的位置 T 变化，损失随 T 呈幂律缩放。右：每 token 测试损失随训练步数变化。

在固定模型规模时，损失随上下文位置 T 呈幂律缩放（见图 20）。这可能源于语言中的幂律相关性 (EP94, ACDE12, LT16)，也可能是模型结构和优化的普遍特性。这说明训练更大上下文可能带来（或不带来）额外收益。大模型不仅在 $T = 1024$ 时表现更好，在前部 token 上提升也更快，说明大模型在上下文信息较少时更善于识别模式。右图展示了固定模型在训练步数变化时的每 token 性能。模型先学会短程信息，后期才学会长程相关。

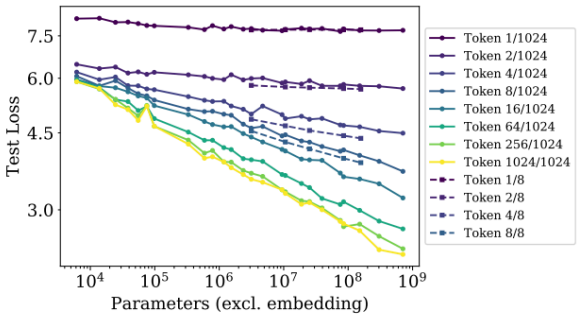

图 22: 除了平均损失外，1024-token 上下文内的各 token 损失也会随模型规模平滑提升。短上下文 $n_{ctx} = 8$（虚线）训练的模型在前部 token 上表现更好，因为其全部容量可分配给这些 token。

我们还包含了 $n_{ctx} = 8$ 的小上下文模型，以与大上下文模型对比。即使是中等规模、$n_{ctx} = 8$ 的模型，在前部 token 上也能优于最大 $n_{ctx} = 1024$ 的模型。这也说明用更大模型和更大上下文训练还有提升空间。

D.6 D.6 学习率调度与误差分析

我们尝试了多种学习率和调度。图 22 展示了小模型在不同调度下的测试性能。结论是，只要总学习率足够大，且调度包含 warmup 和最终衰减，调度方式基本无关紧要。不同调度间的差异主要是统计噪声，可作为不同训练 run 之间变异的粗略尺度。大模型实验表明，不同随机种子下最终测试损失的变异量对不同模型规模基本恒定。

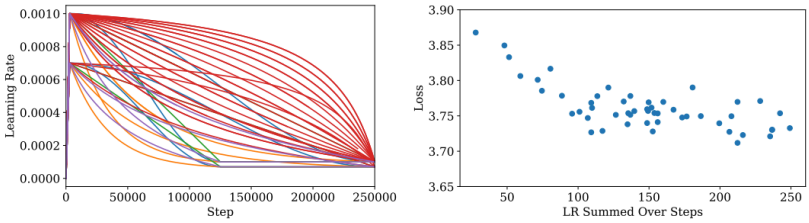

图 23: 我们在一个 300 万参数模型上测试了多种学习率调度，包括余弦衰减、线性衰减及其他快/慢衰减。左图为实验结果。实验中未衰减至零，因为这样做会在训练末期带来固定提升。只要学习率不太小且衰减不太快，性能对调度不敏感。不同 run 之间的损失变异约为 0.05，因此需多次取平均以验证小于此量级的性能变化。

我们发现大模型需用更小学习率防止发散，小模型则可用更大学习率。大多数实验采用如下经验公式：

$$\mathrm{LR}(N) \approx 0.003239 - 0.0001395 \log(N) \tag{1}$$

该公式可进一步改进，可能还与网络宽度有关（由初始化尺度决定）。对于 $N > 10^{10}$ 参数时该公式失效。但对我们实验的模型已足够好用。

D.7 D.7 拟合细节与幂律质量

我们尝试了多种函数形式拟合 $L(N)$、$L(C)$ 和 $L(D)$，幂律拟合远优于对数等其他形式（见图 23）。对于 $L(C)$，不包含仅 1 层的小模型，因为 1 到 2 层的转变会造成数据突变。$L(N)$ 拟合时也不包含仅 1 层的小模型，且排除未完全收敛的最大模型。即使包含这些模型，拟合参数变化很小，趋势在两端都能很好外推。

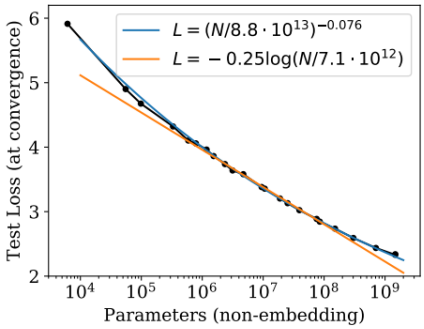

图 24: 参数量 $L(N)$ 与性能的关系用幂律拟合比用对数等其他函数拟合在定性上更好。

D.8 D.8 泛化与结构

图 24 显示，在固定总参数量时，泛化到其他数据分布与网络深度无关，仅依赖于训练分布上的性能。

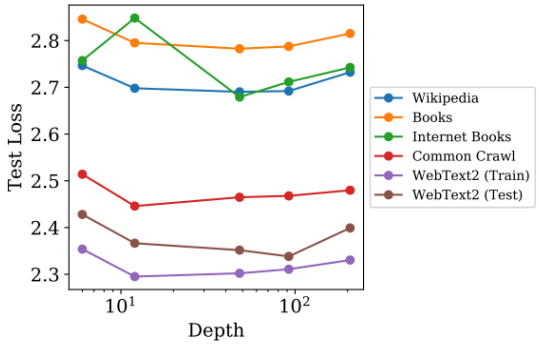

图 25: 我们在一系列数据集上评估了约 15 亿参数的模型。未观察到深度对泛化的影响；泛化性能主要取决于训练分布上的表现。12 层模型在 Internet Books 数据集上出现过拟合，图中展示了提前停止的性能；在其他实验中未见类似现象。