Analysis of Memory and Compute for AdamW and GPT-2 XL

Here is the breakdown of memory usage, computational cost, and training time based on the provided constraints (float32 precision, specific model architecture).

(a) Peak Memory Requirements

We assume all tensors are `float32` (4 bytes per element).

Definitions:

- B: Batch size
- L: Context length
- V: Vocabulary size
- N: Number of layers
- D: Model dimension (d_{model})
- H: Number of heads
- P: Total number of parameters

1. Parameters (Φ)

The parameters consist primarily of the embedding layers and the Transformer blocks.

- **Embeddings:** Token ($V \times D$) + Positional ($L \times D$)
- **Per Layer:**
  - Attention: $4 \times D \times D$ (Q, K, V, O projections)
  - Layer Norms: $2 \times 2 \times D$ (Scale and Bias for 2 LNs)
  - Feed-Forward: $2 \times D \times 4D = 8D^2$ (W_1 and W_2)
- **Final LN:** $2 \times D$
- **Output Head:** $D \times V$ (Assuming untied embeddings for generality, though often tied)

$$P \approx VD + LD + N(12D^2 + 4D) + 2D + VD$$

Approximation (Dominant terms): $P \approx 12ND^2 + 2VD$

Memory (Params): $4P$ bytes.

2. Optimizer State

AdamW maintains two moment vectors (m and v) for every parameter, each of the same shape as the parameter.

- State: $2 \times P$ elements.

Memory (Opt State): $8P$ bytes.

3. Gradients

We must store the gradient ($\nabla\theta$) for every parameter during backpropagation.

- Gradients: $1 \times P$ elements.

Memory (Gradients): $4P$ bytes.

4. Activations

We calculate the activations stored per layer for backpropagation based on the components listed.

- **Per Transformer Block:**
  - RMSNorm input: $B \cdot L \cdot D$
  - QKV Output: $3 \cdot B \cdot L \cdot D$
  - QK^T Matrix (Scores): $B \cdot H \cdot L \cdot L$
  - Softmax Output: $B \cdot H \cdot L \cdot L$
  - Weighted Sum (Context): $B \cdot L \cdot D$
  - Output Projection Input: $B \cdot L \cdot D$
  - FFN W_1 Input: $B \cdot L \cdot D$
  - SiLU Input/Output (internal width): $B \cdot L \cdot 4D$
  - FFN W_2 Input: $B \cdot L \cdot 4D$
  - **Layer Total:** $B \cdot L \cdot (11D + 8D) + 2 \cdot B \cdot H \cdot L^2 = B \cdot L \cdot 19D + 2BHL^2$ *(Note: Often intermediate FFN activations are grouped, but listed strictly: W1 mult, SiLU, W2 mult imply storing intermediates. We will simplify to the dominant terms: linear activations vs attention matrices).*
  - **Simplified Component Sum:** $B \cdot L \cdot 11D + 2 \cdot B \cdot H \cdot L^2$ (Strict interpretation of list items).
- **Non-Layer Components:**
  - Final RMSNorm: $B \cdot L \cdot D$
  - Output Embeddings/Logits: $B \cdot L \cdot V$

Memory (Activations): $4 \times B \times [N(11LD + 2HL^2) + LD + LV]$ bytes.

Total Peak Memory Formula

$$\text{Total} = \underbrace{16P}_{\text{Static Model + Opt}} + \underbrace{4B[N(11LD + 2HL^2) + LD + LV]}_{\text{Activations}} \text{ bytes}$$

(b) Instantiation for GPT-2 XL

Hyperparameters:

- $N = 48$
- $D = 1600$
- $H = 25$ (Note: $1600/25 = 64$ head dimension)
- $L = 1024$
- $V = 50257$

1. Static Memory (Fixed term b): First, calculate P (approximate):

$$P \approx 12(48)(1600)^2 + 2(50257)(1600) \approx 1.474 \times 10^9 + 0.16 \times 10^9 \approx 1.63 \text{ Billion Params}$$

Refining calculation: $P = 1.63 \times 10^9$ parameters. Static Memory ($16P$) = $1.63 \times 10^9 \times 16 \text{ bytes} \approx 26.08 \text{ GB}.$

2. Activation Memory per Batch (Coefficient a): Calculate elements per batch ($B = 1$):

- Layer Linear term: $48 \times 11 \times 1024 \times 1600 \approx 0.865 \times 10^9$
- Layer Attention term: $48 \times 2 \times 25 \times 1024^2 \approx 2.516 \times 10^9$
- Logits term: $1024 \times 50257 \approx 0.051 \times 10^9$
- Total elements $\approx 3.43 \times 10^9$ floats.
- Size in GB: $3.43 \times 4 \text{ bytes} \approx 13.72 \text{ GB}.$

Expression:

$$\text{Memory (GB)} \approx 13.72 \cdot \text{batch\_size} + 26.08$$

Maximum Batch Size: We need: $13.72B + 26.08 \leq 80$

$$13.72B \leq 53.92$$

$$B \leq 3.93$$

Result: The maximum batch size is **3**.

(Note: This low number highlights why techniques like Gradient Checkpointing, Mixed Precision (fp16/bf16), and ZeRO optimization are essential for training Large Language Models on GPUs).

(c) FLOPs per step of AdamW

For every parameter θ, AdamW performs the following element-wise operations:

1. **First Moment:** $m = \beta_1 m + (1 - \beta_1) g$
   - 3 FLOPs: 1 mul, 1 add, 1 mul (scalar broadcast).

2. **Second Moment:** $v = \beta_2 v + (1 - \beta_2) g^2$
   - 4 FLOPs: 1 square, 1 mul, 1 add, 1 mul.
3. **Update:** $\theta_{new} = \theta - \alpha \frac{m}{\sqrt{v+\epsilon}}$
   - 4 FLOPs: 1 sqrt, 1 add (ϵ), 1 div, 1 sub. (Absorbing α mult).
4. **Weight Decay:** $\theta = \theta - \alpha\lambda\theta$
   - 2 FLOPs: 1 mul, 1 sub.

Total: ≈ 13 FLOPs per parameter.

Expression: $13P$ FLOPs. *(Justification: Summing the element-wise floating point operations required to update state vectors and parameters).*

(d) Training Time Estimation

Parameters:

- Model Params $P \approx 1.63 \times 10^9$
- Hardware Peak: 19.5 TFLOPs (A100 FP32)
- MFU: 50%
- Effective Throughput: $9.75 \text{ TFLOPs} = 9.75 \times 10^{12} \text{ FLOPs/s}$
- Total Steps: $400,000$
- Batch Size: 1024
- Sequence Length: 1024

Compute per Step: Standard approximation for Transformer training (Kaplan/Hoffmann): Forward pass $\approx 2P$ FLOPs per token. Backward pass $\approx 4P$ FLOPs per token. Total $\approx 6P$ FLOPs per token.

Tokens per step $= B \times L = 1024 \times 1024 \approx 1.048 \times 10^6$. FLOPs per step $= 6 \times (1.63 \times 10^9) \times (1.048 \times 10^6) \approx 1.025 \times 10^{16} \text{ FLOPs}$.

Total Compute: Total FLOPs $= 400,000 \times 1.025 \times 10^{16} \approx 4.1 \times 10^{21} \text{ FLOPs}$.

Time:

$$\text{Time (seconds)} = \frac{4.1 \times 10^{21}}{9.75 \times 10^{12}} \approx 420,512 \text{ seconds}$$

$$\text{Time (days)} = \frac{420,512}{3600 \times 24} \approx 4.87 \text{ days}$$

Deliverable: Training would take approximately **4.9 days**. *(Justification: Total FLOPs calculated as $6 \times P \times \text{tokens}_{total}$ divided by the effective throughput of 9.75 TFLOPs/s).*