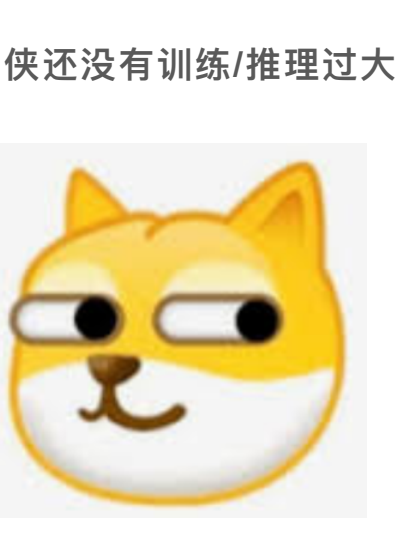
大模型炼丹无从下手？谷歌、OpenAI烧了几百万刀，总结出这些方法论…

原创 Yimin_饭煲 夕小瑶的卖萌屋 2021-10-19 12:05

文 | Yimin_饭煲

都1202年了，不会真有深度学习炼丹侠还没有训练/推理过大模型吧



“没吃过猪肉，还没见过猪跑吗？”在深度学习算力高度增长的今天，不论是学术界还是工业界的从业者，即使尚未达到从头预训练一个百亿级别参数模型的土豪程度，也必定尝试过在自己的科研/工作中使用大模型。训练大模型已经成为了刷排行榜SOTA、处理业务问题、提高工作受关注程度甚至是大厂“**秀肌肉**”的一个重要手段。如何高效地训练大模型，快速地推理大模型，方便地部署大模型也是产学研界所面临的关键挑战。

想刷SOTA想涨点，没有大模型，万万不行。可是，自己训练一个大模型，就一定能刷出SOTA和涨点么，那也未必？且不论数据质量、工程实现难度这些一个比一个令人头疼的问题，**单单是大模型结构和训练方法的设计，就已经让人抓狂了。** 作为一个训练大模型方面的小白，如果让我训练一个大模型，我最关心的问题是：

1. 为了解决我面对的问题，**我需要训练一个多大的模型**（十亿 / 百亿 / 千亿）？这决定了需要多少算力资源，能否在尽可能节省资源的前提下完成任务。
2. **我需要收集多少数据来“匹配”选择的模型**。如果我所需要处理的领域只能收集到几十 / 几百GB数据，我应该如何设置模型的参数量和训练方式?
3. 在决定好模型的参数量之后，**我应该在模型的什么部分“堆料”**？例如：我是应该将模型的层数垒高，还是应该将隐状态的长度加的更大，还是应该将注意力头个数目加多？在资源受限的情况下，我应该更侧重在什么部分增加参数?
4. 在预训练语言模型、微调语言模型、跨域迁移语言模型这些**常用的应用大语言模型方式下**，有没有什么**可以遵循的设计准则**?

幸运地是，针对以上令小白无从下手的问题，训练大模型无数的工业界土豪Google/DeepMind/OpenAI热心地为大家分享了自己的经验。研究者们将**研究模型大小、数据量、模型结构等因素如何影响模型性能**这一领域称为**深度学习的尺度定律**，本文将为大家解析尺度定律这一领域的三篇经典工作，**分析了在预训练、微调和跨域迁移三个常见的应用场景**中的尺度定律，希望大家在读完这篇推送后，能更好地设计和训练自己的大模型~

论文题目:

1. *Scaling Laws for Neural Language Models*
2. *Scaling Laws for Transfer*
3. *Scaling Efficiently: Insights from Pre-training and Fine-tuning Transformers*

预训练语言模型时的尺度定律

深度学习模型种类和任务五花八门，不妨从大模型数量最多、数据最充足、任务相对简单的预训练语言模型开始研究。OpenAI的工作 **Scaling Laws for Neural Language Models** 是尺度定律领域最早的研究。从预训练语言模型入手，想必也有OpenAI对GPT系列工作情有独钟的因素。这篇文章的主要结论是：

- 在预训练语言模型时，**模型的性能和模型的参数量明显正相关，而和模型的结构关系较小**（这对于其他类型的模型不一定成立）
- 模型的表现L（以交叉熵loss衡量）与模型的参数量N，数据集大小D和训练计算开销C都**有着明确的定量关系**$L(N) = (N_c/N)^{\alpha_N}$ $\alpha_N \sim 0.076, N_c \sim 8.8 \times 10^{13}$ (不包含embedding层参数量)$L(D) = (Dc/D)^{\alpha_D}$ $\alpha_D \sim 0.095, D_c \sim 5.4 \times 10^{13}$ (token数目) $L(C_{min}) = (C_c^{min}/C_{min})^{\alpha_C^{min}}$ $\alpha_C^{min} \sim 0.050,\ C_c^{min} \sim 3.1 \times 10^8$ (PF-days) $L(N,D) = [(N_c/N)^{\alpha_N/\alpha_D} + D_c/D]^{\alpha_D}$
- 当数据集较小而模型较大时会出现过拟合现象，作者指出了为避免过拟合的经验规律$D \geq (5 \times 10^3)N^{0.74}$，也即**当我们将模型大小扩大8倍时需要将数据集大小扩大5倍。**
- **大模型具有更好的样本有效性**，仅需要更少的优化步骤，更少的训练数据就能达到与小模型相同的表现
- **在固定训练开销的前提下，训练超大模型并且在收敛前停下能够达到最优的表现**（而不是将小模型训练到收敛），由下图可知，在训练开销限制增加的过程中，最优的训练配置是在模型大小上“堆料”，而不是增加Batch Size和训练步数。

  OpenAI的这一工作为随后的尺度定律研究工作提供了基本的框架，在实验设计上有许多可取之处，限于推送篇幅无法为大家一一介绍，有兴趣的小伙伴可以去看看原文哦！

微调语言模型时的尺度定律

OpenAI的工作告诉我们在预训练语言模型时，模型的参数量决定了模型的表现，而模型结构的影响很小。但是，**OpenAI研究的因变量只有语言模型的交叉熵，而在实际应用中，预训练语言模型更多地是为了在下游任务上进行微调和应用**。Google和DeepMind的研究者们，进一步探索了在微调语言模型时遵循的尺度定律，证明了模型的结构对下游任务微调的影响很大，并且发现了现有的T5模型并不是最优的结构。**作者们还壕无人性地发布了100个T5的Checkpoint!** 这篇文章的主要发现是：

- **模型在预训练阶段的表现基本仅依赖于模型的参数量，在微调阶段时的表现和模型结构关系很大**。在预训练阶段表现更好的模型（NL12-XXL), 在下游任务上的表现却明显低于NL32-XL。
- 在不同的计算开销条件下(Small, Base, Large)，所得到的最优模型配置是不相同的。因此，**在较小尺度下得到的最优模型配置未必能在较大的尺度上也是最优的**
- **模型的层数（NL）对下游任务上表现的影响很大**，而注意力头的数目（NH）、前向传播层（FF）的维度对下游任务上表现的影响相对较小。
- **作者提出了DeepNarrow策略**，也就是**在“堆料”时提高模型的层数**，而不需要将其他参数过于提高。按照此策略的模型**相对于标准版的T5模型更加“瘦高”**。从结果可以发现，尽管参数量和FLOPs更小，DeepNarrow策略的模型表现要优于T5标准模型（在Small,Base,Large,XL,XXL多个尺度上)
- 作者们在GLUE之外的多个和常识推理、自然语言生成任务有关的数据集上同样验证了DeepNarrow策略的有效性。

跨域迁移学习时的尺度定律

跨域迁移学习是大模型另一个极其重要的应用场景。例如，我想预训练一个Python数据上的大模型，然而Python数据的量毕竟是十分有限的。这时，**可以先利用大规模的文本数据训练一个大模型，再利用Python数据进行迁移学习，就能够得到一个强大的Python语言模型**。如下图所示，如果先在文本上预训练一个模型，之后在Python数据集上进行微调，就能够比从头训练的Python模型取得更好的效果（小数据场景下）。固定测试误差，定义从头训练Python模型所需要的数据量为D_E,从预训练好的文本模型迁移学习Python模型所需要的数据量为D_F，$D_T = D_E - D_F$就是从预训练文本模型迁移的“数据量”

研究者们采取了三种方式进行训练：直接在Python代码上训练/在文本上预训练在Python代码上微调/在文本和非Python的代码上预训练在Python代码上微调。通过大量地实验发现了尺度定律在迁移学习上的许多规律。

- D_T遵循和D_F以及模型参数量N的明确函数关系 $D_T = k(D_F)^{\alpha}(N)^{\beta}$，**其中$\beta = 0.38$,与预训练的语料分布无关，衡量了参数量对模型表现的影响。α和预训练的语料分布有关，预训练和微调的语料分布越接近，α越小**。在文本到Python的迁移过程中$\beta \approx 2\alpha$。
- **当下游的Python数据集较小时，从文本预训练模型迁移而来的Python模型有着更好的表现**。由于Python数据集较小，当从头训练的模型参数量进一步扩大时会受到过拟合的影响，使得表现难以进一步提升，而预训练过的文本模型能够随着参数量的提高进一步提升性能。
- **预训练模型在少数据场景下更加的计算高效**，也即仅需要更少的计算资源进行微调就能达到与从头训练相同的效果。
- 在小数据迁移场景下，作者发现微调模型的损失可以和前文中预训练模型的公式放在统一的框架中，得到公式$L \approx [(N_c/N)^{\alpha_N/\alpha_D} + D_c/(k(D_F)^{\alpha}(N)^{\beta})]^{\alpha_D}$这个公式可以通过将$D$替换为$D_T$得到。

尺度定律对炼丹师有什么用

了解完干货满满的尺度定律的三种应用后，大家会发现尺度定律的研究大多基于充分的实证实验和数据分析，往往需要耗费大量的算力资源，需要收集大量的数据以供实验。而作为学术界和工业界的普通炼丹师，也许无法获得如此充足的资源来研究如何训练自己的大模型。这里，我想谈谈我从这三篇“耗资巨大”的论文中学到的东西，顺便来回答文章开头提到的问题~

- 大模型的堆料思路：在设计大模型结构时，**优先考虑将模型的层数做深**，再考虑增加注意力头数、隐层尺寸等其他参数。
- 大模型的样本有效性：在训练数据较少的情况下，并不一定要采取小模型进行训练。可以采用大模型进行训练，在收敛前停下，仅需较少的训练步骤就能达到很好的效果。这是由于**大模型具有更好的样本有效性。**
- 大模型的迁移性：在目标任务领域数据量较小时，可以**采用通用文本上预训练过的大模型进行微调**，可能取得更优的效果。
- 大模型对数据的要求：模型的表现同时和数据量与模型参数量相关。为了避免过拟合，在增大模型参数量的同时需要增大数据量，但要求的数据量增长速度相对较慢。**当扩大模型大小8倍时只需要扩大数据量5倍。**
- 大模型的架构设计：如果不确定应该选取怎样的大模型结构和参数配置，可以参考Paper中已经给出的一些优秀的设计策略。

结语

尺度定律并非是在深度学习中提出的概念，而是物理学中的一个常用名词。**凡是可以用幂次关系表示的两个或多个物理量都称为满足尺度定律**。最早研究预训练语言模型的尺度定律论文 **Scaling Laws for Neural Language Models** 的第一作者Jared Kaplan是一位来自约翰霍普金斯大学的物理学教授，因此在这篇论文中可以看到很多类似于**物理学中的唯象**的研究思路。深度学习和物理学都需要对实验现象进行观察并总结出规律，也许深度学习也可以被看成一种新的“物理学”分支。希望将来能看到更多物理学家参与的深度学习工作，能给深度学习带来更多的理论和实践Insights~

[1] Scaling Laws for Neural Language Models.https://arxiv.org/abs/2001.08361

[2] Scaling Laws for Transfer.https://arxiv.org/abs/2102.01293

[3] Scaling Efficiently: Insights from Pre-training and Fine-tuning.https://arxiv.org/abs/2109.10686