

IGU-LoRA: Adaptive Rank Allocation via Integrated Gradients and Uncertainty-Aware Scoring

IGU-LoRA：基于梯度积分与不确定性感知评分的
自适应秩分配方法

Published as a conference paper at ICLR 2026

大语言模型微调面临的现实挑战

大型语言模型LLM的参数规模扩展至数十亿级别，全参数微调在计算资源和内存消耗方面变得难以承受。参数高效微调PEFT是全参数微调的友好替代



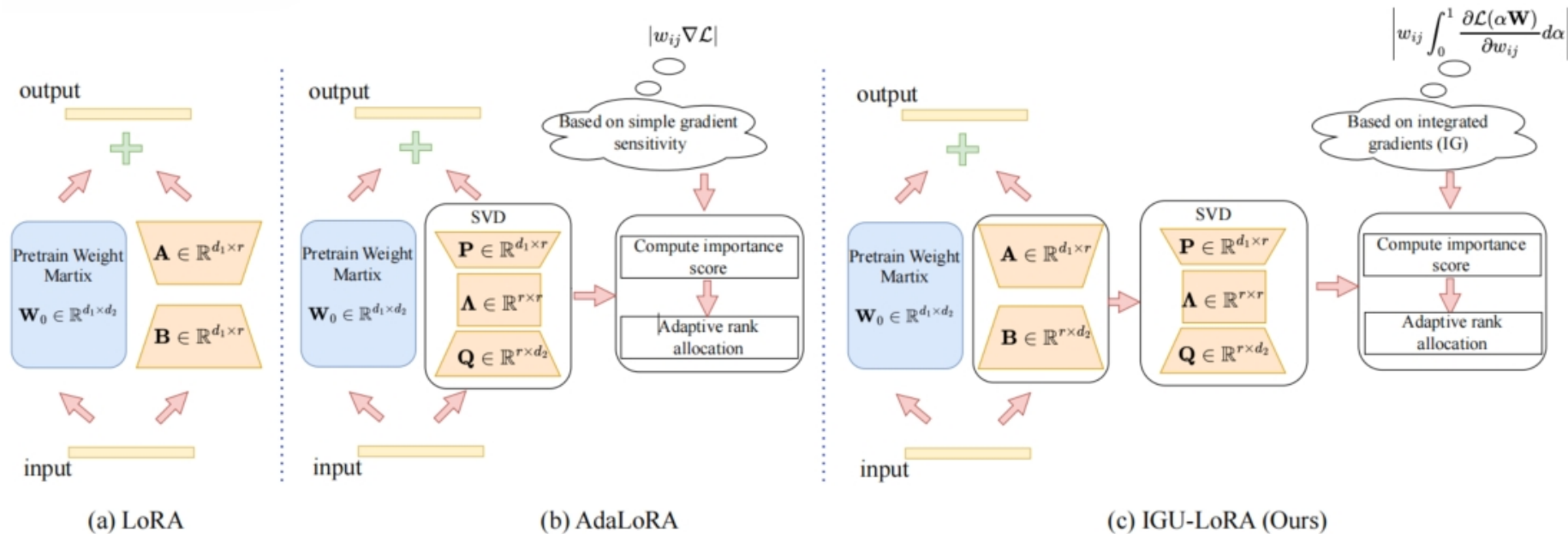
LoRA在所有层中采用固定统一秩值，忽略各层重要性存在显著差--提出逐层秩分配方案



AdaLoRA基于重要性评分进行秩分配，但依赖仅反映局部敏感性的瞬时梯度，可能导致评分结果不稳定且存在偏差。

本文提出 IGU-LoRA: (1)计算层内积分梯度 (IG) 敏感度并将其聚合为层级评分用于秩分配; (2)采用结合偏差追踪的指数移动平均算法构建不确定性感知机制，抑制噪声更新并优化秩选择。

LoRA AdaLoRA IGU-LoRA对比



基础准备

LoRA: 预训练参数矩阵 $W_0 \in \mathbb{R}^{d_1 \times d_2}$, $A \in \mathbb{R}^{d_1 \times r}$, $B \in \mathbb{R}^{r \times d_2}$

$$W = W_0 + AB$$

AdaLoRA: 左奇异矩阵 $P \in \mathbb{R}^{d_1 \times r}$, 右奇异矩阵 $Q \in \mathbb{R}^{r \times d_2}$

$$W = W_0 + \text{SVD}(AB) = W_0 + P\Lambda Q$$

每个奇异值 λ_i 定义了一个重要性评分 S_i , 同时考虑了奇异值及其相关参数 (即矩阵 P 第 i 列元素和矩阵 Q 第 i 行元素) 的敏感度:

$$S_i = \boxed{s_\lambda(\lambda_i)} + \frac{1}{d_1} \sum_{k=1}^{d_1} s_{snr}(P_{ki}) + \frac{1}{d_2} \sum_{k=1}^{d_2} \boxed{s_{snr}(Q_{ik})}$$

奇异值本身强度 $|\lambda_i|$

$s_{snr}(w)$: 参数 w 对损失函数 L 有多重要 $|w \nabla_w L|$

Q 中第 i 行参数的重要性平均值

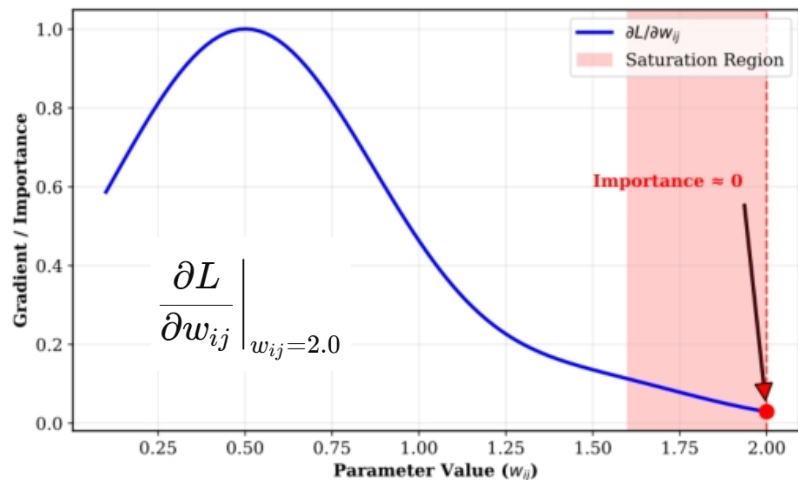
基础准备

$$\text{ReLU}(x) = \max(0, x)$$

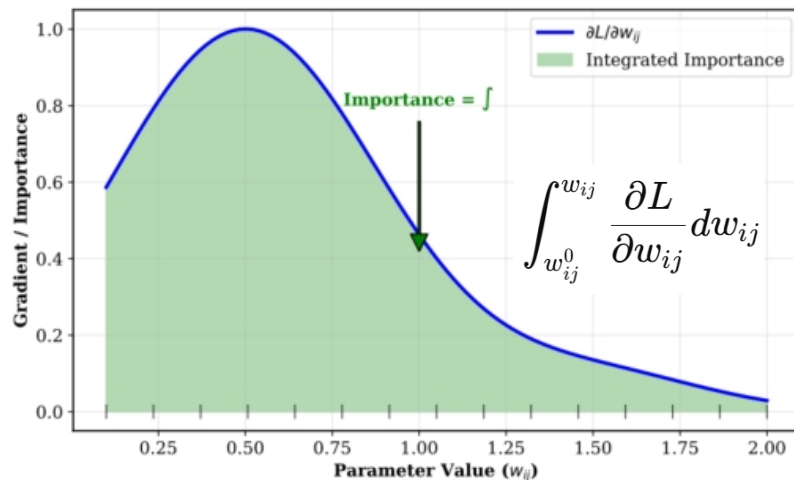
$$\frac{d}{dx} \text{ReLU}(x) = \begin{cases} 1, & x > 0 \\ 0, & x < 0 \end{cases}$$

基于梯度敏感性的简单方法的缺陷：

- 独立评估各参数权重，忽视了参数间的结构化交互关系，**结构可解释性不足**
- 仅反映参数对损失函数的**即时影响**，忽略其在整个训练过程中产生的累积或长期影响，可能导致估计结果不稳定
- 在大型语言模型中，ReLU等激活函数可能导致非活跃区域出现**梯度饱和**现象（即梯度信号完全消失），从而使重要性估计变得不可靠。



(a) Simple Gradient (Instantaneous)



(b) Integrated Gradients (IG)

梯度积分法

在深度学习的可解释性研究中，梯度积分法是一种广泛采用的归因方法，通过计算从基线输入到实际输入路径上的梯度积分来缓解梯度饱和问题。

该方法满足两个关键公理——完备性和敏感性，确保能定量反映每个输入特征对模型输出的贡献。

完备性(整体上能解释模型输出变化):

$$\sum_{i=1}^n IG_i(x) = F(x) - F(x')$$

敏感性:

$$x_i \neq x'_i$$

$$\Rightarrow F(x) \neq F(x') \Rightarrow IG_i(x) \neq 0$$

$$x_j = x'_j, \quad j \neq i$$

基于梯度积分法的重要性评分

IGU -LoRA使用梯度积分法整合损失函数关于模型参数的梯度值，以捕捉每个参数对训练损失函数的累积影响。

权重矩阵 $\Delta \mathbf{W}$ ，特定权重值 w_{ij} ，损失函数 $\mathcal{L}$ ，基线 $\Delta \mathbf{W}^{(0)}$ 的初始值为0，路径比例系数 α ，重要性得分如下：

$$\Delta w_{ij}^{(0)} \in \Delta \mathbf{W}^{(0)}$$

$$s_e(w_{ij}) = \left| (w_{ij} - \Delta w_{ij}^{(0)}) \int_{\alpha=0}^1 \frac{\partial \mathcal{L}(\alpha(\Delta \mathbf{W} - \Delta \mathbf{W}^{(0)}))}{\partial w_{ij}} d\alpha \right| = \left| w_{ij} \int_{\alpha=0}^1 \frac{\partial \mathcal{L}(\alpha \Delta \mathbf{W})}{\partial w_{ij}} d\alpha \right|$$

近似求解：将路径 $[0,1]$ 离散化为 N 个等长区间（节点 $\alpha_k = k/N$, $k = 1, \dots, N-1$ ），并应用梯形法则进行计算。

$$\hat{s}_e(w_{ij}) \approx \frac{|w_{ij}|}{2N} \left| \frac{\partial \mathcal{L}(0)}{\partial w_{ij}} + 2 \sum_{k=1}^{N-1} \frac{\partial \mathcal{L}(\alpha_k \Delta \mathbf{W})}{\partial w_{ij}} + \frac{\partial \mathcal{L}(\Delta \mathbf{W})}{\partial w_{ij}} \right|.$$

基于梯度积分法的重要性评分

$$\hat{s}_e^p(w_{ij}) \approx \frac{|w_{ij}|}{2N} \left| \frac{\partial \mathcal{L}(0)}{\partial w_{ij}} + 2 \frac{\partial \mathcal{L}(\alpha_k \Delta \mathbf{W})}{\partial w_{ij}} + \frac{\partial \mathcal{L}(\Delta \mathbf{W})}{\partial w_{ij}} \right|$$

$$s_{agg}(w_{ij}) = \frac{1}{M} \sum_{p=1}^M \hat{s}_e^p(w_{ij}).$$

不确定性感知评分

研究表明, 随机采样和复杂的训练会导致上页计算的重要性分数估计值存在**高方差**, 从而削弱其可靠性。

为解决这一问题, 引入两种互补机制: 平滑敏感度法和不确定性量化法。

第 t 个 epoch 平滑后的敏感度

$$\bar{s}_e^{(t)}(w_{ij}) = \beta_1 \bar{s}_e^{(t-1)}(w_{ij}) + (1 - \beta_1) s_{agg}^{(t)}(w_{ij})$$

控制历史信息保留多少

当前 epoch 算出来的原始重要性

$$\bar{U}^{(t)}(w_{ij}) = \beta_2 \bar{U}^{(t-1)}(w_{ij}) + (1 - \beta_2) \left| s_{agg}^{(t)}(w_{ij}) - \bar{s}_e^{(t)}(w_{ij}) \right|$$

最终重要性评分定义为:

当前原始重要性分数和当前平滑重要性分数之间相差

$$s_{snr}^{(t)}(w_{ij}) = \text{SNR}_t = \frac{\bar{s}_e^{(t)}(w_{ij})}{\bar{U}^{(t)}(w_{ij}) + \epsilon}$$

算法流程

Algorithm 1 IGU-LoRA

Input: Dataset $\mathcal{D}$; the number of total iterations T ; a pre-trained parameter matrix $\mathbf{W}_0 \in \mathbb{R}^{d_1 \times d_2}$ of a large language model, number of mini-batches M ; budget of remaining singular values b ; randomly initialize trainable low-rank matrices $\mathbf{A} \in \mathbb{R}^{d_1 \times r}$ and $\mathbf{B} \in \mathbb{R}^{r \times d_2}$; hyperparameters β_1, β_2 .

- 1: **for** $t = 1$ to T **do**
 - 2: **for** $p = 1$ to M **do**
 - 3: Sample a mini-batch from $\mathcal{D}$ and train $\mathbf{A}$ and $\mathbf{B}$.
 - 4: Perform SVD on the matrix product $\mathbf{AB}$ to obtain $\mathbf{P}\mathbf{\Lambda}\mathbf{Q} = \text{SVD}(\mathbf{AB})$, where $\mathbf{\Lambda} = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_r\}$.
 - 5: Compute the $\hat{s}_e^p$ in Eq. (6) for every parameter in $\mathbf{P}, \mathbf{Q}$.
 - 6: **end for**
 - 7: Compute the aggregated importance score s_{agg} in Eq. (7) for every parameter in $\mathbf{P}, \mathbf{Q}$.
 - 8: Compute the $\bar{s}_e^{(t)}$ in Eq. (10) and $\bar{U}^{(t)}$ in Eq. (11) for every parameter in $\mathbf{P}, \mathbf{Q}$.
 - 9: Update the final importance score $s_{snr}^{(t)}$ in Eq. (12).
 - 10: Compute the importance score of each singular value S_i in Eq. (3) for $\mathbf{P}\mathbf{\Lambda}\mathbf{Q}$.
 - 11: Find the top b eigen value: $\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_b$ by importance score S_i .
 - 12: Set $\tilde{\mathbf{\Lambda}} \leftarrow \text{diag}(\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_b, 0, \dots, 0)$.
 - 13: Update $\mathbf{A} \leftarrow \mathbf{P}_{:, \pi_{1:b}} \tilde{\mathbf{\Lambda}}^{1/2}$, $\mathbf{B} \leftarrow \tilde{\mathbf{\Lambda}}^{1/2} \mathbf{Q}_{\pi_{1:b}, :}$; $\triangleright$ The subscript π denotes the index set obtained by sorting the columns of $\mathbf{P}$ and $\mathbf{Q}$ in descending order; $\pi_{1:k}$ represents the indices of the first b selected columns; $\mathbf{P}_{1:\pi_{1:b}}$ represents selecting the first b columns according to the order defined by π .
 - 14: **end for**
- Output:** $\mathbf{W} = \mathbf{W}_0 + \mathbf{AB}$
-

实验对比

RoBERTa-large模型 GLUE基准测试

Method	# Params	CoLA (mcc)	SST-2 (acc)	MRPC (acc-f1)	QQP (acc-f1)	STS-B (corr)	MNLI (acc)	QNLI (acc)	RTE (acc)	Avg.
Full FT	355M	69.19	95.63	89.46	91.10	91.60	90.01	94.03	86.94	88.50
Housbly-Adapter	0.35M	67.80	94.38	89.75	89.41	91.08	90.28	93.52	84.36	87.57
P-tuning v2	0.31M	67.35	93.13	88.49	88.63	90.41	89.19	91.94	82.42	86.45
(IA) ³	0.33M	68.62	93.82	89.54	89.78	90.84	89.87	92.60	82.75	87.23
SSP	0.36M	69.89	94.96	90.08	90.14	91.37	90.42	94.16	84.88	88.24
LoRA	0.33M	68.71	94.84	89.71	90.26	91.63	90.34	93.87	85.56	88.12
AdaLoRA	0.35M	70.04	95.62	<u>90.34</u>	90.37	91.57	90.18	<u>94.29</u>	87.06	88.68
DoRA	0.33M	70.26	<u>95.80</u>	90.12	90.16	<u>91.68</u>	<u>90.43</u>	<u>94.17</u>	87.38	88.75
AutoLoRA	0.34M	<u>70.47</u>	95.53	90.26	90.31	91.52	90.26	94.08	<u>87.64</u>	<u>88.76</u>
IGU-LoRA	0.33M	71.93	96.17	90.69	<u>90.68</u>	91.95	90.76	94.72	88.46	89.42

实验对比

Qwen-2.5-0.5B模型 数学推理和常识推理任务

Method	# Params	BoolQ (acc)	ARC-e (acc)	ARC-c (acc)	GSM8K (acc)	AQuA (acc)	Avg.
Full FT	494.0M	<u>81.74</u>	74.82	<u>54.98</u>	34.64	<u>48.72</u>	<u>58.98</u>
Housbly-Adapter	9.0M	78.36	71.04	53.26	28.67	42.85	54.84
LoRA	8.8M	78.94	72.78	54.38	31.42	45.33	56.57
AdaLoRA	8.9M	80.32	73.90	54.23	33.27	46.58	57.67
GoRA	8.8M	79.24	71.20	51.91	32.07	45.81	56.04
IGU-LoRA	8.8M	82.45	<u>74.62</u>	55.67	<u>34.16</u>	48.93	59.17

Qwen-2.5-0.5B模型
BoolQ任务

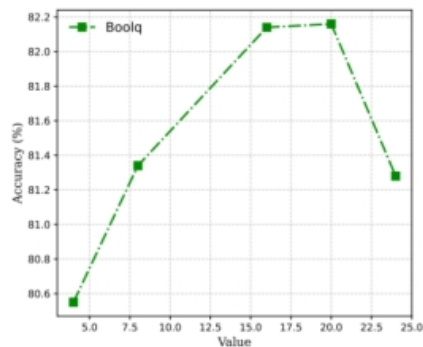
Method	Training		Inference	
	Time cost (h)	GPU Mem (GB)	Speed (it/s)	GPU Mem (GB)
LoRA	0.42	10.21	5.50	10.3
AdaLoRA	0.73	10.60	5.21	10.4
DoRA	0.95	9.53	5.30	10.3
IGU-LoRA	0.87	10.32	5.23	10.3

消融实验

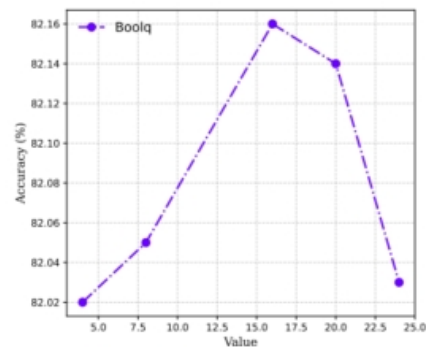
Qwen-2.5-0.5B模型
在BoolQ和GSM8K任务

Method	BoolQ	GSM8K	Avg.
IGU-LoRA-1 (w/o α)	81.87	33.76	57.82
IGU-LoRA-2 ($N=10$)	82.14	<u>33.95</u>	<u>58.05</u>
IGU-LoRA-3 ($N=4$)	82.02	33.83	57.93
IGU-LoRA-4 ($s_e = \bar{s}_e \cdot \bar{U}$)	<u>82.28</u>	33.69	57.99
IGU-LoRA	82.45	34.16	58.31

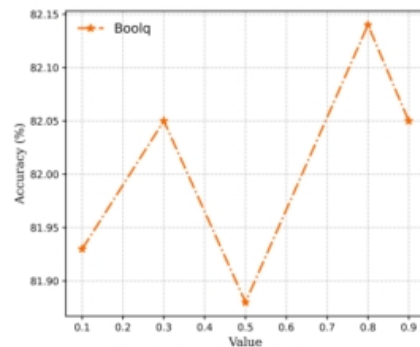
不同超参数在BoolQ数据集上微调Qwen2.5-0.5B模型



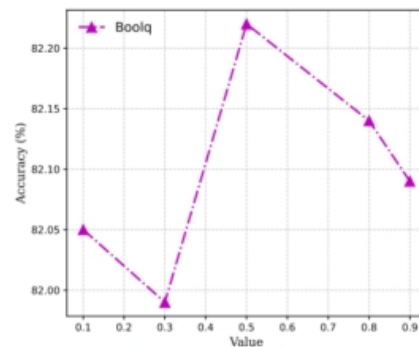
(a) M parameter



(b) N parameter



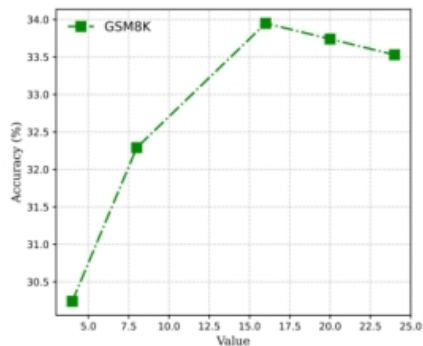
(c) β_1 coefficient



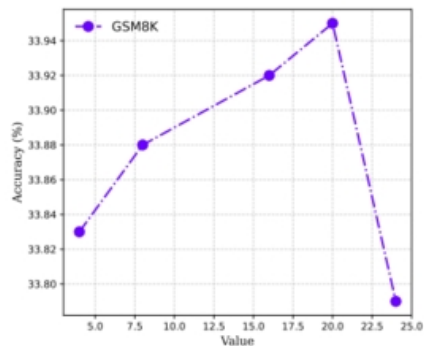
(d) β_2 coefficient

消融实验

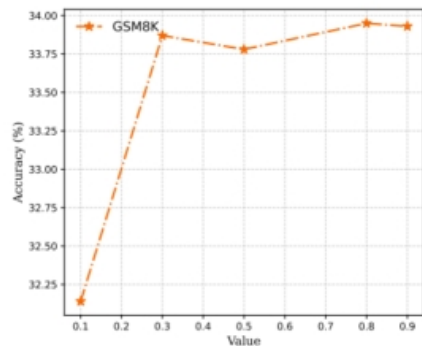
不同超参数在GSM8K数据集上微调Qwen2.5-0.5B模型



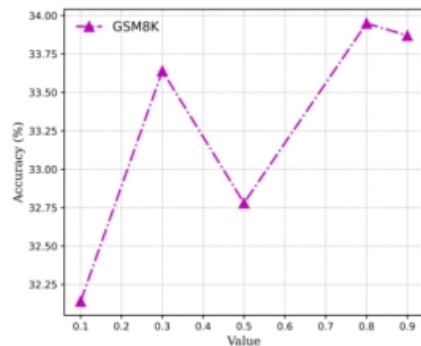
(a) M parameter



(b) N parameter



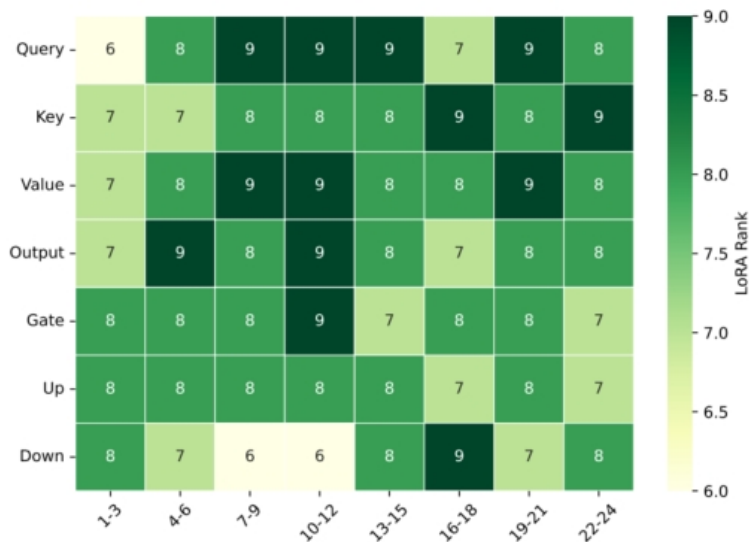
(c) β_1 coefficient



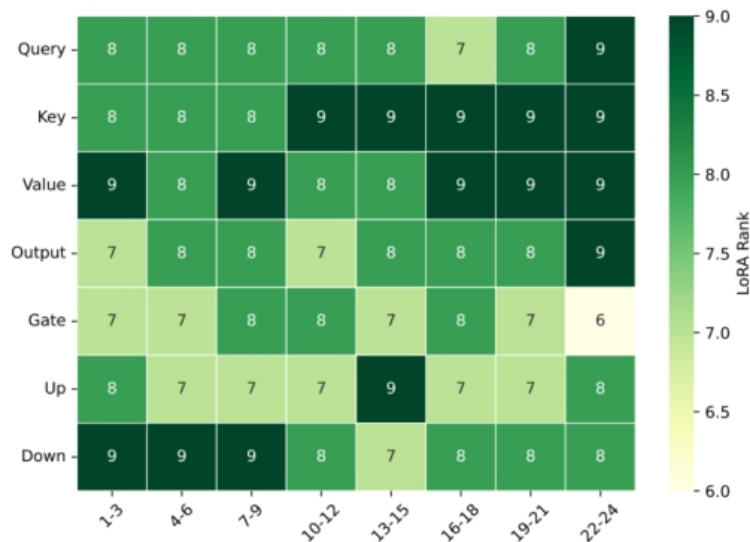
(d) β_2 coefficient

IGU -LoRA中排名分配的可视化呈现

在针对BoolQ和GSM8K任务完成微调后，使用 IGU -LoRA算法在Qwen-2.5-0.5B模型架构上进行排名分配。

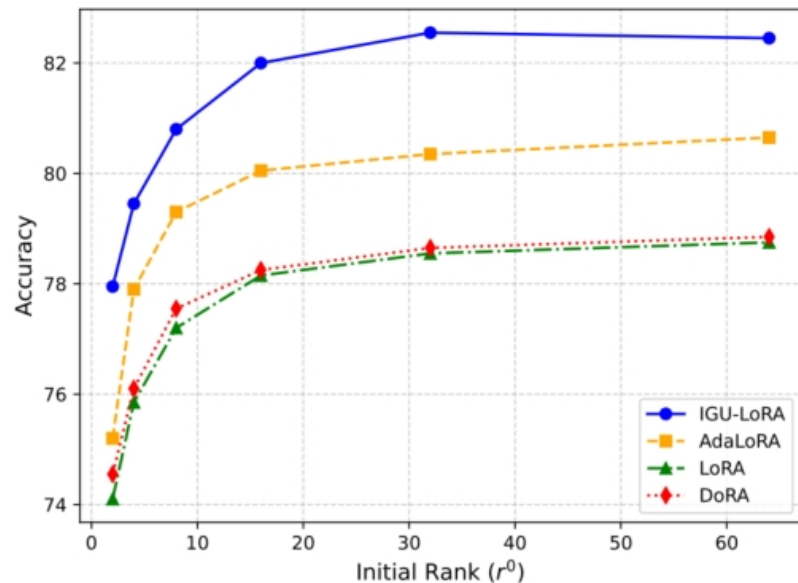


(a) BoolQ task

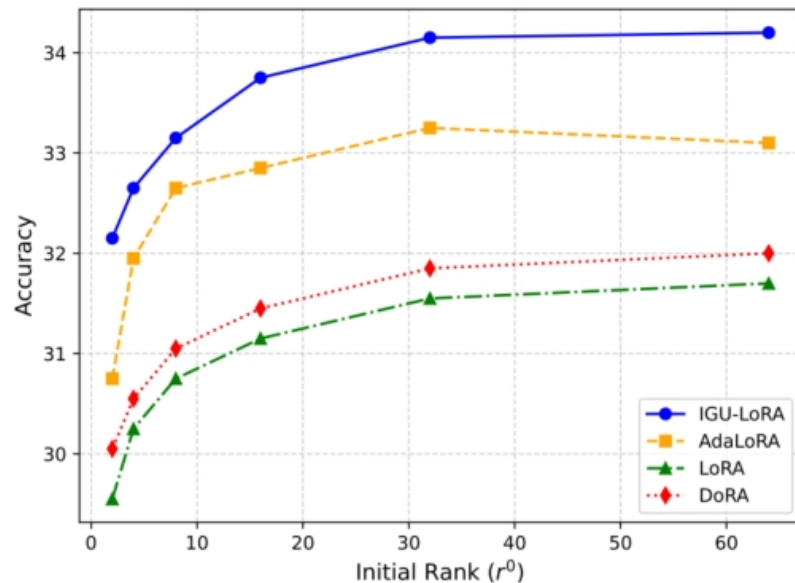


(b) GSM8K task

秩预算比较分析



(a) BoolQ task



(b) GSM8K task

不同模型架构下的 PEFT 方法性能对比

左图：Llama-2-7B模型下的GLUE准确率

右图：Llama-3-8B和DeepSeek-R1-Distill-Qwen-2.5-7B模型下BoolQ与GSM8K的准确率

Llama-2-7B							
Method	# Params	SST-2	RTE	QNLI	BoolQ	COPA	Avg.
Full FT	6738M	95.83	92.11	92.54	87.30	93.01	92.16
Adapter P-tuning v2 SSP	21.2M	94.15	82.12	93.10	87.03	91.10	89.50
	20.9M	93.42	79.62	92.64	84.73	90.30	88.14
	40.0M	94.14	83.11	93.10	87.11	91.65	89.82
LoRA	20.0M	94.12	83.37	93.10	<u>87.34</u>	91.33	89.85
AdaLoRA	20.0M	94.12	83.51	<u>93.20</u>	87.11	91.62	89.91
DoRA	40.0M	94.24	84.12	<u>91.23</u>	85.51	90.01	89.02
IGU-LoRA	40.0M	<u>94.34</u>	<u>84.33</u>	93.33	88.11	<u>92.10</u>	<u>90.44</u>

Llama-3-8B			
Method	BoolQ	GSM8K	Avg.
LoRA	88.48	73.54	81.01
AdaLoRA	<u>91.65</u>	<u>75.82</u>	<u>83.74</u>
DoRA	<u>88.07</u>	<u>74.75</u>	<u>81.41</u>
IGU-LoRA	93.33	77.63	85.48
DeepSeek-R1-Distill-Qwen-2.5-7B			
Method	BoolQ	GSM8K	Avg.
LoRA	88.38	74.60	81.49
AdaLoRA	90.54	73.30	<u>81.92</u>
DoRA	<u>88.48</u>	69.52	<u>79.00</u>
IGU-LoRA	92.82	<u>74.28</u>	83.55

THANKS

