

科研工具

1-文献检索

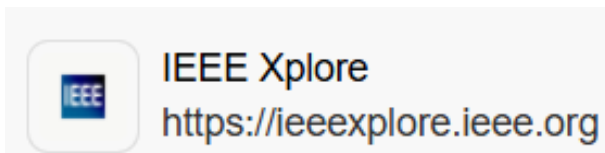
中文文献:



英文文献:

需要
VPN

IEEE

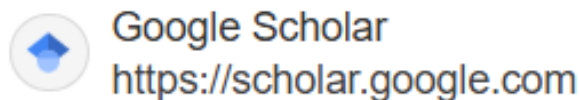


dblp--计算机领域文献



检索论文 可以加入关键词 .trans 找高质量文献

谷歌学术



2-文献阅读

小插件--沉浸式翻译



沉浸式翻译

3. MoE-LoRA with Layer-wise Allocation

Combining MoE and LoRA has shown promising results (Zadouri et al., 2023; Liu et al., 2023; Dou et al., 2023). However, most such efforts only replace experts with LoRA adapters under the MoE framework, and each layer has a fixed number of experts. Thus, some shortcomings of MoE may persist in these methods. For instance, experts in MoE may be redundant due to representational collapse or learned routing policy overfitting (Chen et al., 2023; Zoph et al., 2022). Inspired by this insight, we argue that the number of LoRA experts need not be the same across all Transformer layers.

We thus introduce a novel parameter-efficient tuning approach, called MoE-LoRA with Layer-wise Allocation (MoLA), which combines LoRA and MoE techniques with smart layer-wise expert allocation. As most LLMs use Transformer-based architectures, we study how MoLA should be applied to the Transformer model. Instead of allocating the same number of experts to all layers of the Transformer, MoLA uses different numbers of experts on different layers. In this section, we first describe the details of our architecture and then propose several layer-wise expert allocations based on different assumptions.

3.1. The MoLA Architecture

MoLA integrates LoRA adapters into the MoE framework so each layer may have a different number of experts. When training a pre-trained LLM with LoRA, instead of decomposing each weight matrix of a dense linear layer into a pair of low-rank matrices, we create multiple pairs of low-rank

matrices — each pair is called a LoRA expert. A router module is learned to route each input token to different LoRA experts. Given a Transformer model with m layers, we allocate N_j experts for layer j and have $\sum_{j=1}^m N_j$ experts in total. Specifically, given a pre-trained weight matrix $W_0^{jl} \in \mathbb{R}^{d_q \times d_v}$ from the module l in layer j , we create N_j pairs of low-rank matrices $\{A_i^{jl}\}_{i=1}^{N_j}, \{B_i^{jl}\}_{i=1}^{N_j}$. As in the case of LoRA, each matrix A_i^{jl} is initialized from a random Gaussian distribution. We set B_i^{jl} to zero, where $A_i^{jl} \in \mathbb{R}^{d_q \times r}, B_i^{jl} \in \mathbb{R}^{r \times d_v}$, and $r \ll \min(d_q, d_v)$. Then, a router S_i^{jl} with a trainable weight matrix $W_i^{jl} \in \mathbb{R}^{d_q \times N_j}$ is used to specify different LoRA experts for the input x . As in the original MoE, MoLA selects the top K experts for computation and applies the load balancing loss on each layer. Figure 1 shows an overview of the architecture. The mathematical representation is:

$$S_i^{jl}(x) = \frac{\text{TopK}(\text{Softmax}(W_i^{jl}x), K)_i}{\sum_{i=1}^K \text{TopK}(\text{Softmax}(W_i^{jl}x), K)_i} \quad (3)$$

$$h^{jl} = W_0^{jl}x + \sum_{i=1}^K S_i^{jl}(x)A_i^{jl}B_i^{jl}x \quad (4)$$

Eq. 3 represents the router with the input x and Eq. 4 mathematically shows the LoRA expert in MoLA, where h^{jl} is the output embedding. This MoLA architecture provides the flexibility to modify the number of experts for each Transformer layer. The next section addresses the question of how experts should be allocated in each layer.

3. 逐层分配的 MoE-LoRA

将 MoE 和 LoRA 相结合已经展现出有希望的结果 (Zadouri 等, 2023; Liu 等, 2023; Dou 等, 2023)。然而, 大多数此类尝试仅在 MoE 框架下用 LoRA 适配器替换专家, 并且每一层都有固定数量的专家。因此, 这些方法中可能仍然存在一些 MoE 的缺点。例如, MoE 中的专家可能由于表示坍塌或学习到的路由策略过拟合而变得冗余 (Chen 等, 2023; Zoph 等, 2022)。受此启发, 我们认为不同 Transformer 层中的 LoRA 专家数量不必相同。

我们因此提出了一种新颖的参数高效微调方法, 称为具有逐层分配的 MoE-LoRA (MoLA), 该方法结合了 LoRA 和 MoE 技术, 并采用智能的逐层专家分配策略。由于大多数 LLMs 使用基于 Transformer 的架构, 我们研究 MoLA 如何应用于 Transformer 模型。MoLA 不像传统方法那样为 Transformer 的所有层分配相同数量的专家, 而是为不同层分配不同数量的专家。在本节中, 我们首先描述我们的架构细节, 然后基于不同的假设提出几种逐层专家分配方案。

3.1. MoLA 架构

MoLA 将 LoRA 适配器集成到 MoE 框架中, 使得每一层可以拥有不同数量的专家。在使用 LoRA 训练一个预训练的 LLM 时, 我们不是将每个密集线性层的权重矩阵分解为一对低秩矩阵

而是创建多个低秩矩阵对——每一对被称为一个 LoRA 专家。一个路由模块被学习用来将每个输入 token 路由到不同的 LoRA 专家。给定一个具有 m 层的 Transformer 模型, 我们为第 j 层分配 N_j 个专家。具体来说, 给定第 j 层模块 l 的预训练权重矩阵 $W \in \mathbb{R}^{d_q \times d_v}$, 我们创建 N 对低秩矩阵 $\{A_i\}, \{B_i\}$ 。如同 LoRA 的情况, 每个矩阵 A 从随机高斯分布中初始化。我们将 B 设置为零, 其中 $A \in \mathbb{R}^{d_q \times r}, B \in \mathbb{R}^{r \times d_v}$, 且 $r \ll \min(d, d_v)$ 。然后, 使用一个具有可训练权重矩阵 $W \in \mathbb{R}^{d_q \times N}$ 的路由器 S 来指定输入 x 的不同 LoRA 专家。如同原始的 MoE, MoLA 会选择计算所需的前 K 个专家, 并在每一层应用负载均衡损失。图 1 展示了架构的概观。数学表示为:

$$S(x) = \frac{\text{TopK}(\text{Softmax}(Wx), K)}{\sum_{i=1}^K \text{TopK}(\text{Softmax}(Wx), K)} \quad (3)$$

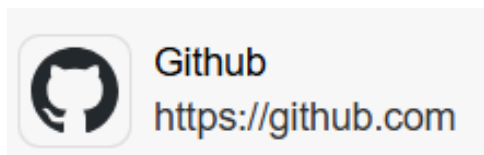
$$h = Wx + \sum_{i=1}^K S(x)ABx \quad (4)$$

式 3 表示以输入 x 为输入的路由器, 式 4 则数学上展示了 MoLA 中的 LoRA 专家, 其中 h 是输出嵌入。这种 MoLA 架构提供了灵活性, 可以针对每个 Transformer 层修改专家的数量。下一节将探讨每个层中专家应该如何分配的问题。

中英对照阅读

3-代码检索

Github--开源社区



4-AI大模型



ChatGPT

Claude Pro

5-论文撰写

在线LaTeX公式编辑

LaTeX公式编辑器 ver1.8.2

实时输入输出--支持多种输出

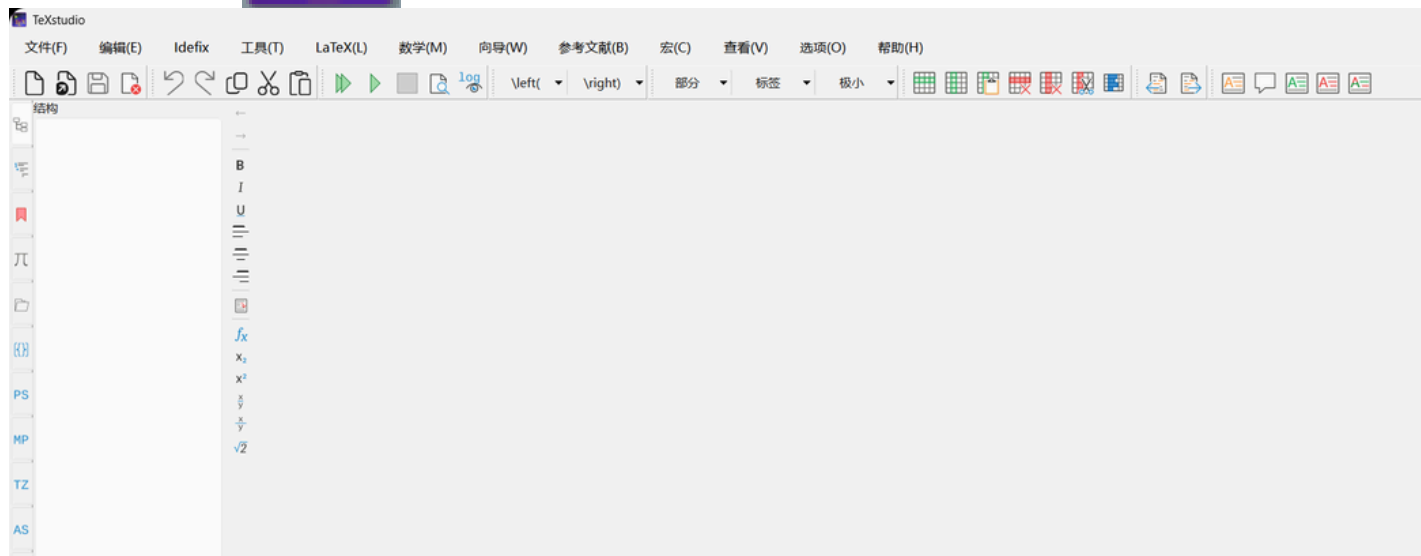


离线LaTeX



特别好用!

熟练后可以在某些情况下代替Word 美观 不存在排版问题



在线Overleaf

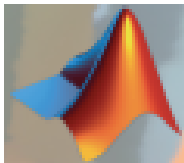


支持在线编辑 可以多人协作!

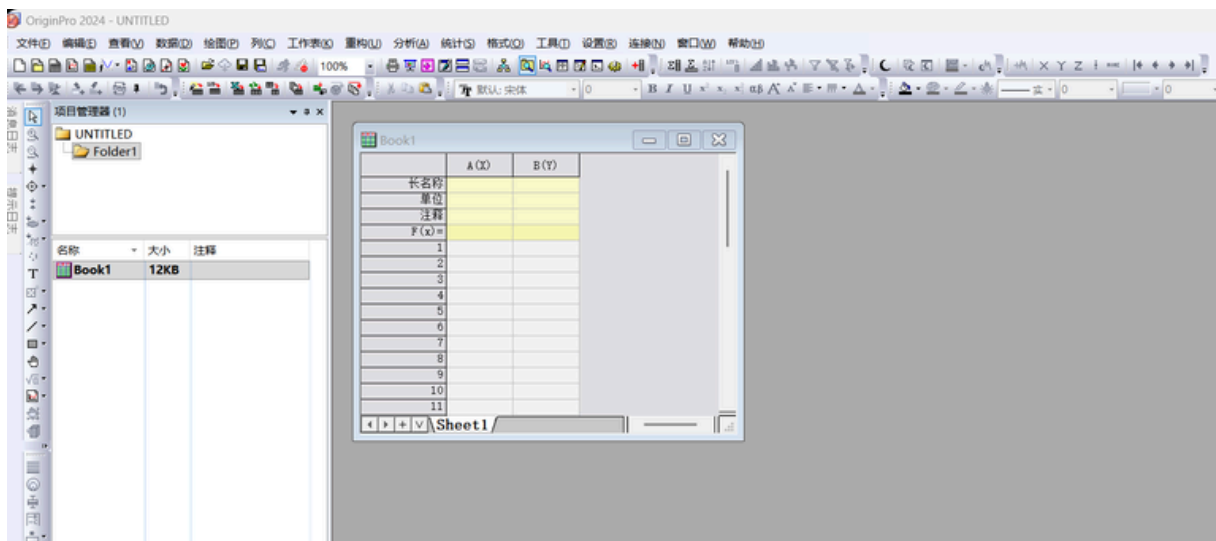
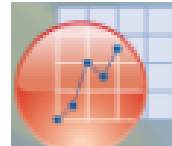
6-绘图

可以用编程软件进行绘制

Python/Matlab

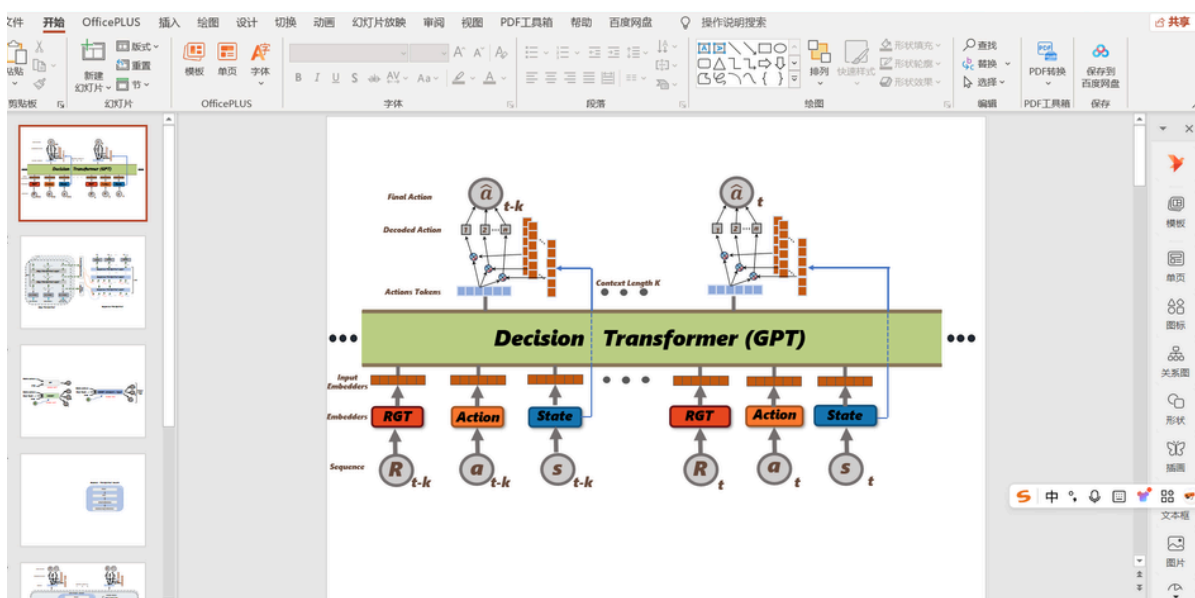


一些图的绘制可以用 Origin-- 导入数据 生成 图表
支持美化调整!



PPT - 流程图绘制

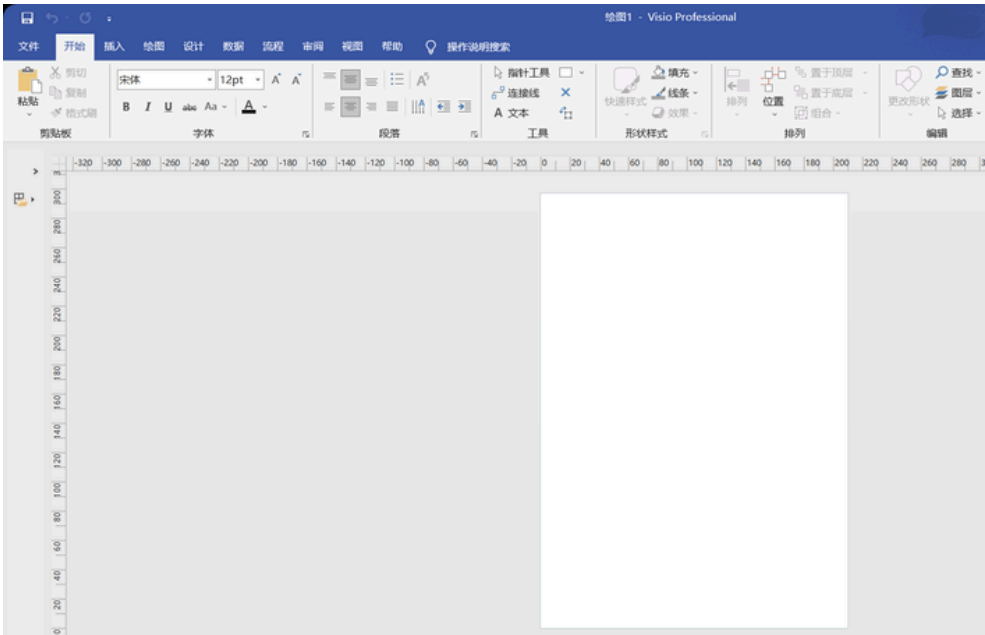
可以存一个自己的绘图模板--一些小插件可以直接复制使用!



Visio--矢量绘图工具 --流程图绘制



有些图片进行放大后会失真 Visio能很好规避这个问题



7-其他小工具

ColorSpace--颜色调色盘 – 16进制的色彩微调



阿里巴巴矢量图 -- 小插图 必备

