



大模型 - AI 集群

大模型显存占用分析



ZOMI

关于本内容

关于本内容

- LLMs 大模型参数量、计算量、显存占用分析
 - Transformer 背景介绍
 - 大模型参数量计算方式
 - 大模型训练时间估计
 - 大模型 HBM (显存) 占用分析

1. Transformer

背景介绍

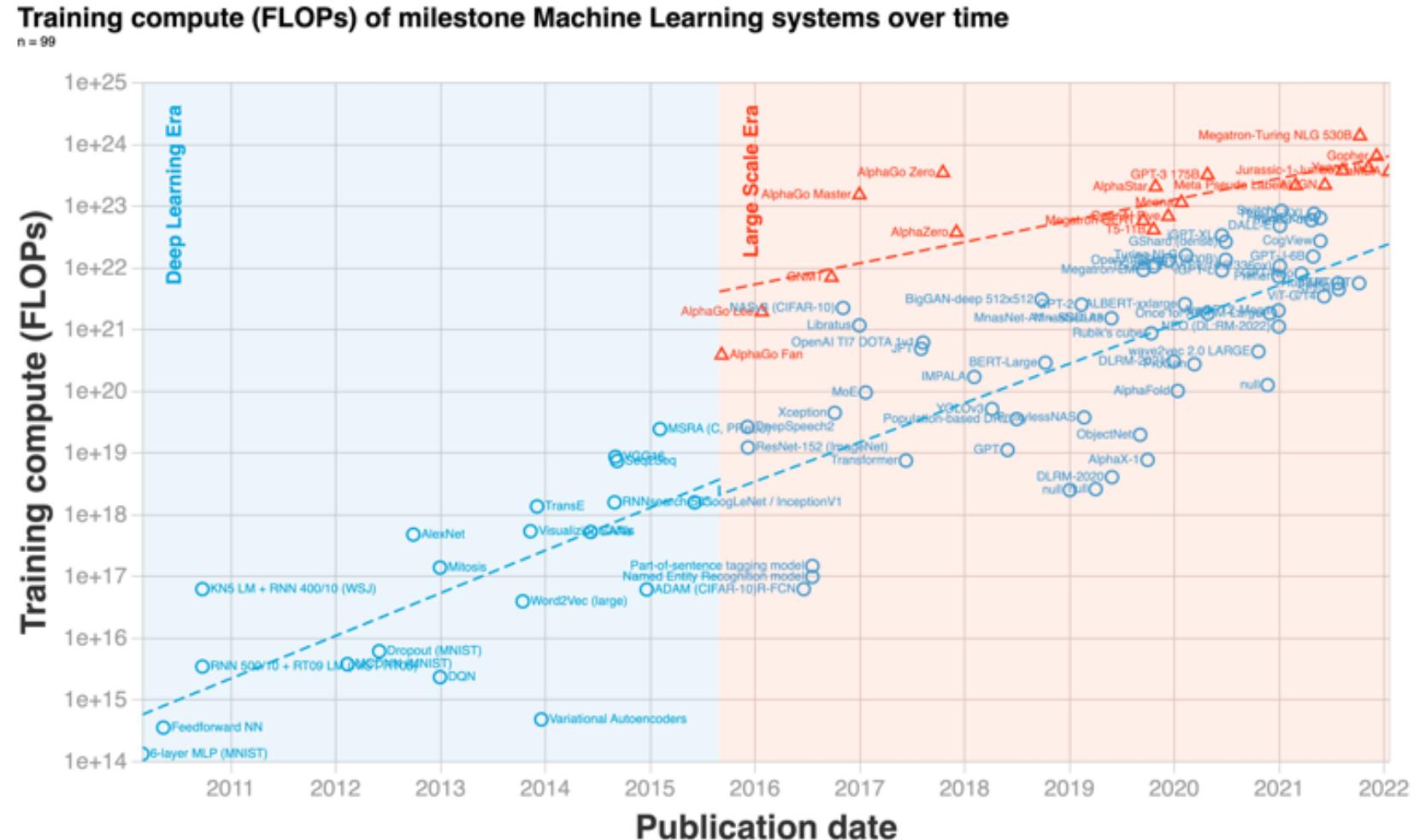
模型参数与数据集规模增长

- 千亿规模参数模型，数据集在TB量级；万亿规模参数模型，数据集在10TB量级。

模型	发布时间	参数		预训练数据集	模型类型
GPT-1	2018.06	1.17亿	0.1B	5GB	NLP
GPT-2	2019.02	15亿	1.5B	40GB	NLP
GPT-3	2020.05	1750亿	175B	3TB	NLP
GPT-4	2022.05	/	/	45TB	NLP
Switch Transformer	2021.03	1.6万亿	1.6T	750GB	CV
Megatron	2021.10	5300亿	530B	/	NLP
LLAMA	2022.10	650亿	65B	2TB	NLP
LLAMA2	2023.07	700亿	70B	4.5TB	NLP

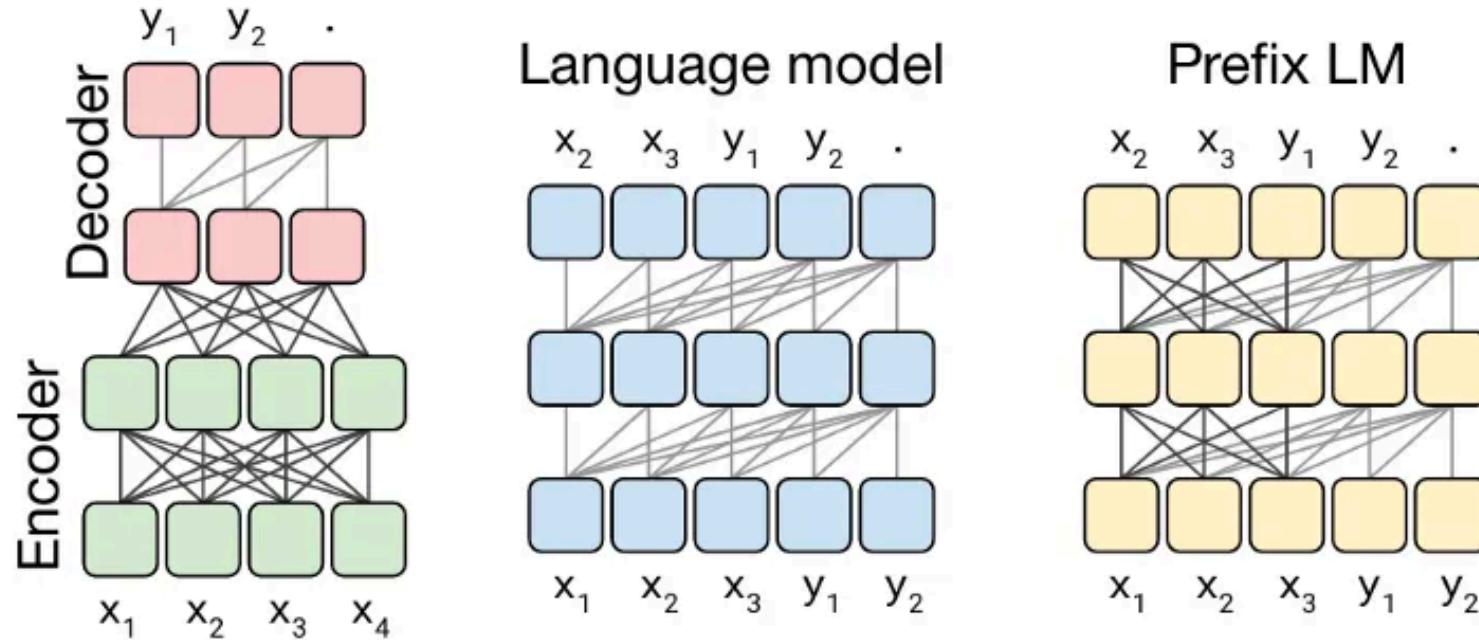
训练大模型面临的挑战

- 训练 LLMs 大模型面临两个主要挑战：
 - 显存效率
 - 计算效率



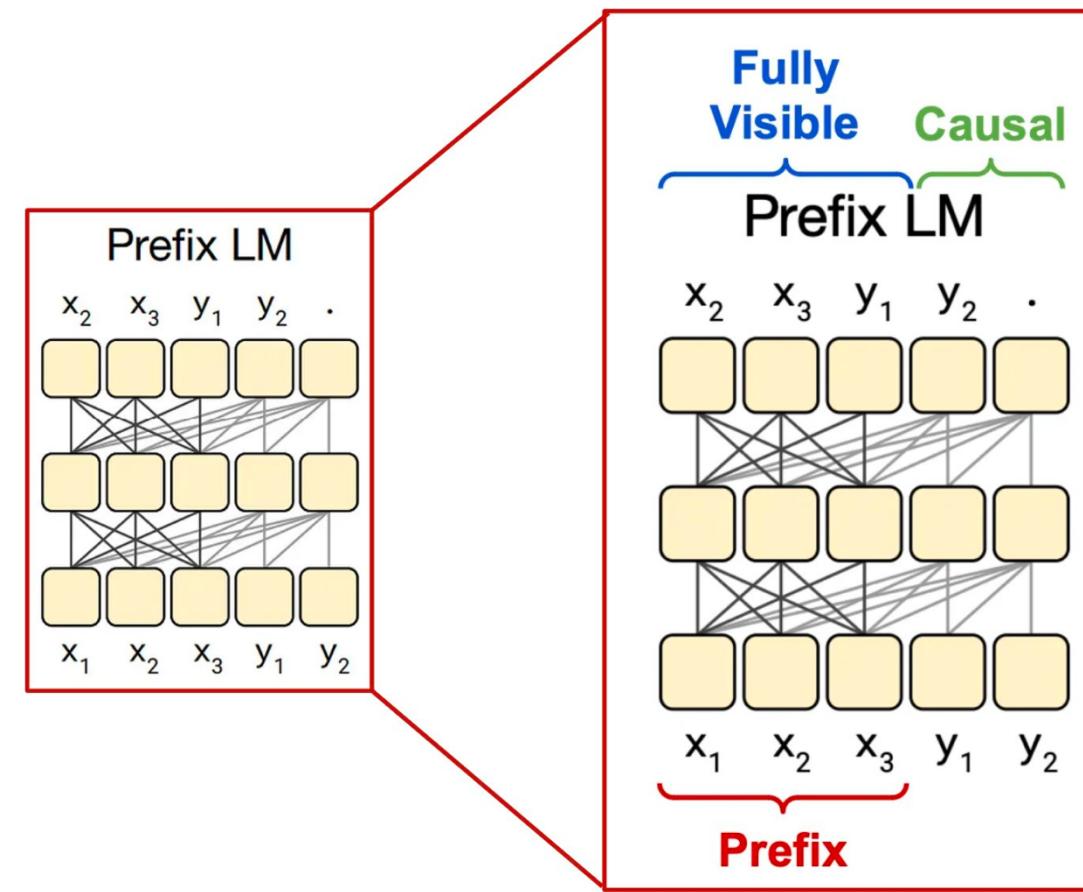
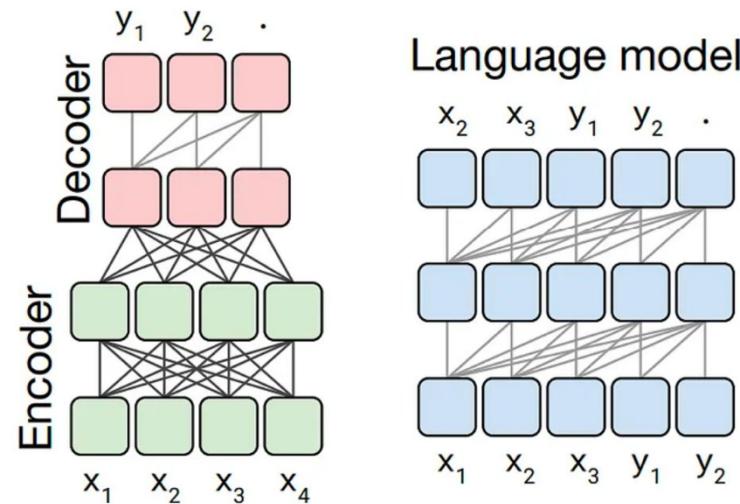
大模型结构

- 现在 LLMs 都基于 transformer 模型结构，主要有两大类：encoder-decoder 和 decoder-only。
- Decoder-only 又分为 Causal LM (e.g. GPT 系列) 和 Prefix LM (e.g. GLM) 。 GPT 系列取得巨大成功，目前主流 LLMs 大模型都采用 Causal LM 结构。



大模型结构

- 为了更好理解训练大模型的显存效率和计算效率，以 Decoder-only 结构来计算和评估大模型的参数量、计算量和显存占用情况。



符号

- Transformer 模型的层数为 l ，隐藏层维度为 h ，注意力头数为 a ，词表大小为 V ，训练数据的批次 batch 大小为 b ，序列长度为 s 。

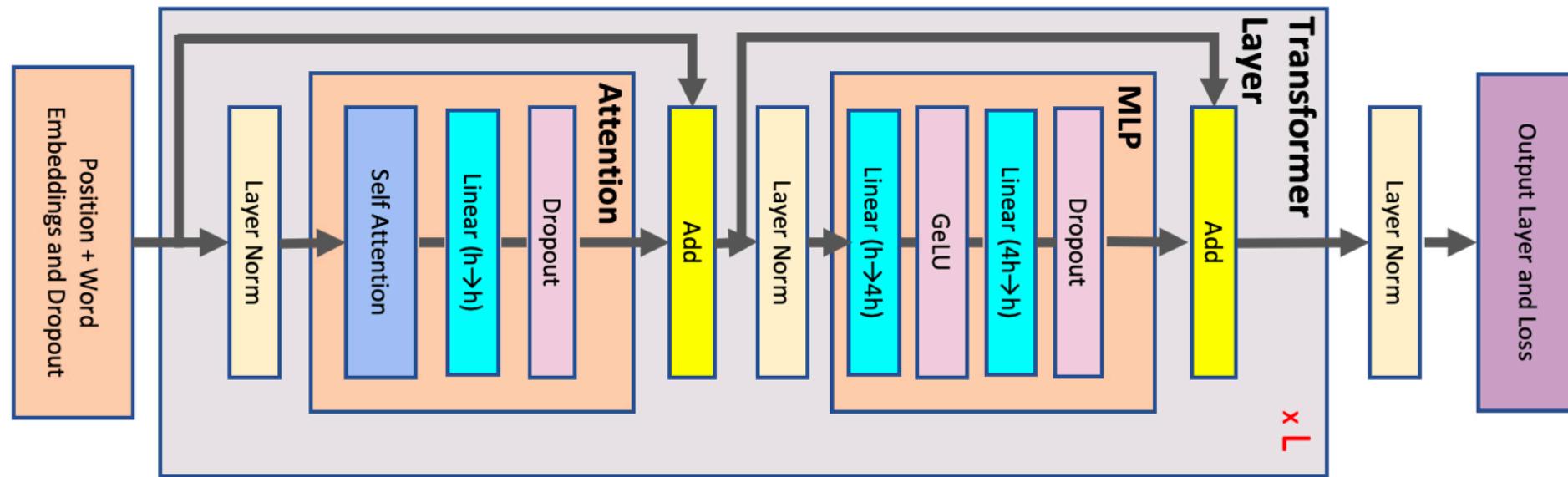


Figure 2: Transformer Architecture. Each gray block represents a single transformer layer that is replicated L times.

2. 大模型参数量

计算方式

Embedding 层

- 词嵌入矩阵 Embedding 层的参数量也较多，词向量维度 V 通常大于隐藏层维度 h ，词嵌入矩阵的参数量为 Vh 。

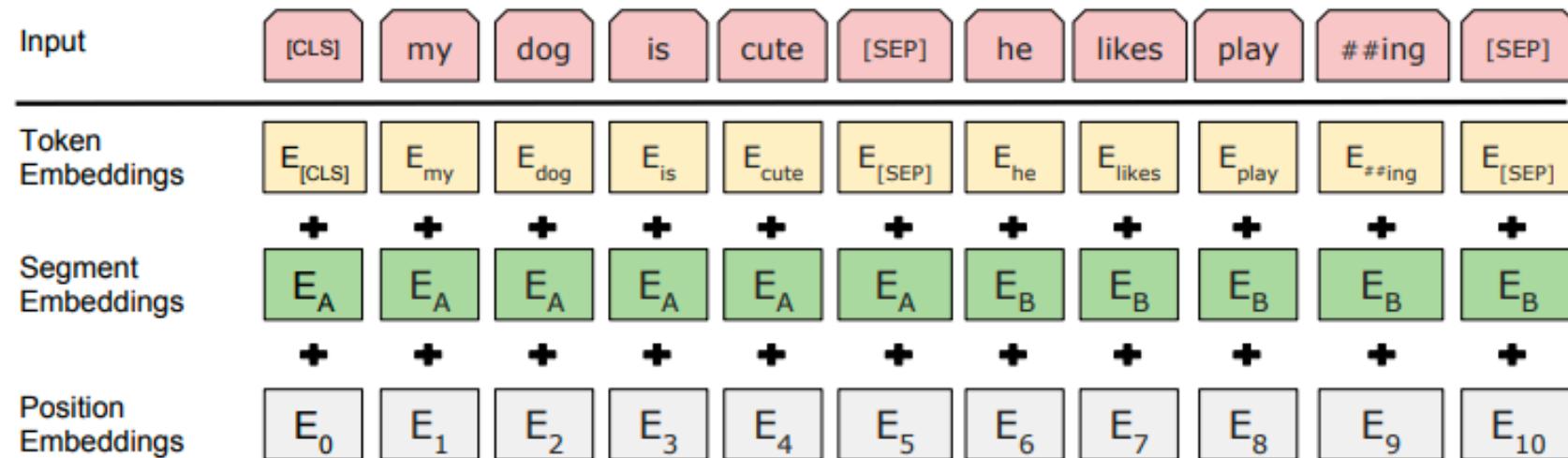


Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

Transformers 结构

- transformer 中最核心的为两个部分：Self-attention 层和 MLP 层

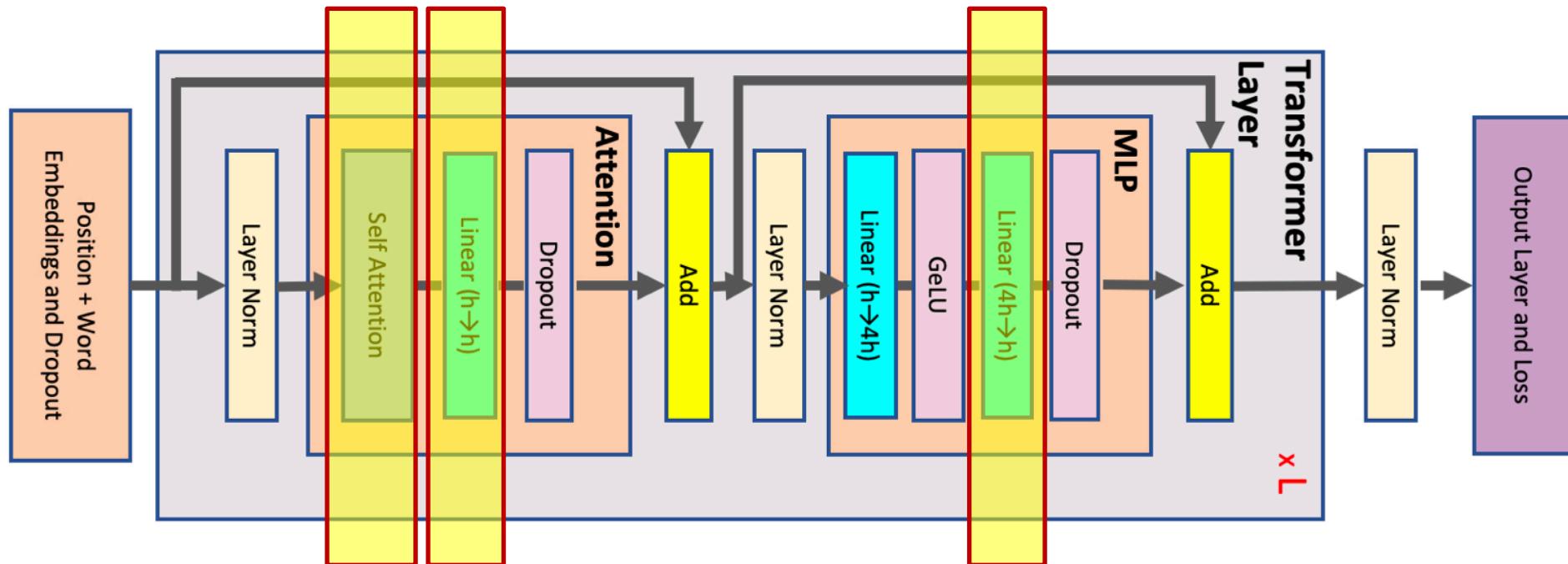
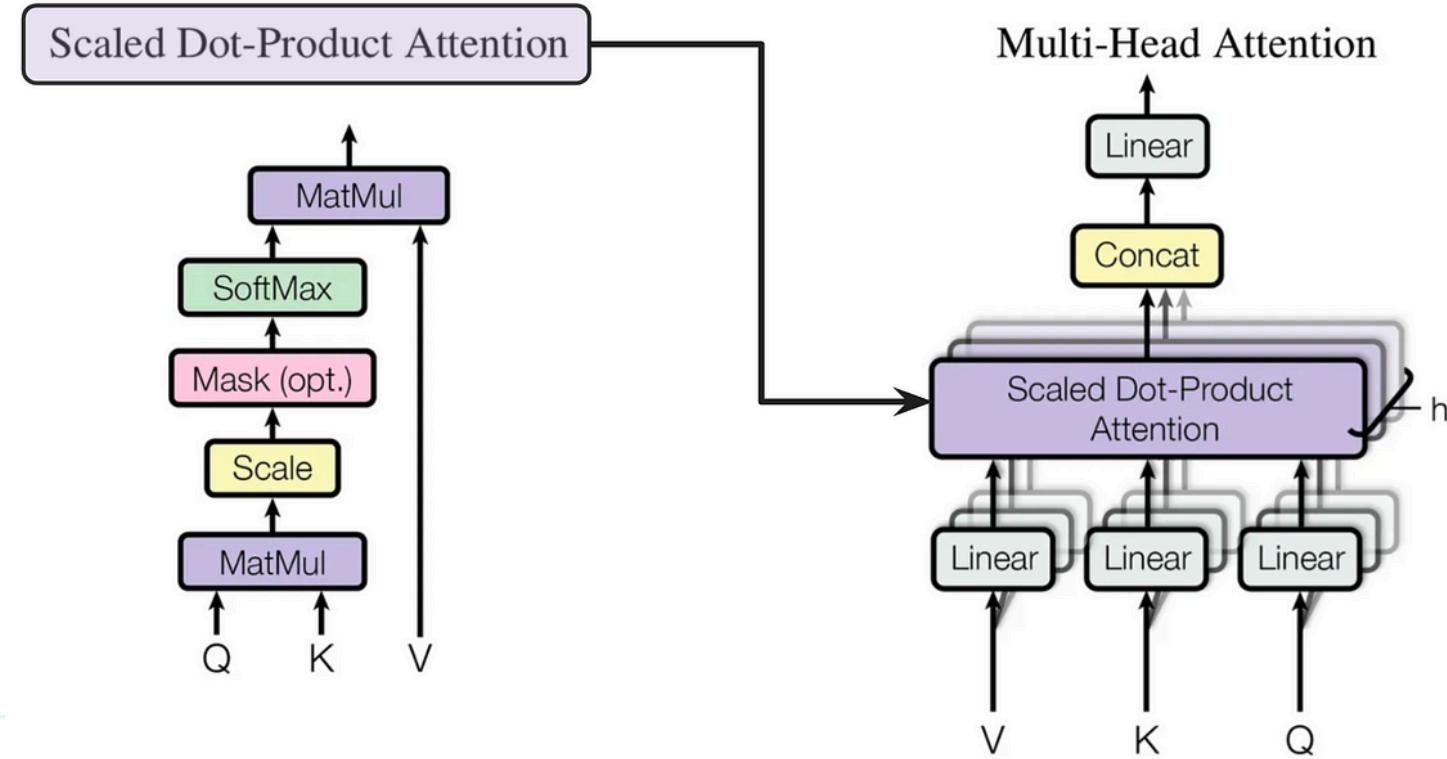


Figure 2: Transformer Architecture. Each gray block represents a single transformer layer that is replicated L times.

Self-attention 自注意力层

- Self-attention 模块参数包含 Q, K, V 权重矩阵 W_Q, W_K, W_V ，以及输出 W_O
- 4 个权重矩阵 Shape 为 $[h, h]$ ，因此 Self-attention 层自身的参数量为 $4 \times (ah^2 + Vha)$



MLP 全连接层

- MLP 由2个 Linear 线性层组成，第一个线性层将维度从 h 映射到 $4h$ ，权重 Shape 为 $[h, 4h]$ ；第二个线性层将维度从 $4h$ 映射回 h ，权重 Shape 为 $[4h, h]$ 。因此 MLP 层参数量为 $8h^2$ 。

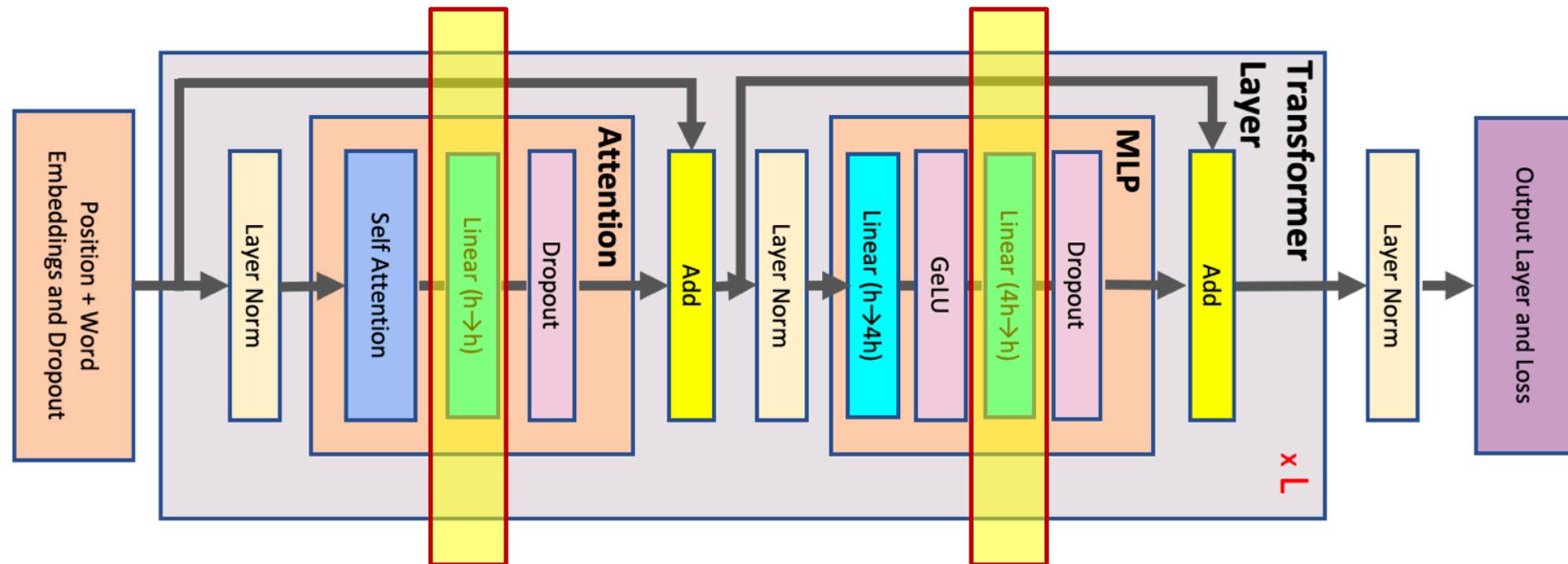


Figure 2: Transformer Architecture. Each gray block represents a single transformer layer that is replicated L times.

Layer Norm 归一化层

- Self-attention 层和 MLP 层各有 layer normalization，包含 2 个可训练参数：缩放参数 γ 和平移参数 β ，2 层 layer normalization 参数量为 $4h$ 。

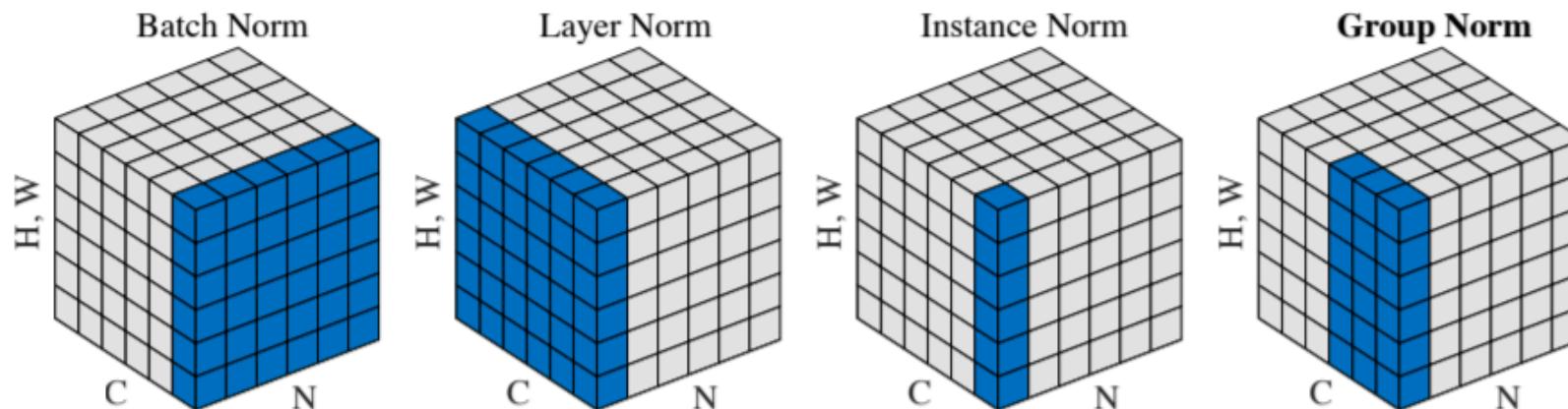


Figure 2. **Normalization methods.** Each subplot shows a feature map tensor, with N as the batch axis, C as the channel axis, and (H, W) as the spatial axes. The pixels in blue are normalized by the same mean and variance, computed by aggregating the values of these pixels.

模型参数量与模型大小关系

- l 层 Transformer 结构可训练参数量为 $l(12h^2) + Vh$
- 模型使用 FP16/BF16 方式保存和加载到显存，那么占用 2 个 Byte
- 模型使用 FP32 方式保存和加载到显存，那么占用 4 个 Byte

模型名称	隐藏层维度 h	层数 l	$12h^2$	实际参数量	模型大小 (FP16)
LLAMA-6B	4096	32	6442450944	6.7B	12 GB
LLAMA-13B	5120	40	12582912000	13.0B	23.4 GB
LLAMA-33B	6656	60	31897681920	32.5B	59.4 GB
LLAMA-65B	8192	80	64424509440	65.2B	120 GB

大模型训练内存占用

- l 层 Transformer 结构可训练参数量为 $l(12h^2) + Vh$

- **模型 Model**

- Parameters 权重参数 (half) 2 bytes ,
 - Gradient 梯度参数 (half) 2 bytes ,

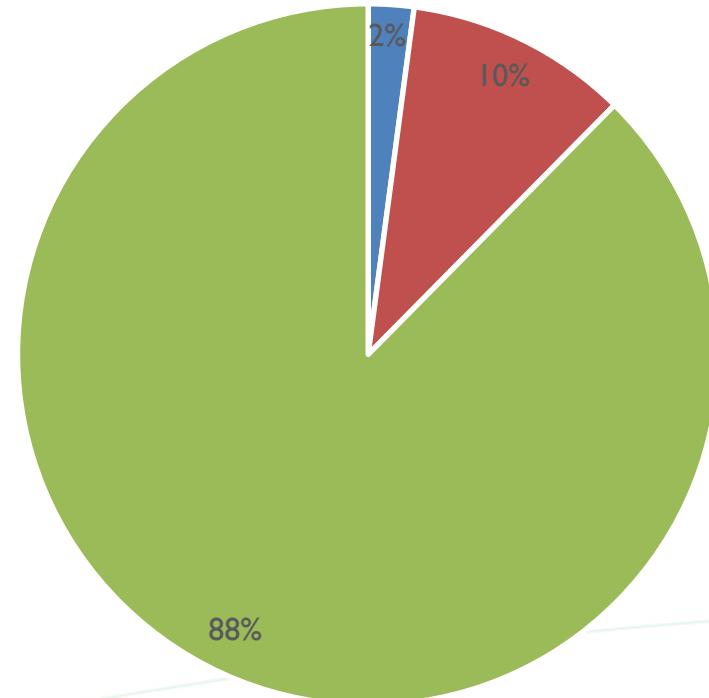
- **优化器状态 Optimizers status**

- Master Weight (FP32) 4 bytes
 - Adam m (FP32) 4 bytes
 - Adam v (FP32) 4 bytes

- **激活值：forward 中保存，用于反向传播**

BERT-Base(GB)

■ model ■ optimizer ■ activation



3. 大模型

训练时间估计

大模型训练时间计算公式

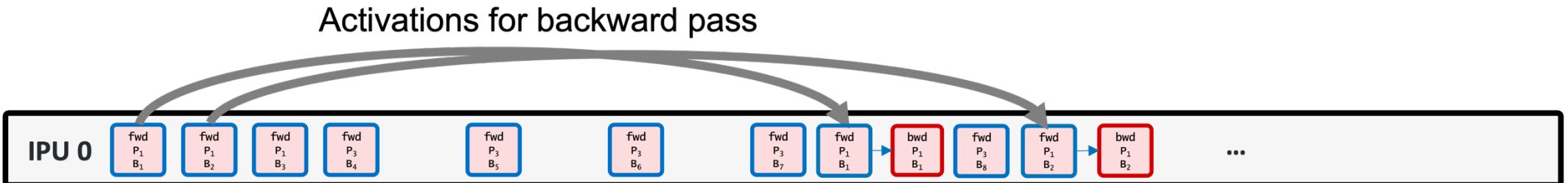
- 模型参数量（模型的大小）和训练总 tokens 数（训练的数据）决定 LLMs 模型需要的计算量。
- 给定计算量，训练时间不仅跟 NPU 类型和 NPU 集群数量有关系，还与 NPU 利用率相关。

$$E_t = \frac{K * T * P}{n * X}$$

- 其中， E_t 为端到端训练的理论时间（s）
- T 为训练数据的 Token 数量（B）
- P 为大模型的模型参数量（B）
- n 为 AI 集群的 NPU 卡数
- X 为每块卡的有效算力（标称 * 实际算力利用率）

系数 K 的来源

- LLMs 大模型中每个模型参数的计算主要是指点乘 ($C = A \cdot B$)，都需要将元素按位相乘，再按位相加，因此每个参数都需要进行 2 次浮点运算。
- 反向传播的计算量是前向传播的两倍，因此每个参数需要进行 4 次浮点运算。执行前向 + 反向 = $2 + 4 = 6$ 次浮点运算。
- 一般大模型训练为了节省内存，使用激活重计算来减少中间激活值，需要进行一次额外前向计算（Forward），因此对于每个 token，每个模型参数，需要进行 $2 + 4 + 2 = 8$ 次浮点数运算。



NPU 利用率

- 随着 AI 集群的 NPU 卡数越多，理论的算力利用率会更高。

Number of parameters (billion)	Attention heads	Hidden size	Number of layers	Tensor model-parallel size	Pipeline model-parallel size	Number of GPUs	Batch size	Achieved teraFLOP/s per GPU	Percentage of theoretical peak FLOP/s	Achieved aggregate petaFLOP/s
1.7	24	2304	24	1	1	32	512	137	44%	4.4
3.6	32	3072	30	2	1	64	512	138	44%	8.8
7.5	32	4096	36	4	1	128	512	142	46%	18.2
18.4	48	6144	40	8	1	256	1024	135	43%	34.6
39.1	64	8192	48	8	2	512	1536	138	44%	70.8
76.1	80	10240	60	8	4	1024	1792	140	45%	143.8
145.6	96	12288	80	8	8	1536	2304	148	47%	227.1
310.1	128	16384	96	8	16	1920	2160	155	50%	297.4
529.6	128	20480	105	8	35	2520	2520	163	52%	410.2
1008.0	160	25600	128	8	64	3072	3072	163	52%	502.0

Table 1: Weak-scaling throughput for GPT models ranging from 1 billion to 1 trillion parameters.

LLAMA-65B 训练时间

- 以 LLaMA-65B 为例，2048 张 80GB HBM A100，使用 1.4TB tokens 数据训练 65B 参数量模型。80GB 显存 A100 峰值性能为 312 TFLOPS，设 GPU 利用率为 0.6，需要训练 21 天。

$$E_t = \frac{8 \times (1.4 \times 10^{12}) \times (65 \times 10^9)}{2048 \times (312 \times 10^{12}) \times 0.6} \approx 1898871 \text{ s} \approx 21 \text{ days}$$

NVIDIA A100 Specs Table

	Peak Performance
FP64 Tensor Core	19.5 teraFLOPS
FP32	19.5 teraFLOPS
TF32 Tensor Core	156 teraFLOPS 312 teraFLOPS*
BFLOAT16 Tensor Core	312 teraFLOPS 624 teraFLOPS*
FP16 Tensor Core	312 teraFLOPS 624 teraFLOPS*
INT8 Tensor Core	624 TOPS 1,248 TOPS*
INT4 Tensor Core	1,248 TOPS 2,496 TOPS*
GPU Memory	40 GB

4. 大模型 HBM (显存) 占用分析

模型训练的总内存

- 在一次训练迭代中，每个可训练参数都对应 1 个梯度，2 个优化器状态（Adam）。设模型参数量为 $\varphi(FP16)$ ，那么梯度的参数量为 $2\varphi(FP32)$ ，Adam 优化器的参数量为 $4\varphi(FP32)$
- 大模型混合精度训练过程中，使用 BF16 进行前向传递，FP32 反向传递梯度信息；优化器更新模型参数时，使用 FP32 优化器状态、FP32 的梯度来更新模型参数。

$$\text{训练总内存} = \underbrace{\text{模型内存}}_{\varphi} + \underbrace{\text{梯度内存}}_{2\varphi} + \underbrace{\text{优化器内存}}_{4\varphi} + \underbrace{\text{激活内存}}_{None} + \underbrace{\text{其他内存}}_{1.X\varphi}$$

模型训练的总内存

$$\text{训练总内存} = \underbrace{\text{模型内存}}_{\varphi} + \underbrace{\text{梯度内存}}_{2\varphi} + \underbrace{\text{优化器内存}}_{4\varphi} + \underbrace{\text{激活内存}}_{None} + \underbrace{\text{其他内存}}_{1.X\varphi}$$

- 因此在 LLMs 大模型训练过程中最少需要 $8.X$ 倍于模型权重的大小内存，e.g. LLAMA-65B：
 - 65B 模型权重参数 130 GB，训练时候总消耗 NPU 内存最小为 $130GB \times 8 \approx 1TB$ 。
 - $1TG / 64 GB \approx 17$ ，除去激活值需要的内存，仅仅放下一个大模型就需要 32 张 NPU。

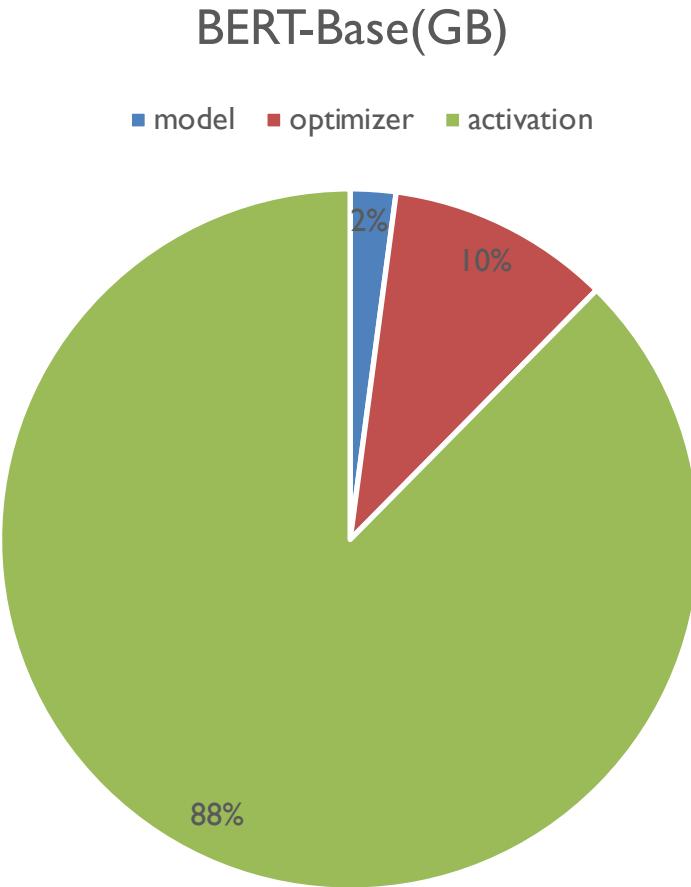
模型训练的总内存

- I. LLAMA-13B 模型权重为 25GB , 8倍为200GB , $200GB / 64GB \approx 3.2$, 理论上可以放在单机八卡的一个 NPU 节点。为什么 LLAMA-13B 一般最小资源需要两个节点 16卡 ?



大模型训练内存占用

- l 层 Transformer 结构可训练参数量为 $l(12h^2) + Vh$
- **模型 Model**
 - Parameters 权重参数 (half) 2 bytes ,
 - Gradient 梯度参数 (half) 2 bytes ,
- **优化器状态 Optimizers status**
 - Master Weight (FP32) 4 bytes
 - Adam m (FP32) 4 bytes
 - Adam v (FP32) 4 bytes
- **激活值：forward 中保存，用于反向传播**



激活值的数据（特征数据）占了大头

- 除了模型参数、梯度、优化器状态外，占用显存的大头是前向传播过程中计算得到的**中间激活值**，需要保存中间激活值以便在反向传播计算梯度时使用。

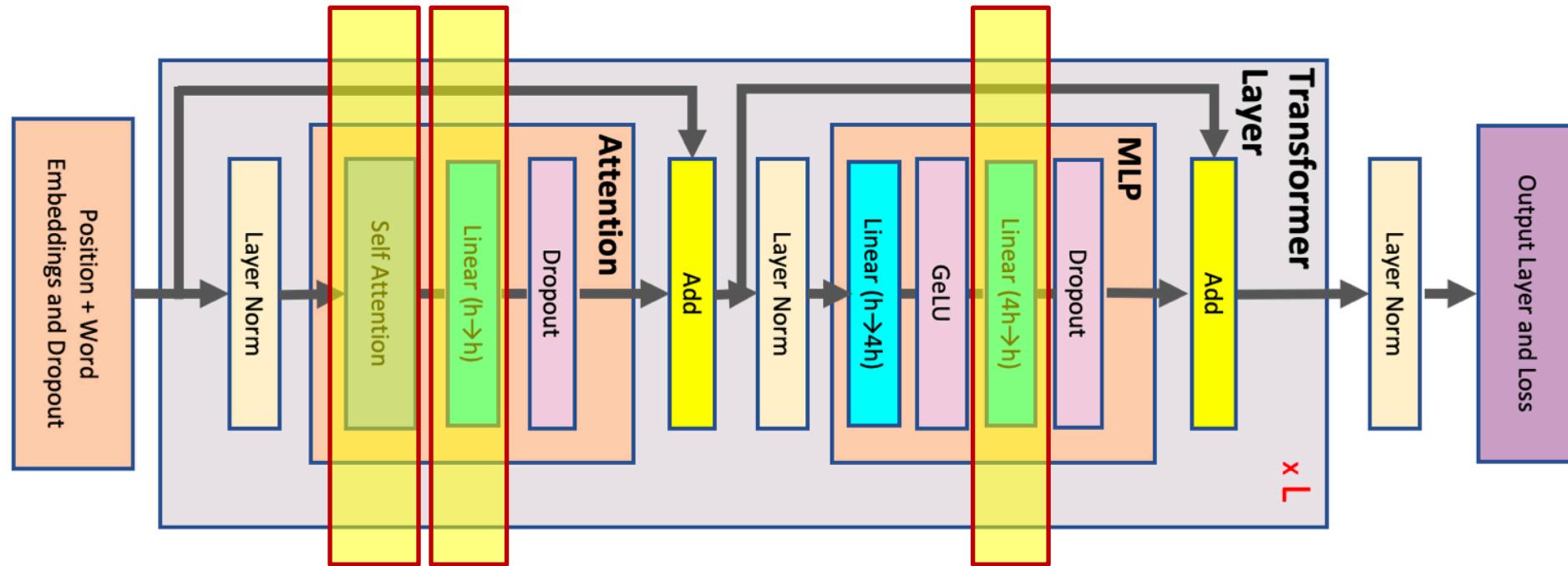


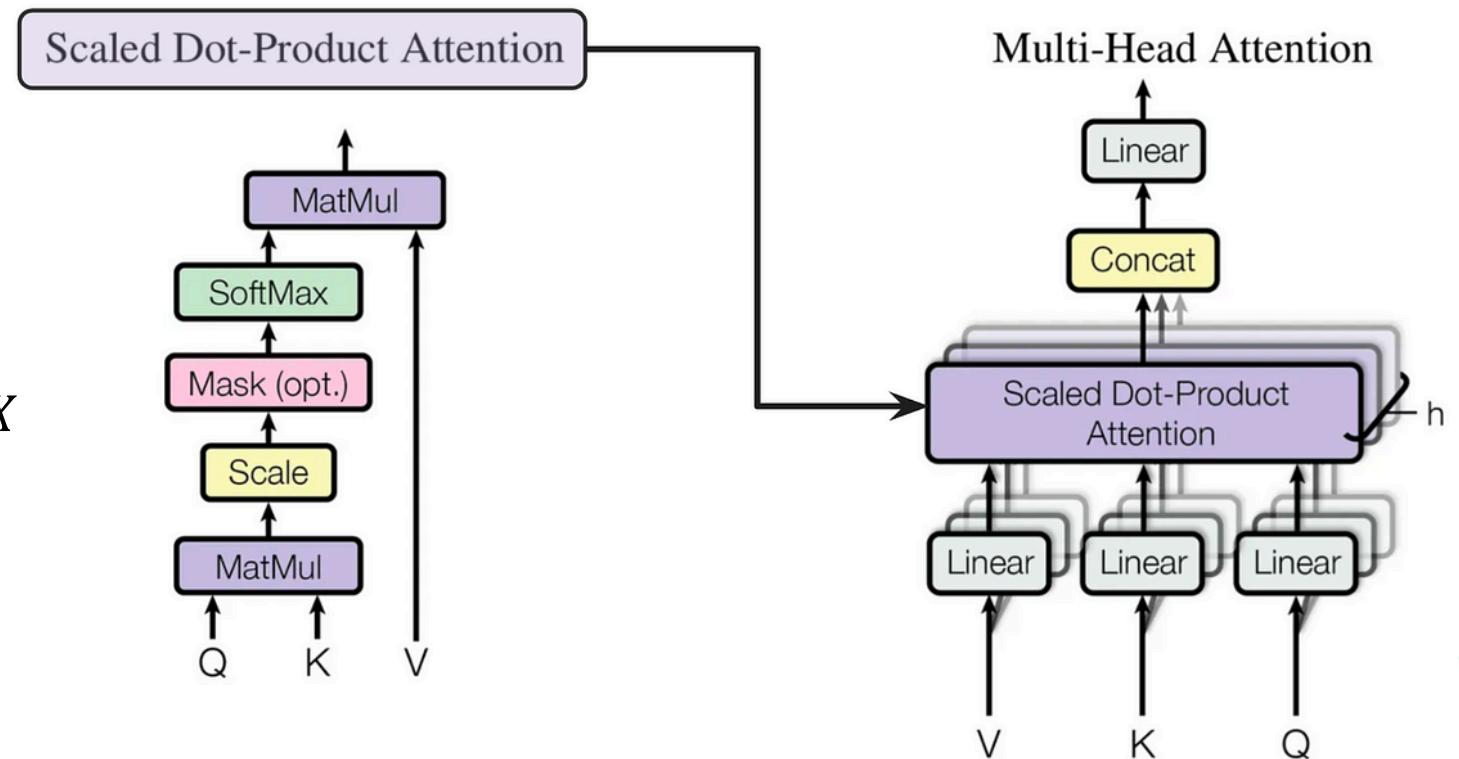
Figure 2: Transformer Architecture. Each gray block represents a single transformer layer that is replicated L times.

Self-attention 自注意力层

- Self-attention 模块参数包含 Q, K, V 权重矩阵 W_Q, W_K, W_V ，以及输出 W_O

$$Q = XW_Q, K = XW_K, V = XW_V$$

$$X_{out} = \text{soft}(QK^T / \sqrt{h}) \cdot V \cdot W_O + X$$



Self-attention 自注意力层

$$Q = XW_Q, K = XW_K, V = XW_V$$

$$X_{out} = \text{soft}(QK^T / \sqrt{h}) \cdot V \cdot W_0 + X$$

1. Q, K, V 中保存相同的输入 X , X 为激活值 , $X.\text{shape}$ 为 $[b, s, h]$, 占用显存大小为 $2 \times bsh$;
2. QK^T 中需要保存中间激活 Q, K , Shape 均为 $[b, s, h]$, 占用显存大小为 $2 \times 2 \times bsh = 4bsh$;
3. softmax 计算需要保存函数的输入 QK^T , 占用显存大小为 $2bs^2a$;
4. dropout 操作需要保存一个 mask 矩阵 , mask 矩阵的形状与 QK^2 相同 , 占用显存大小 bs^2a ;
5. 计算 $score \cdot V$, score 显存大小为 $2bs^2a$, V 显存大小 $2bsh$, 占用显存大小 $2bs^2a + 2bsh$;
6. 输出映射及 dropout 操作。输入映射大小为 $2bsh$, dropout 需要保存输入 mask , 显存为 bsh ;

Self-attention 自注意力层

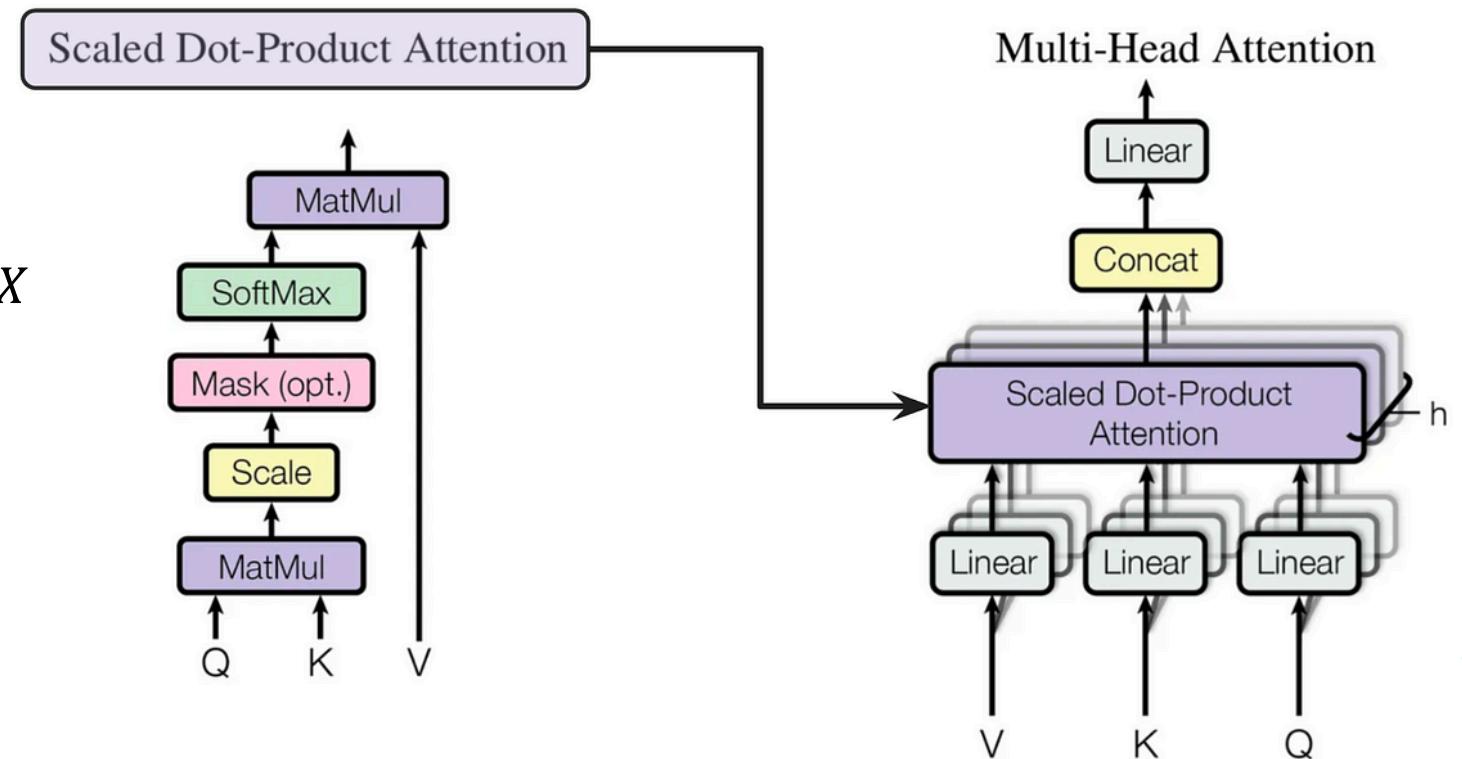
- Self-attention 模块参数包含 Q, K, V 权重矩阵 W_Q, W_K, W_V ，以及输出 W_0

$$Q = XW_Q, K = XW_K, V = XW_V$$

$$X_{out} = \text{soft}(QK^T/\sqrt{h}) \cdot V \cdot W_0 + X$$



$$5bs^2a + 11bsh$$



MLP 全连接层

- MLP 由 2 个 Linear 线性层组成，第一个线性层将维度从 h 映射到 $4h$ ，权重 Shape 为 $[h, 4h]$ ；第二个线性层将维度从 $4h$ 映射回 h ，权重 Shape 为 $[4h, h]$ 。因此 MLP 层参数量为 $8h^2$ 。

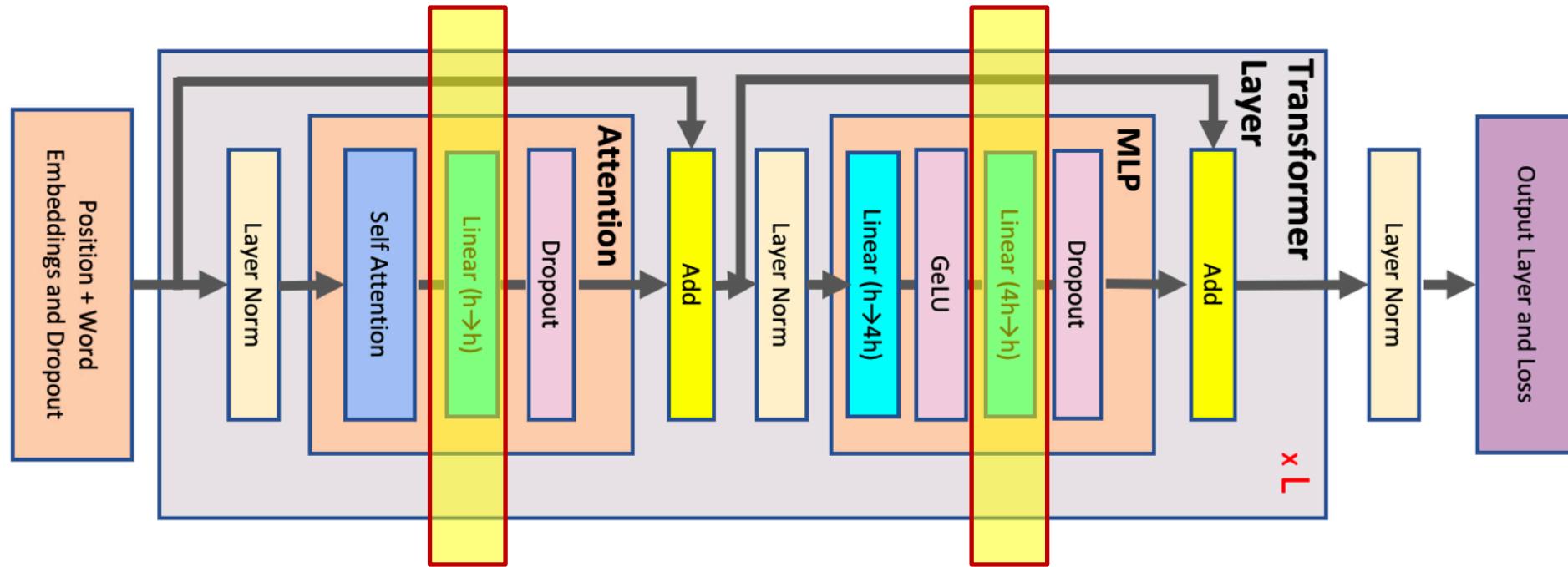


Figure 2: Transformer Architecture. Each gray block represents a single transformer layer that is replicated L times.

MLP 全连接层

$$X = f_{gelu}(X_{out}W_1)W_2 + X_{out}$$

- 第一个线性层需要保存其输入，占用显存大小为 $2bsh$ ；
- 激活函数需要保存其输入，占用显存大小为 $2 \times 4bsh = 8bsh$ ；
- 第二个线性层需要保存其输入，占用显存大小为 $8bsh$ ；
- 最后 dropout 操作，需要保存 mask 矩阵，占用显存大小为 bsh ；
- 因此，需要保存的中间激活值为 $19bsh$ ；

Layer Norm 归一化层

- Layer Norm 层需要保存其输入，大小为 $2bsh$ ，2 层 Layer Norm 需要保存的中间激活为 $4bsh$ 。

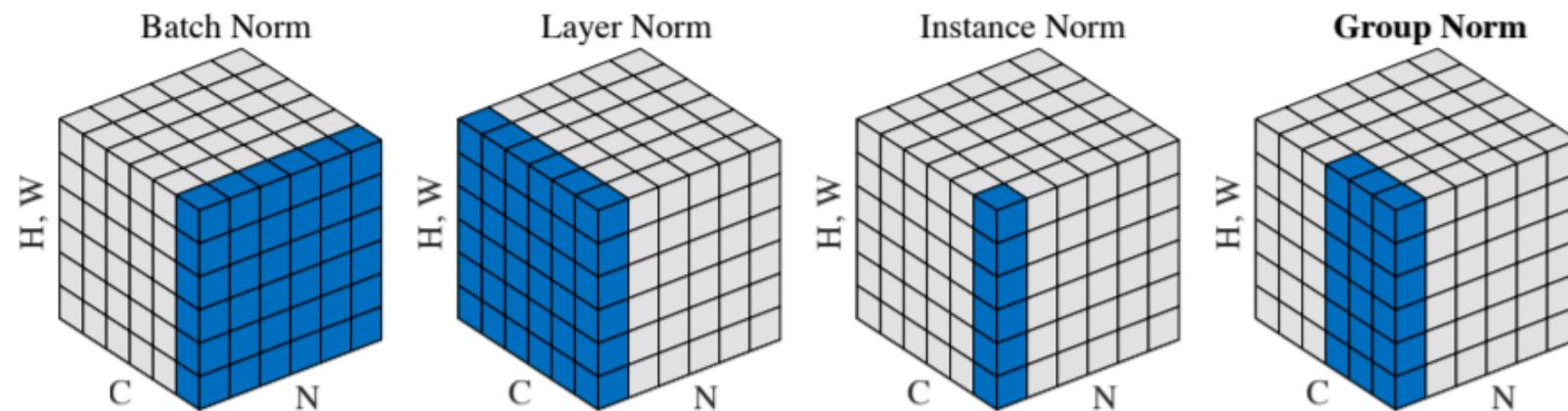


Figure 2. **Normalization methods.** Each subplot shows a feature map tensor, with N as the batch axis, C as the channel axis, and (H, W) as the spatial axes. The pixels in blue are normalized by the same mean and variance, computed by aggregating the values of these pixels.

Transformers 结构的激活值内存

$$Memory = l \times (5bs^2a + 34bsh)$$

- 激活值与输入数据的大小（输入数据的批次大小 b 和序列长度 s ）成正相关，随着批次大小 b 和序列长度 s 增大，激活占用显存会同步增大。

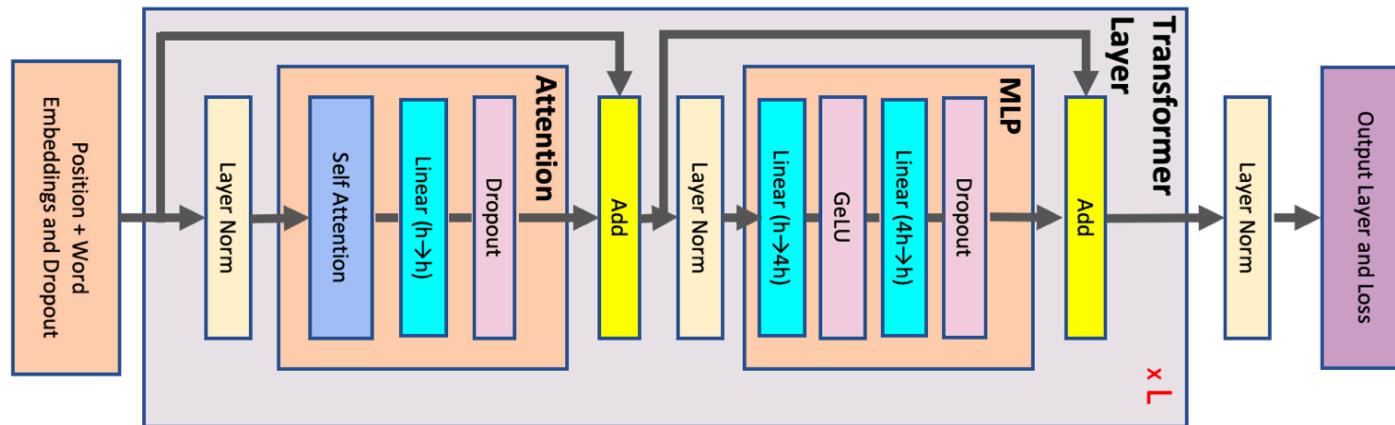


Figure 2: Transformer Architecture. Each gray block represents a single transformer layer that is replicated L times.

小结 & 思考



小结

1. 介绍了如何计算 Transformer 结构的参数量，及其与模型大小之间的计算关系。
2. 了解在给定训练 tokens 数据量下，大模型的训练时间计算方式。
3. 基于模型参数量可以进一步估计模型参数、梯度和优化器状态占用的显存大小。
4. 了解大模型的参数量、计算量、显存占用分析，有助于理解其训练和推理的显存和计算效率。



引用 & 参考

1. Efficient Large-Scale Language Model Training on GPU Clusters Using Megatron-LM
2. Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism
3. Reducing Activation Recomputation in Large Transformer Models
4. A Survey of Large Language Models
5. <https://medium.com/@amirhossein.abaskohi/text-to-text-models-in-natural-language-processing-f203afb34a8b>
6. [https://twitter.com/cwolferearch...](https://twitter.com/cwolferesearch/status/1649476518811148314)
7. <https://mdnice.com/writing/ce291e46450e415abd0c71f7282f3f20>
8. <https://medium.com/@ngiengkianyew/multi-headed-attention-8b940b76c351>
9. <https://stats.stackexchange.com/questions/620002/why-is-the-layer-normalization-same-with-the-instance-normalization-in-transform>
10. <https://www.kaggle.com/code/gabedossantos/killer-bert-tutorial>



Thank you

把AI系统带入每个开发者、每个家庭、
每个组织，构建万物互联的智能世界

Bring AI System to every person, home and
organization for a fully connected,
intelligent world.

Copyright © 2023 XXX Technologies Co., Ltd.
All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. XXX may change the information at any time without notice.



Course chenzomi12.github.io

GitHub github.com/chenzomi12/DeepLearningSystem