PDC Assignment 04
NanoGpt

Warm-Up

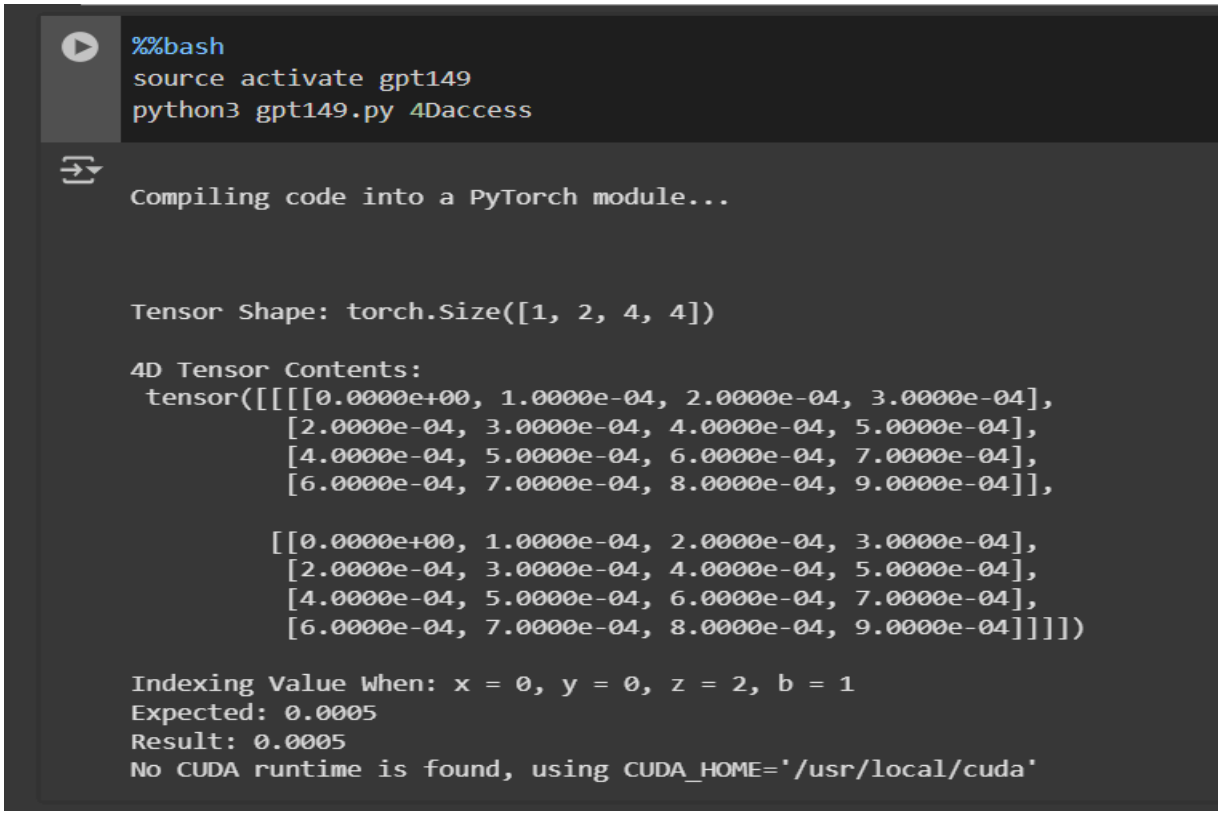
Q: Briefly describe how a 4D tensor/array is laid out in memory. Why do you think this convention was chosen and how does it leverage hardware?

A 4D tensor in C++ (flattened into a 1D vector) is stored in row-major order.

For a tensor of shape `[B][H][N][d]`, the flattened index is:

index = b * (H * N * d) + h * (N * d) + n * d + d

This layout maximizes spatial locality: elements along the innermost dimension are stored contiguously, which improves cache efficiency, reduces memory latency, and enables SIMD vectorization. It's optimized for the hardware's memory access patterns, especially during tight inner loops such as dot products in attention.



```
%%bash
source activate gpt149
python3 gpt149.py 4Daccess
```

```
Compiling code into a PyTorch module...



Tensor Shape: torch.Size([1, 2, 4, 4])

4D Tensor Contents:
 tensor([[[[0.0000e+00, 1.0000e-04, 2.0000e-04, 3.0000e-04],
           [2.0000e-04, 3.0000e-04, 4.0000e-04, 5.0000e-04],
           [4.0000e-04, 5.0000e-04, 6.0000e-04, 7.0000e-04],
           [6.0000e-04, 7.0000e-04, 8.0000e-04, 9.0000e-04]],

          [[0.0000e+00, 1.0000e-04, 2.0000e-04, 3.0000e-04],
           [2.0000e-04, 3.0000e-04, 4.0000e-04, 5.0000e-04],
           [4.0000e-04, 5.0000e-04, 6.0000e-04, 7.0000e-04],
           [6.0000e-04, 7.0000e-04, 8.0000e-04, 9.0000e-04]]]])

Indexing Value When: x = 0, y = 0, z = 2, b = 1
Expected: 0.0005
Result: 0.0005
No CUDA runtime is found, using CUDA_HOME='/usr/local/cuda'
```

Part 1: Naive Attention

python3 gpt149.py part1

Output :

Running Part 1 Test: Naive Unfused Attention

-----RUNNING REFERENCE IMPLEMENTATION-----

manual attention == pytorch attention True

Manual Execution Time: 0.35216212272644043

Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	CPU Mem	Self CPU Mem	# of Calls
aten::empty	0.02%	66.000us	0.02%	66.000us	22.000us	5.00 Mb	5.00 Mb	3
REFERENCE - NAIVE ATTENTION	99.06%	348.907ms	99.97%	352.121ms	352.121ms	4.50 Mb	-1.00 Mb	1
aten::zeros	0.02%	58.000us	0.60%	2.115ms	1.058ms	4.50 Mb	0 b	2
aten::clone	0.02%	73.000us	0.28%	995.000us	497.500us	1.00 Mb	0 b	2
model_inference	0.03%	98.000us	100.00%	352.219ms	352.219ms	512.00 Kb	-4.00 Mb	1
aten::flatten	0.02%	74.000us	0.18%	648.000us	129.600us	512.00 Kb	0 b	5
aten::empty_like	0.00%	10.000us	0.01%	26.000us	26.000us	512.00 Kb	0 b	1
aten::empty_strided	0.01%	43.000us	0.01%	43.000us	43.000us	512.00 Kb	512.00 Kb	1
aten::zero_	0.01%	49.000us	0.57%	2.007ms	1.004ms	0 b	0 b	2
aten::fill_	0.56%	1.958ms	0.56%	1.958ms	979.000us	0 b	0 b	2

Self CPU time total: 352.219ms

REFERENCE - NAIVE ATTENTION STATISTICS

cpu time: 352.121ms

mem usage: 4718592 bytes

-----RUNNING STUDENT IMPLEMENTATION-----

manual attention == pytorch attention True

Manual Execution Time: 0.31907129287719727

Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	CPU Mem	Self CPU Mem	# of Calls
aten::empty	0.01%	46.000us	0.01%	46.000us	15.333us	5.00 Mb	5.00 Mb	3
STUDENT - NAIVE ATTENTION	99.30%	316.913ms	99.95%	319.015ms	319.015ms	4.50 Mb	-1.00 Mb	1
aten::zeros	0.01%	44.000us	0.38%	1.226ms	613.000us	4.50 Mb	0 b	2
aten::clone	0.02%	50.000us	0.25%	793.000us	396.500us	1.00 Mb	0 b	2
model_inference	0.05%	144.000us	100.00%	319.159ms	319.159ms	512.00 Kb	-4.00 Mb	1
aten::flatten	0.02%	49.000us	0.19%	606.000us	121.200us	512.00 Kb	0 b	5
aten::empty_like	0.00%	9.000us	0.01%	19.000us	19.000us	512.00 Kb	0 b	1
aten::empty_strided	0.00%	15.000us	0.00%	15.000us	15.000us	512.00 Kb	512.00 Kb	1
aten::zero_	0.01%	35.000us	0.36%	1.146ms	573.000us	0 b	0 b	2
aten::fill_	0.35%	1.111ms	0.35%	1.111ms	555.500us	0 b	0 b	2

Self CPU time total: 319.159ms

STUDENT - NAIVE ATTENTION statistics

cpu time: 319.015ms

mem usage: 4718592 bytes

python3 gpt149.py part1 -N 64

Output :

Running Part 1 Test: Naive Unfused Attention

-----RUNNING REFERENCE IMPLEMENTATION-----

manual attention == pytorch attention True

Manual Execution Time: 0.002266407012939453

Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	CPU Mem	Self CPU Mem	# of Calls
aten::empty	1.08%	25.000us	1.08%	25.000us	8.333us	80.00 Kb	80.00 Kb	3
aten::clone	1.59%	37.000us	4.31%	100.000us	50.000us	64.00 Kb	0 b	2
REFERENCE - NAIVE ATTENTION	84.53%	1.961ms	95.86%	2.224ms	2.224ms	48.00 Kb	-64.00 Kb	1
aten::zeros	1.85%	43.000us	4.35%	101.000us	50.500us	48.00 Kb	0 b	2
model_inference	4.14%	96.000us	100.00%	2.320ms	2.320ms	32.00 Kb	-16.00 Kb	1
aten::flatten	1.94%	45.000us	5.39%	125.000us	25.000us	32.00 Kb	0 b	5
aten::empty_like	0.26%	6.000us	0.39%	9.000us	9.000us	32.00 Kb	0 b	1
aten::empty_strided	0.34%	8.000us	0.34%	8.000us	8.000us	32.00 Kb	32.00 Kb	1
aten::zero_	1.55%	36.000us	1.55%	36.000us	18.000us	0 b	0 b	2
aten::view	0.60%	14.000us	0.60%	14.000us	3.500us	0 b	0 b	4

Self CPU time total: 2.320ms

REFERENCE - NAIVE ATTENTION statistics

cpu time: 2.224ms

mem usage: 49152 bytes

-----RUNNING STUDENT IMPLEMENTATION-----

manual attention == pytorch attention True

Manual Execution Time: 0.001355886459350586

Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	CPU Mem	Self CPU Mem	# of Calls
aten::clone	1.22%	17.000us	4.36%	61.000us	30.500us	64.00 Kb	0 b	2
STUDENT - NAIVE ATTENTION	84.92%	1.188ms	94.50%	1.322ms	1.322ms	48.00 Kb	-64.00 Kb	1
aten::zeros	1.64%	23.000us	3.22%	45.000us	22.500us	48.00 Kb	0 b	2
aten::empty	0.93%	13.000us	0.93%	13.000us	4.333us	48.00 Kb	48.00 Kb	3
model_inference	5.50%	77.000us	100.00%	1.399ms	1.399ms	32.00 Kb	-16.00 Kb	1

aten::flatten 1.43% 20.000us 4.93% 69.000us 13.800us 32.00 Kb 0 b 5

aten::empty_like 0.29% 4.000us 0.36% 5.000us 5.000us 32.00 Kb 32.00 Kb 1

aten::empty_strided 0.36% 5.000us 0.36% 5.000us 5.000us 32.00 Kb 32.00 Kb 1

aten::zero_ 0.71% 10.000us 0.71% 10.000us 5.000us 0 b 0 b 2

aten::view 0.50% 7.000us 0.50% 7.000us 1.750us 0 b 0 b 4

Self CPU time total: 1.399ms

STUDENT - NAIVE ATTENTION statistics

cpu time: 1.322ms

mem usage: 49152 bytes

Part 2: Blocked Matrix Multiply + Unfused Softmax

Q1: Tile Size

With N = 1024:

- 16 → 218 ms
- 32 → 176 ms
- 64 → 197 ms
- 128 → 215 ms

Tile size 32 worked best. Size 32 best balanced loop overhead with cache reuse.

Q2: DRAM Access Ratio

Naive vs Blocked is roughly 32:1. Blocking with tile size 32 reduces memory traffic by a factor of ~32 because of better cache locality.

python3 gpt149.py part2

Output :

Running Part 2 Test: Unfused Attention with Blocked Matmul

-----RUNNING REFERENCE IMPLEMENTATION-----

manual attention == pytorch attention True

Manual Execution Time: 0.4500565528869629

Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	CPU Mem	Self CPU Mem	# of Calls
aten::empty	0.02%	91.000us	0.02%	91.000us	30.333us	5.00 Mb	5.00 Mb	3
REFERENCE - BLOCKED MATMUL + UNFUSED SOFTMAX	98.85%	444.907ms	99.98%	450.014ms	450.014ms	4.50 Mb	-1.00 Mb	1
aten::zeros	0.01%	65.000us	0.80%	3.595ms	1.798ms	4.50 Mb	0 b	2
aten::clone	0.02%	68.000us	0.31%	1.402ms	701.000us	1.00 Mb	0 b	2
model_inference	0.02%	91.000us	100.00%	450.105ms	450.105ms	512.00 Kb	-4.00 Mb	1
aten::flatten	0.02%	79.000us	0.20%	915.000us	183.000us	512.00 Kb	0 b	5
aten::empty_like	0.00%	16.000us	0.01%	35.000us	35.000us	512.00 Kb	0 b	1
aten::empty_strided	0.01%	45.000us	0.01%	45.000us	45.000us	512.00 Kb	512.00 Kb	1
aten::zero_	0.01%	64.000us	0.77%	3.458ms	1.729ms	0 b	0 b	2
aten::fill_	0.75%	3.394ms	0.75%	3.394ms	1.697ms	0 b	0 b	2

Self CPU time total: 450.105ms

REFERENCE - BLOCKED MATMUL + UNFUSED SOFTMAX statistics

cpu time: 450.014ms

mem usage: 4718592 bytes

-----RUNNING STUDENT IMPLEMENTATION-----

manual attention == pytorch attention True

Manual Execution Time: 0.24812936782836914

Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	CPU Mem	Self CPU Mem	# of Calls
aten::empty	0.01%	33.000us	0.01%	33.000us	11.000us	5.00 Mb	5.00 Mb	3
STUDENT - BLOCKED MATMUL + UNFUSED SOFTMAX	99.39%	246.652ms	99.96%	248.085ms	248.085ms	4.50 Mb	-1.00 Mb	1
aten::zeros	0.01%	35.000us	0.28%	686.000us	343.000us	4.50 Mb	0 b	2
aten::clone	0.02%	53.000us	0.27%	669.000us	334.500us	1.00 Mb	0 b	2

model_inference	0.04%	89.000us	100.00%	248.174ms	248.174ms	512.00 Kb	-4.00 Mb	1
aten::flatten	0.02%	48.000us	0.17%	426.000us	85.200us	512.00 Kb	0 b	5
aten::empty_like	0.00%	7.000us	0.01%	13.000us	13.000us	512.00 Kb	0 b	1
aten::empty_strided	0.01%	17.000us	0.01%	17.000us	17.000us	512.00 Kb	512.00 Kb	1
aten::zero_	0.01%	20.000us	0.25%	624.000us	312.000us	0 b	0 b	2
aten::fill_	0.24%	604.000us	0.24%	604.000us	302.000us	0 b	0 b	2

Self CPU time total: 248.174ms

STUDENT - BLOCKED MATMUL + UNFUSED SOFTMAX statistics

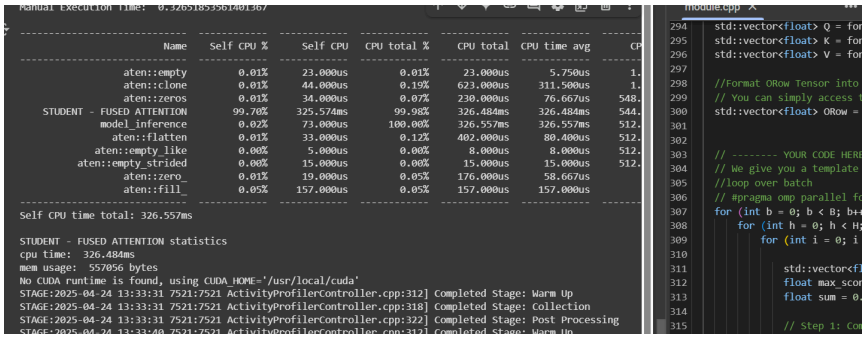
cpu time: 248.085ms

mem usage: 4718592 bytes

Part 3: Fused Attention

Q1: Why faster & smaller memory?
Fused steps eliminate intermediates, use cache more efficiently, and access tensors in a tight fashion.
No QK^T or softmax matrices in memory. Output is computed row-by-row and discarded.

Q2: Without OpenMP
CPU time increases from 236.778ms to 326.557 ms.



Q3: Multithreading
Each row of output can be computed independently, no shared data or writes, making it perfect for multithreading. Unlike Parts 1 & 2, there's no need for synchronization or shared buffers.

Running Part 3 Test: Fused Attention

-----RUNNING REFERENCE IMPLEMENTATION-----

manual attention == pytorch attention True

Manual Execution Time: 0.29456400871276855

Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	CPU Mem	Self CPU Mem	# of Calls
aten::empty	0.02%	50.000us	0.02%	50.000us	16.667us	1.03 Mb	1.03 Mb	3
aten::clone	0.03%	79.000us	0.35%	1.033ms	516.500us	1.00 Mb	0 b	2
REFERENCE - FUSED ATTENTION	92.63%	272.903ms	99.97%	294.525ms	294.525ms	544.00 Kb	-1.00 Mb	1
aten::zeros	0.02%	49.000us	0.15%	438.000us	219.000us	544.00 Kb	0 b	2
model_inference	0.03%	85.000us	100.00%	294.610ms	294.610ms	512.00 Kb	-32.00 Kb	1
aten::flatten	5.58%	16.430ms	6.03%	17.760ms	34.419us	512.00 Kb	0 b	516
aten::empty_like	0.00%	13.000us	0.01%	30.000us	30.000us	512.00 Kb	0 b	1
aten::empty_strided	0.01%	34.000us	0.01%	34.000us	34.000us	512.00 Kb	512.00 Kb	1
aten::zero_	0.01%	38.000us	0.12%	356.000us	178.000us	0 b	0 b	2
aten::fill_	0.11%	318.000us	0.11%	318.000us	318.000us	0 b	0 b	1

Self CPU time total: 294.610ms

REFERENCE - FUSED ATTENTION statistics

cpu time: 294.525ms

mem usage: 557056 bytes

-----RUNNING STUDENT IMPLEMENTATION-----

manual attention == pytorch attention True

Manual Execution Time: 0.23673725128173828

Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	CPU Mem	Self CPU Mem	# of Calls
aten::empty	0.01%	28.000us	0.01%	28.000us	7.000us	1.04 Mb	1.04 Mb	4
aten::clone	0.02%	47.000us	0.26%	614.000us	307.000us	1.00 Mb	0 b	2
aten::zeros	0.02%	41.000us	0.11%	261.000us	87.000us	548.00 Kb	0 b	3

STUDENT - FUSED ATTENTION	99.57%	235.764ms	99.97%	236.699ms	236.699ms	544.00 Kb	-1.00 Mb	1
model_inference	0.03%	79.000us	100.00%	236.778ms	236.778ms	512.00 Kb	-32.00 Kb	1
aten::flatten	0.02%	36.000us	0.19%	454.000us	90.800us	512.00 Kb	0 b	5
aten::empty_like	0.00%	5.000us	0.00%	9.000us	9.000us	512.00 Kb	0 b	1
aten::empty_strided	0.01%	18.000us	0.01%	18.000us	18.000us	512.00 Kb	512.00 Kb	1
aten::zero_	0.01%	22.000us	0.08%	196.000us	65.333us	0 b	0 b	3
aten::fill_	0.07%	174.000us	0.07%	174.000us	174.000us	0 b	0 b	1

Self CPU time total: 236.778ms

STUDENT - FUSED ATTENTION statistics

cpu time: 236.699ms

mem usage: 557056 bytes

Part 4: Flash Attention

Q1: Memory Usage
1572928 bytes, reduced memory usage. No full QK^T or P, only tile-sized buffers for Qi, Kj, Vj, PV, etc.

Q2: Slower
More complex logic and indexing per tile. Not yet parallelized. More passes per output row increase overhead.

Optimizations Possible

- Add OpenMP over batch/head/row
- Use SIMD (e.g., ISPC) for dot products and softmax
- Manually unroll inner loops
- Tune Br/Bc tile sizes

Running Part 4 Test: Flash Attention

-----RUNNING REFERENCE IMPLEMENTATION-----

manual attention == pytorch attention True

Manual Execution Time: 0.6449627876281738

Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	CPU Mem	Self CPU Mem	# of Calls
aten::zeros	0.02%	97.000us	0.62%	4.021ms	287.214us	9.16 Mb	0 b	14
aten::empty	0.02%	105.000us	0.02%	105.000us	7.500us	9.16 Mb	9.16 Mb	14
model_inference	0.05%	324.000us	100.00%	645.011ms	645.011ms	512.00 Kb	-679.00 Kb	1
REFERENCE - FLASH ATTENTION	96.75%	624.019ms	99.87%	644.164ms	644.164ms	512.00 Kb	-8.00 Mb	1
aten::zero_	0.32%	2.067ms	3.16%	20.374ms	55.065us	0 b	0 b	370
aten::fill_	2.85%	18.399ms	2.85%	18.399ms	138.338us	0 b	0 b	133

Self CPU time total: 645.011ms

REFERENCE - FLASH ATTENTION statistics

cpu time: 644.164ms

mem usage: 524288 bytes

-----RUNNING STUDENT IMPLEMENTATION-----

manual attention == pytorch attention True

Manual Execution Time: 0.3316774368286133

Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	CPU Mem	Self CPU Mem	# of Calls
aten::empty	0.01%	47.000us	0.01%	47.000us	3.615us	1.66 Mb	1.66 Mb	13
aten::zeros	0.02%	67.000us	0.16%	545.000us	45.417us	1.16 Mb	0 b	12
aten::clone	0.02%	57.000us	0.19%	628.000us	314.000us	1.00 Mb	0 b	2

model_inference 0.07% 217.000us 100.00% 331.715ms 331.715ms 512.00 Kb -679.00 Kb 1

STUDENT - FLASH ATTENTION 99.55% 330.218ms 99.81% 331.100ms 331.100ms 512.00 Kb -1.00 Mb 1

aten::flatten 0.02% 77.000us 0.14% 448.000us 29.867us 512.00 Kb 0 b 15

aten::empty_like 0.00% 11.000us 0.01% 18.000us 18.000us 512.00 Kb 0 b 1

aten::empty_strided 0.00% 14.000us 0.00% 14.000us 14.000us 512.00 Kb 512.00 Kb 1

aten::zero_ 0.01% 41.000us 0.13% 438.000us 36.500us 0 b 0 b 12

aten::fill_ 0.12% 397.000us 0.12% 397.000us 132.333us 0 b 0 b 3

Self CPU time total: 331.715ms

STUDENT - FLASH ATTENTION statistics

cpu time: 331.1ms

mem usage: 524288 bytes
