Tutorial Questions (1):
Performance Modeling and Roofline Analysis

Instructions

These tutorial questions are designed to be completed during the **first 30 ∼ 35 minutes** of the 1-hour tutorial session. In the remaining time, we will go through each question together as a group.

Please **do not use generative AI tools** to solve these questions. The goal is to build your own understanding and develop the independence required to prepare the final exam.

Question 1: Depthwise Convolution

Given a depthwise convolutional layer with:

- Input feature map: 32 channels, each of size 56×56
- Kernel size: 3×3
- Stride: 1 (same padding)
- Output size: same as input (56×56 pixels per channel)

(a) Compute the total number of parameters in the depthwise convolution layer. → $3 \times 3 \times 32$

(b) Compute the total number of FLOPs for one forward pass, where each multiply–accumulate (MAC) operation counts as two FLOPs.

→ ~~54x54 positions~~, 3x3 kernel, 32 channels → $2 \times 56 \times 56 \times 3 \times 3 \times 32 \doteq 1{,}8$ MFLOPs

56x56 positions thanks to the padding

Question 2: Scaled Dot-Product Attention

Given a scaled dot-product attention layer with:

- Single-head attention
- Query, Key, and Value input vectors: 128-dimensional
- Sequence length: 64 → 64×128
- Weight matrices:
  - $W_Q, W_K, W_V \in \mathbb{R}^{128\times128}$
  - $W_O \in \mathbb{R}^{128\times128}$

(a) Compute the total number of weight parameters. → $4 \times 128 \times 128$

$$(L \times d) \cdot (d \times d) = (L \times d)$$

attention score between each position

(b) Compute the number of FLOPs for one forward pass of the following operations.

- Q/K/V projections. per projection: $2 \times 64 \times 128^2$
- Attention score computation ($QK^\top$). $\rightarrow 2 \times (128 \times 64^2)$
- Scaling by $1/\sqrt{d}$. $\rightarrow 64^2$
- Value aggregation (softmax$(S)V$). $2 \times 64^2 \times 128$
- Output projection. $\rightarrow 2 \times 64 \times 128^2$

$Q \in \mathbb{R}^{64 \times 128}$

$W^\top \in \mathbb{R}^{128 \times 64}$

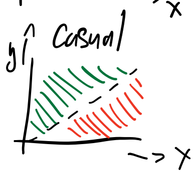
$(QW^\top) \in \mathbb{R}^{64 \times 64}$

(c) Analyze the softmax and specify the operation types and their counts.

exp.: 64^2, arith.: $64 \times (3 \times 64)$ $\rightarrow$ substr., addition, divisions

$$\text{softmax}(\vec{x})_i = \frac{e^{x_i}}{\sum_j e^{x_j}} \approx O(n) \qquad \vec{x} \in \mathbb{R}^n$$

(d) Compare the computational cost of **full attention masking** vs. **causal masking**:



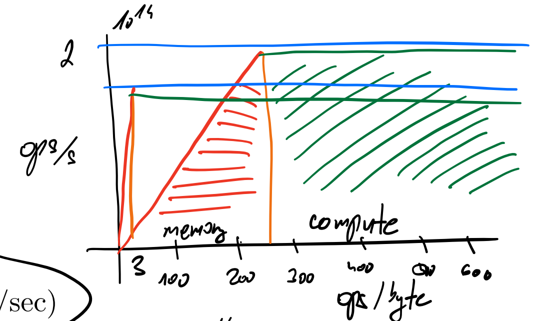
- How many dot-product scores are computed in each case?
- How does the time complexity scale with sequence length L in both cases?

full: d^2, causal: $\frac{64 \cdot 65}{2} \approx d^2/2$

Question 3: Roofline Model for GPU

Given the following hardware setup of a GPU:

- Memory Bandwidth = **800 GB/s** $= 8 \cdot 10^{11}$
- Peak Compute Performance = **200 TFLOPs** (2×10^{14} FLOPs/sec)



(a) Calculate the roofline **turning point** in FLOPs/byte.

$$\frac{2 \cdot 10^{14}}{8 \cdot 10^{11}} = \frac{1}{4} \cdot 10^3 = 2{,}5 \cdot 10^2$$

(b) Sketch the Roofline model showing the memory-bound and compute-bound regions.

$\rightarrow$ when $\underset{B/s}{B} \cdot \underset{ops/s}{TP} = \underset{ops/s}{PCP}$ $\quad TP = \frac{PCP}{B}$ ops/s

Question 4: Roofline Model for Processing-in-Memory (PIM) Processor

Given the hardware setup of a Processing-in-Memory (PIM) accelerator:

- Memory Bandwidth = **6000 GB/s** $= 6 \cdot 10^{12}$
- Peak Compute Performance = **20 TFLOPs** (2×10^{13} FLOPs/sec) $= 2 \cdot 10^{13}$

(a) Calculate the roofline **turning point** in FLOPs/byte.

(b) Sketch the Roofline model using the same axes as in Question 3 for direct comparison.

$$\rightarrow \frac{2 \cdot 10^{13}}{6 \cdot 10^{12}} = \frac{1}{3} \cdot 10 \doteq 3{,}33$$

$\rightarrow$ since it is making less computations and has higher memory bandwidth, we come way earlier to compute-bound issues.

Question 5: Operation Placement on Roofline Models

Assume all activations and weights use 32-bit floating point (4 bytes/element):

(a) Using the operations from **Question 1** (depthwise conv), estimate their **operational/arithmetic intensity** (FLOPs per byte accessed).

(b) Using the operations from **Question 2** (only consider Q/K/V projections), estimate their **operational/arithmetic intensity**

(c) Plot these operation on the two Roofline models (from Questions 3 and 4).

(d) Determine whether each operation is **memory-bound** or **compute-bound** under each hardware setup.

(e) Briefly explain why, based on the arithmetic intensity vs turning point.

check with published results

Question 6: Advantages of Transformer over RNN

- List at least **two key advantages** of Transformer/Attention-based architectures over Recurrent Neural Networks (RNNs).
- Briefly explain each advantage in 1–2 sentences.

1) attention mechanism enables to easilly attend to the begin and end of the seq. at the same step, making it easier to capture overall semantic meaning.

2) RNNs are required to be inferenced sequentially, which is making them slower for inference. Also the problem with dissolving gradient is way more present.

These two were mentioned on the lecture:

- ➢Sequential processing → slow
- ➢Long-range dependencies degrade

5a) 1,8 MFLOPs per one pass

56×56 × 32 one image → 56 × 56 × 32 × 4 = 401 408 B

loading + storing : 2x

kernel: 3×3× 32 = 288

Total operation = 803 104 B

$$\frac{1{,}8 \cdot 10^6}{8 \cdot 10^5} = 2{,}25 \text{ ops/byte}$$

5b) 2 MFLOPs , 2 MB

$$\frac{2 \cdot 10^6}{2 \cdot 10^6} = 1 \text{ ops/byte}$$

– the matrix multiplication is way more memory bound, whereas convolution is more compute-bound, even though in both models is in the memory-bound.