

# 预训练之训练技术

《大语言模型》编写团队：唐天一

➤ 现有大语言模型的预训练优化设置

模型	批次大小	学习率 (预热 → 峰值 → 衰减)	优化器	精度 类型	权重 衰减	梯度 裁剪
GPT-3	32K → 3.2M	预热 → $6 \times 10^{-5}$ → 余弦	Adam	FP16	0.1	1.0
OPT	2M	预热 → $1.2 \times 10^{-4}$ → 手动	AdamW	FP16	0.1	-
PaLM	1M → 4M	$1 \times 10^{-2}$ → 平方根倒数	Adafactor	BF16	$lr^2$	1.0
BLOOM	4M	预热 → $6 \times 10^{-5}$ → 余弦	Adam	BF16	0.1	1.0
LLaMA-2	4M	预热 → $1.5 \times 10^{-4}$ → 余弦	AdamW	-	0.1	1.0
Baichuan-2	-	预热 → $1.5 \times 10^{-4}$ → 余弦	AdamW	BF16	0.1	0.5
Qwen-1.5	4M	预热 → $3 \times 10^{-4}$ → 余弦	AdamW	BF16	0.1	1.0
InternLM-2	5M	预热 → $3 \times 10^{-4}$ → 余弦	AdamW	-	0.1	-
Falcon	预热 → 4M	预热 → $1.25 \times 10^{-4}$ → 余弦	AdamW	BF16	0.1	0.4
DeepSeek	18M	预热 → $3.2 \times 10^{-4}$ → 余弦	AdamW	BF16	0.1	1.0
YuLan	4.5M	预热 → $3 \times 10^{-4}$ → 余弦	Adam	BF16	0.1	1.0
GLM-130B	0.4M → 8.25M	预热 → $8 \times 10^{-5}$ → 余弦	AdamW	FP16	0.1	1.0

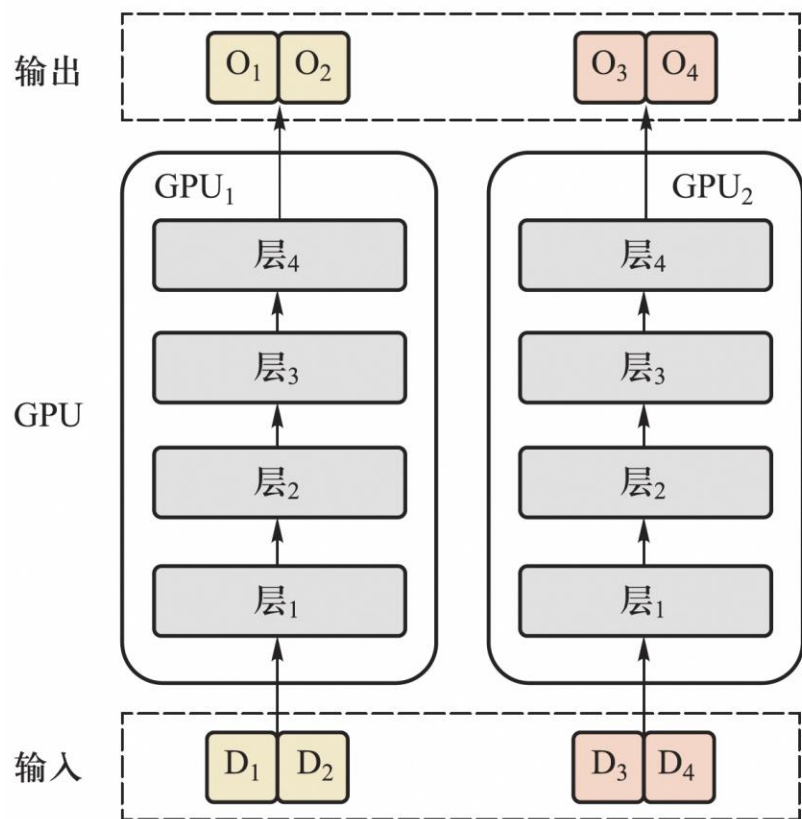
动态批次调整

学习率预热与衰减

混合精度训练

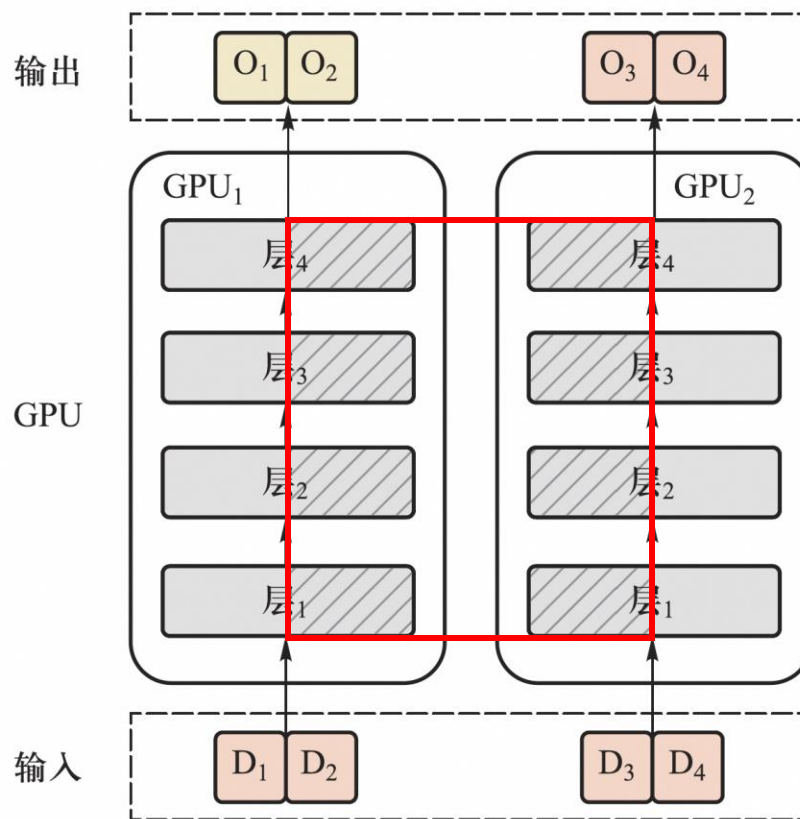
# 3D 并行训练

## 数据并行



将模型复制、数据平均分配  
分别计算后合并梯度，统一更新

## 零冗余优化器 (ZeRO)



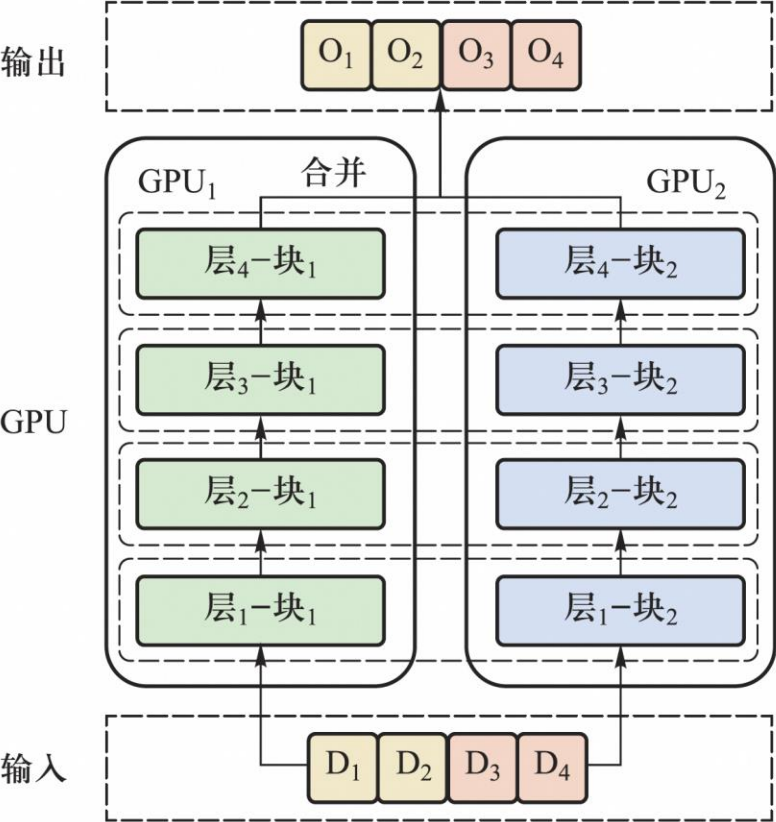
将模型均分到不同卡，缓解冗余  
计算时动态从对应卡读取参数

划线部分不长期存储  
需要时动态获取

# 3D 并行训练



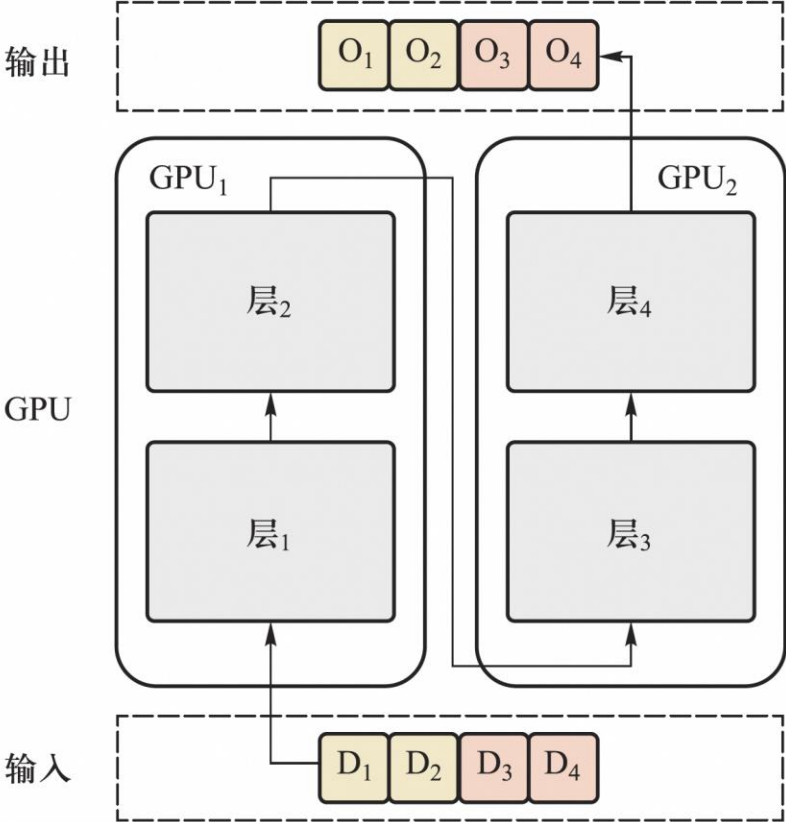
张量并行



将模型参数 $W$ 拆分为 $W_1$ 和 $W_2$

在两张卡并行计算 $XW_i$ 后拼接得到输出

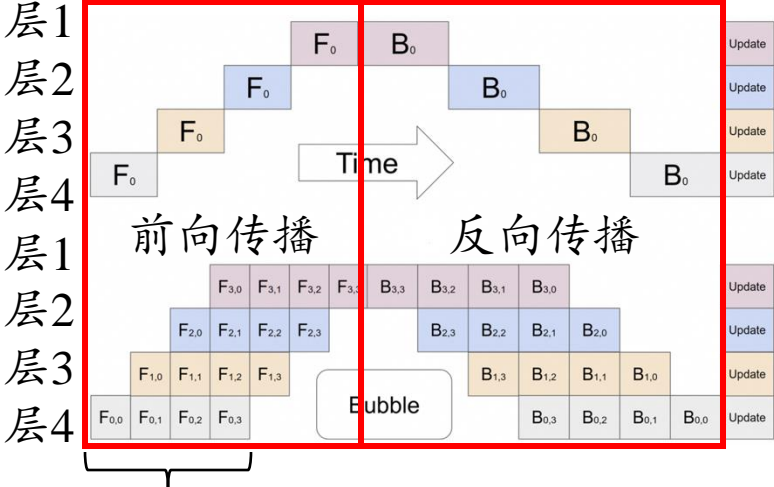
流水线并行



将连续的层分配到不同卡

依次经过每卡串行计算

需要与梯度累积联合使用，  
以达到流水线效果

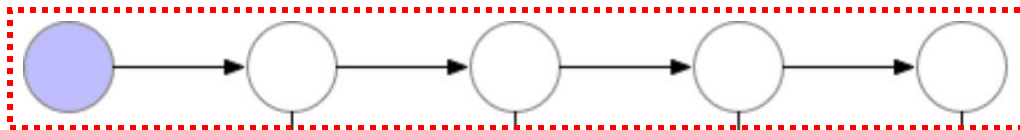


GPipe: Efficient Training of Giant Neural Networks  
using Pipeline Parallelism, *NeurIPS 2024*

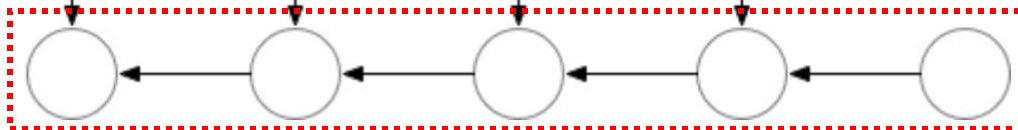
# 激活重计算

- 激活值是前向传播的结果，需要在反向传播时参与梯度计算

前向传播（激活值）

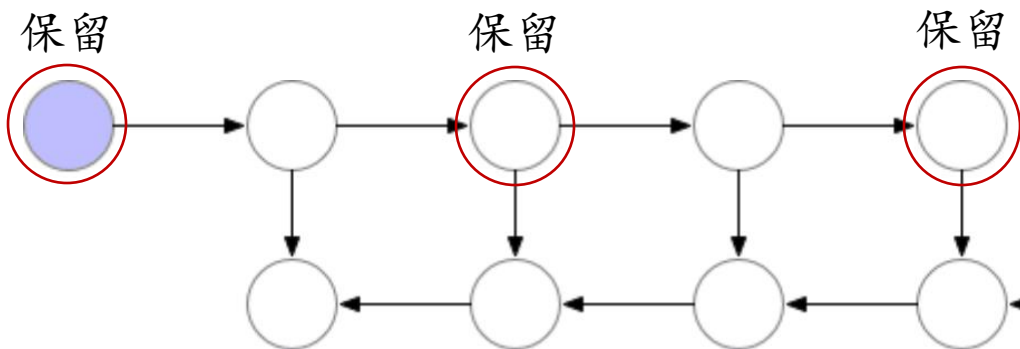


反向传播（梯度）

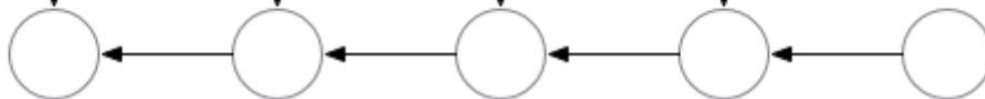


- 保存激活值需要占用大量显存

前向传播（激活值）



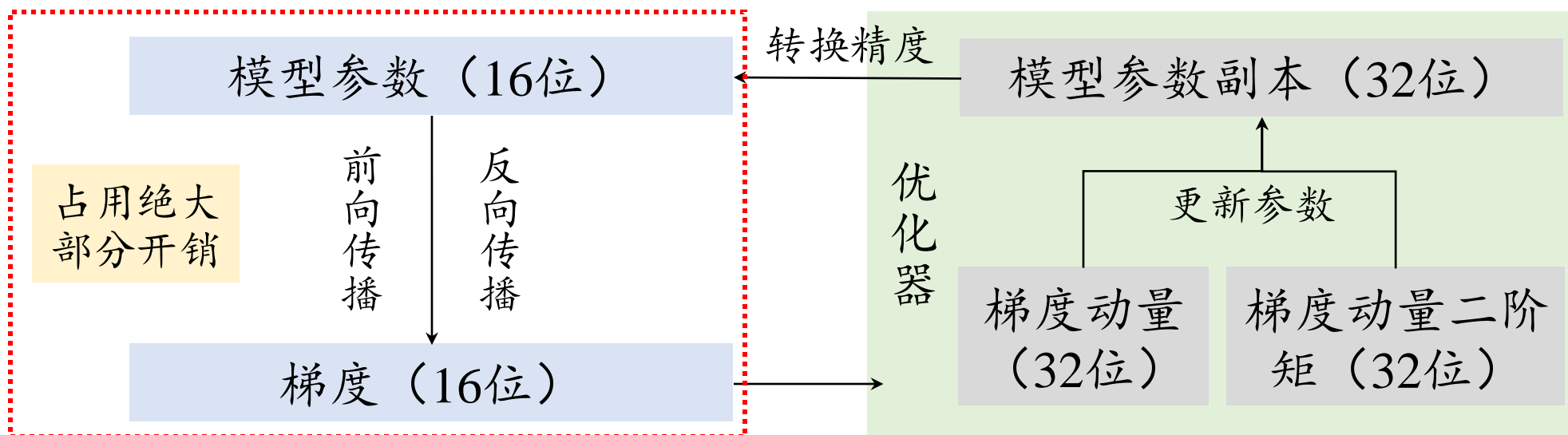
反向传播（梯度）



- 激活重计算：保留部分激活值，反向传播时重新计算其他激活值

# 混合精度训练

- 同时使用 32 位浮点数和 16 位浮点数计算
- 显存减半、效率翻倍
- DeepSeek-V3 使用 FP8 混合训练，进一步减少密集型计算开销





# 模型参数量估计



- LLaMA 模型参数量如何计算？
  - Transformer 层数  $L$
  - 隐含层维度  $H$
  - 中间状态维度  $H'$
  - 注意力头数  $N$
  - 上下文窗口长度  $T$
  - 词表大小  $V$

	Llama				Llama 2				Llama 3			
	Feb 2023				Jul 2023				Apr 2024			
# params	7B	13B	33B	65B	7B	13B	34B	70B	8B			70B
# training tokens	1T	1T	1.4T	1.4T	2T	2T	2T	2T	15T			15T
hidden embed dim	4096	5120	6656	8192	4096	5120		8192	4096			8192
# attn heads	32	40	52	64	32	40		64	32			64
# attn layers	32	40	60	80	32	40		80	32			80
attention	MHA	MHA	MHA	MHA	MHA	MHA	GQA	GQA	GQA			GQA
# kv heads	32	40	52	64	32	40		8	8			8
nlp intermediate size	11008	13824	17920	22016	11008	13824		28672	14336			28672
context	2048				4096				8192			
tokenizer	BPE sentencepiece				BPE sentencepiece				BPE tiktoken			
token vocabulary	32000				32000				128256			
fine-tuned models	-				Llama-2-Chat (Jul 2023)...				Llama-3-Instruct (Apr 2024)			

# 模型参数量估计

## ➤ 输入词嵌入层

➤  $E \in \mathbb{R}^{V \times H}$ , 参数量  $VH$

## ➤ 输出层

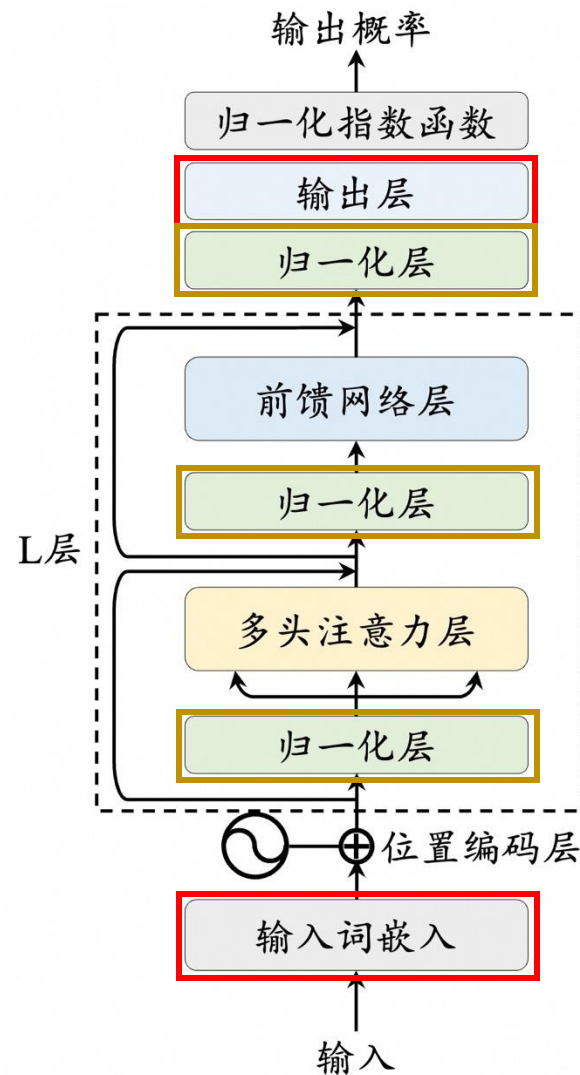
➤  $O = \text{softmax}(YW)$ ,  $W \in \mathbb{R}^{H \times V}$

➤ 参数量  $VH$

## ➤ 归一化层 (2L+1层)

➤  $x = \frac{x}{\text{RMS}(x)} \cdot \gamma$ ,  $\gamma \in \mathbb{R}^H$

➤ 参数量  $(2L + 1)H$





## ➤ 多头注意力层 (L层)

$$Q = XW^Q$$

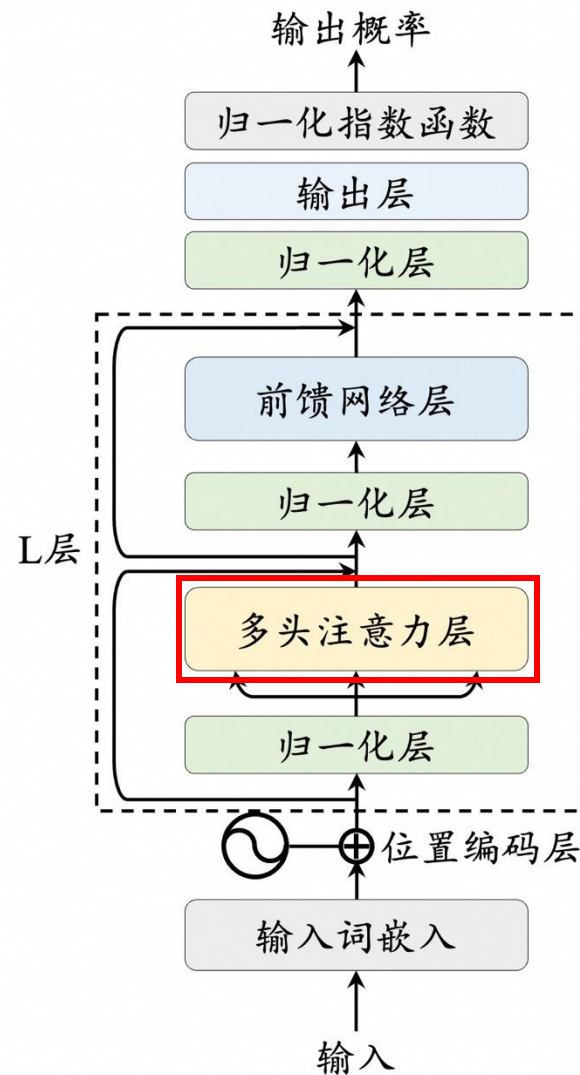
$$K = XW^K$$

$$V = XW^V$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{D}}\right)V$$

$$\text{MHA} = \text{Concat}(\text{head}_1, \dots, \text{head}_N)W^O$$

➤  $W^Q, W^K, W^V, W^O \in \mathbb{R}^{H \times H}$ , 参数量共  $4LH^2$



# 模型参数量估计

## ➤ 前馈网络层 (L层)

$$G = XW^G$$

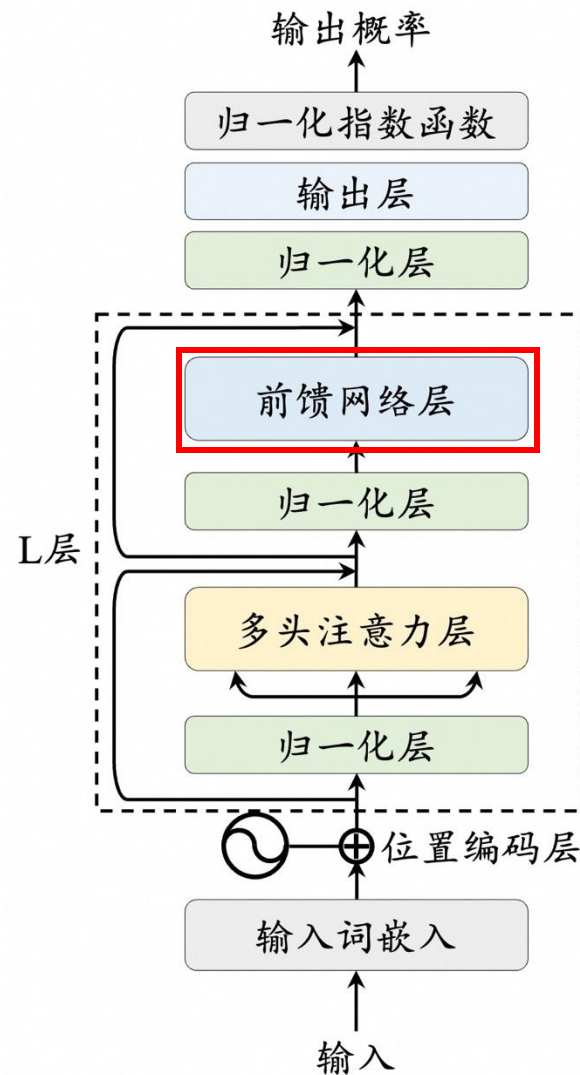
$$U = XW^U$$

$$D = \text{Swish}(G) \odot U$$

$$X = DW^D$$

➤  $W^G, W^U \in \mathbb{R}^{H \times H'}, W^D \in \mathbb{R}^{H' \times H}$

➤ 参数量  $3LHH'$



➤ LLaMA的参数量为：

$$\underbrace{2VH}_{\text{输入输出}} + \underbrace{H}_{\text{归一化}} + L \left( \underbrace{4H^2}_{\text{注意力}} + \underbrace{3HH'}_{\text{前馈网络}} + \underbrace{2H}_{\text{归一化}} \right)$$

➤ LLaMA 7B

➤  $V = 32000, L = 32, H = 4096, H' = 11008$

➤ 参数量为6,738,415,616

➤ LLaMA 65B

➤  $V = 32000, L = 80, H = 8192, H' = 22016$

➤ 参数量为65,285,660,672

# 训练显存估计

- 训练模型需要多少显存？
- 训练时显存占用包括以下三部分
  - 模型参数与优化器的显存占用
  - 激活值的显存占用
  - 其他显存占用

田渊栋等人新作：突破内存瓶颈，让一块4090预训练7B大模型



编辑：陈萍、大盘鸡只用 24G 显存，消费级 GPU 就能搞定大模型了。上...

11 month(s) ago

24GB单卡全量微调Llama 3-8B，仅需添加一行代码



举例来说，当训练一个拥有70亿个参数的模型时，以上参数将占用超过1...

10 month(s) ago

只需单卡RTX 3090，低比特量化训练就能实现LLaMA-3 8B全参微调



例如，即便是相对较小的 7B 规模模型，也可能需要高达 60GB 的 GPU...

9 month(s) ago

LLaMA微调显存需求减半，清华提出4比特优化器



但相比之下，单个 gpu 的显存大小却增长缓慢，这让显存成为了大模型...

2023/9/8

用FP8训练大模型有多香？微软：比BF16快64%，省42%内存



在训练前向和后向传递中全程使用 FP8 格式，极大降低了系统的计算，显...

2023/11/2

650亿参数，8块GPU就能全参数微调：邱锡鹏团队把大模型门槛打下来了



全参数微调的显存使用量和推理一样多，大模型不再只是大型科技公司...

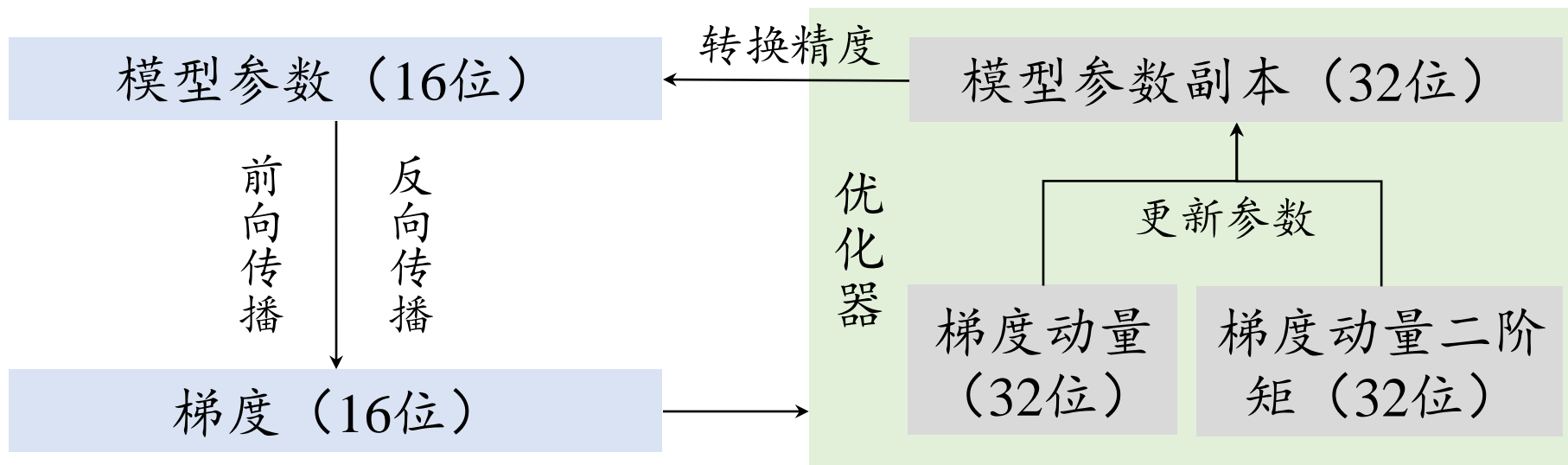
2023/6/20

# 训练显存估计

## ➤ 模型参数与优化器占用

➤ 模型参数量为  $P$ ，模型、梯度、优化器共计需要  $16P$  字节显存

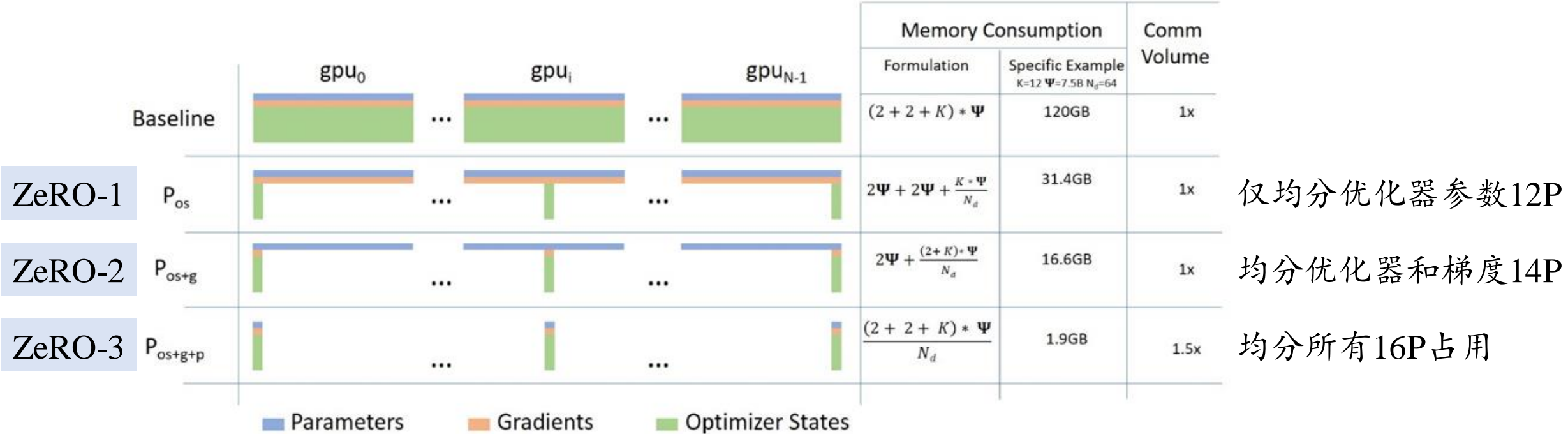
➤ 例如 7B 模型需要  $16 * 7B = 112GB$  显存占用



# 训练显存估计



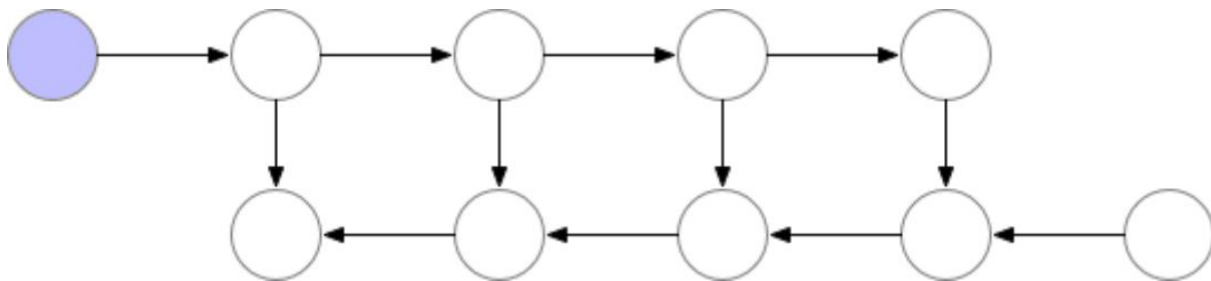
- 配合 ZeRO 技术将模型参数与优化器占用分配到每个 GPU
- N 张 GPU，使用 ZeRO-3 每张 GPU 显存占用  $\frac{16P}{N}$





# 激活值显存估计

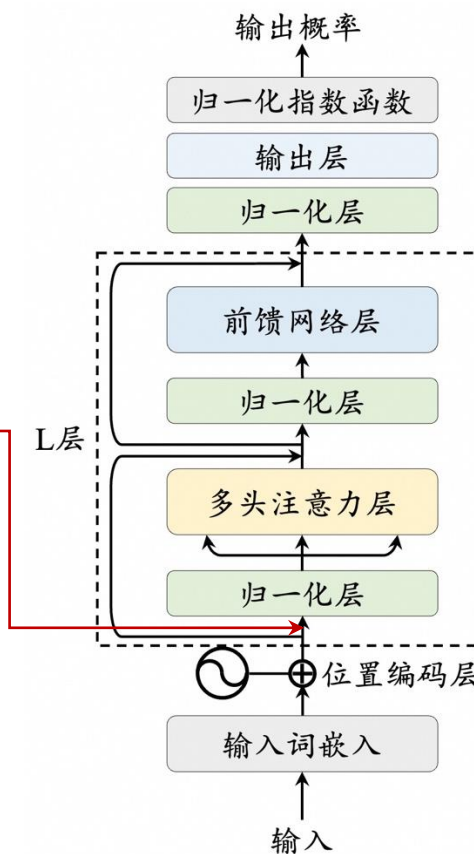
➤ 若保存所有激活值，7B 模型会产生 17GB 激活值占用（推导见教材）



➤ Transformer训练常保留每一层的输入作为激活值

➤ 批次大小为B，则需要  $2BTH$  字节保存每层输入

➤ 还需要  $4BTV$  字节保存输出层 softmax 输入



## ➤ 代码库内核

- PyTorch 框架占用约 1GB

## ➤ ZeRO 实现

- DeepSpeed 占用 1~4GB

## ➤ 中间结果

- 计算输出层 softmax 函数时，Transformers 代码库会额外引入 8BTV 字节显存占用

## ➤ 显存碎片

- 约 1GB

➤ 训练时每张GPU显存占用：

$$\underbrace{\frac{16P}{N}}_{\text{模型与优化器}} + \underbrace{2LBTH + 12BTV}_{\text{激活值}} + \underbrace{6}_{\text{其他}}$$

➤ 以 2 张 A100 80G 训练 LLaMA 7B 为例 (B=8)

- 模型与优化器：50GB
  - 激活值：6.2GB
  - 其他：10GB
- } 单卡总计 66GB



谢谢