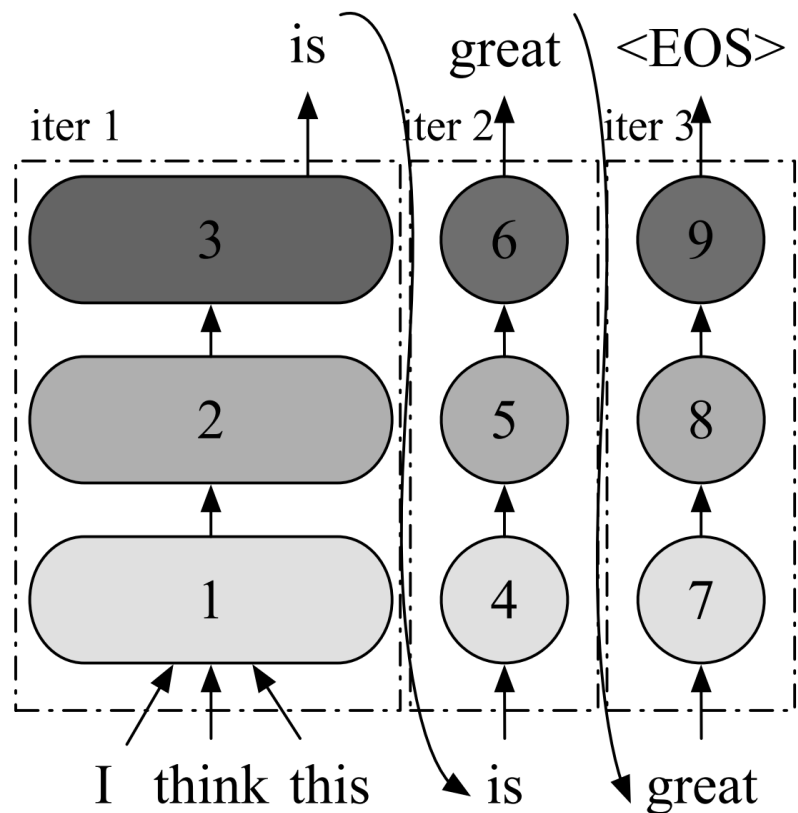


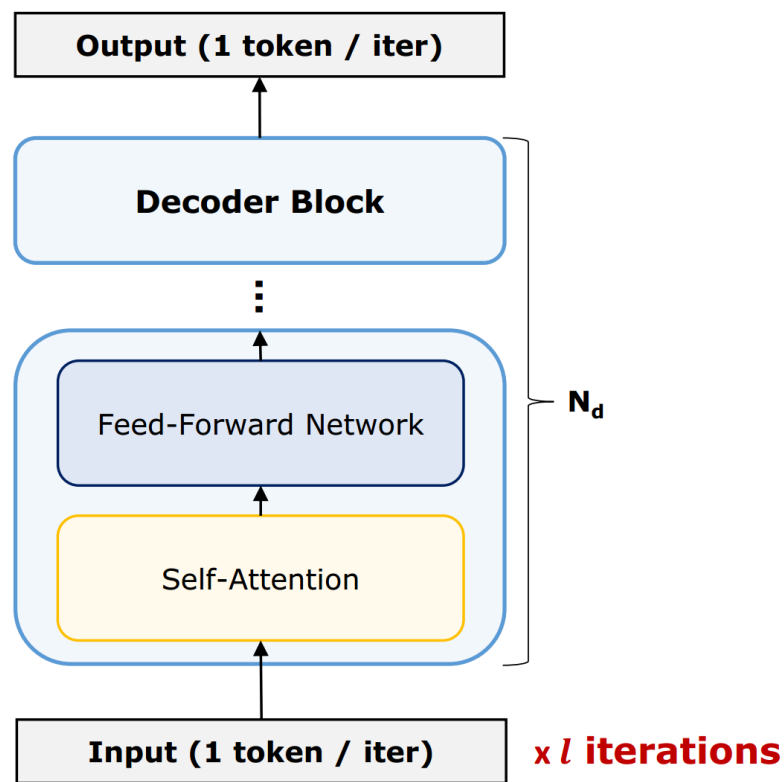
大语言模型的异构计算和加速

戴金权 (Jason Dai)
英特尔院士

自回归大语言模型(基于Transformer解码器架构)

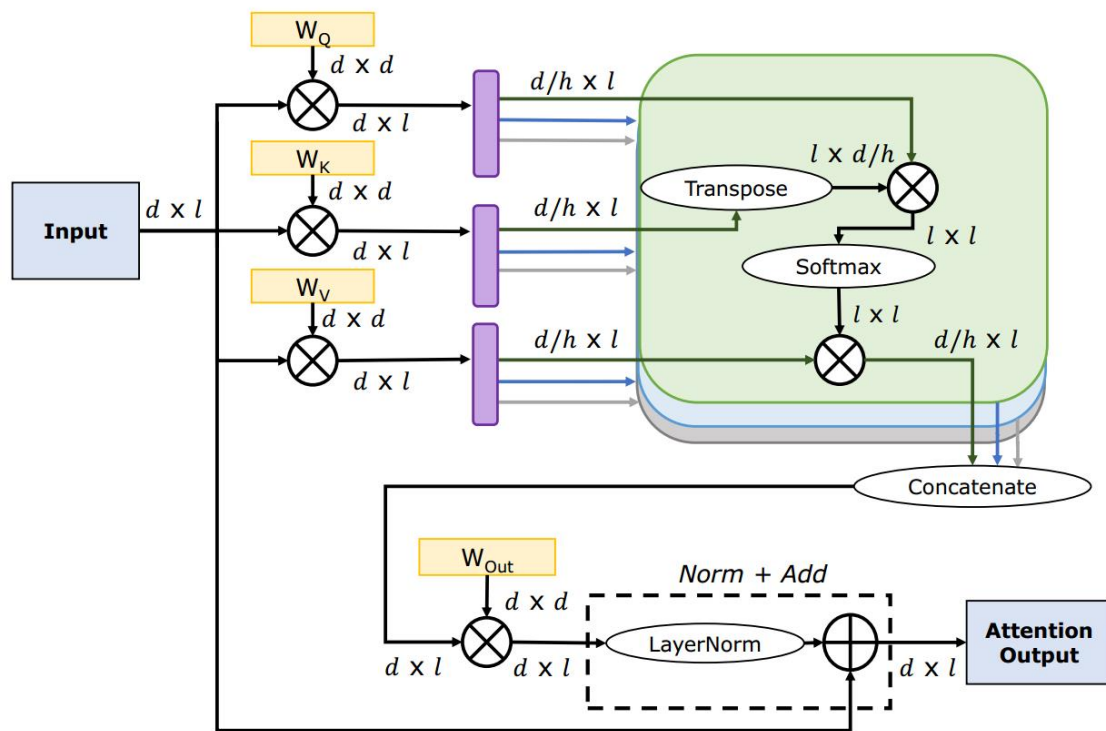


自回归大语言模型：预测下一个token

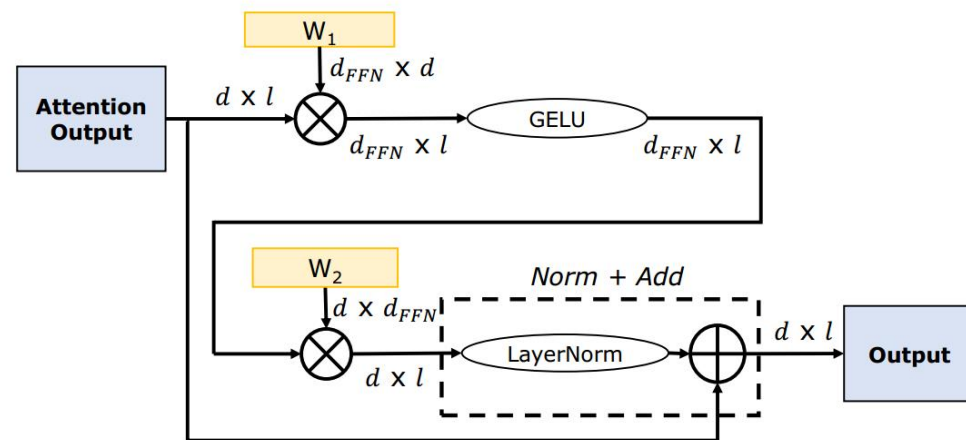


Transformer解码器架构

Transformer 解码器架构



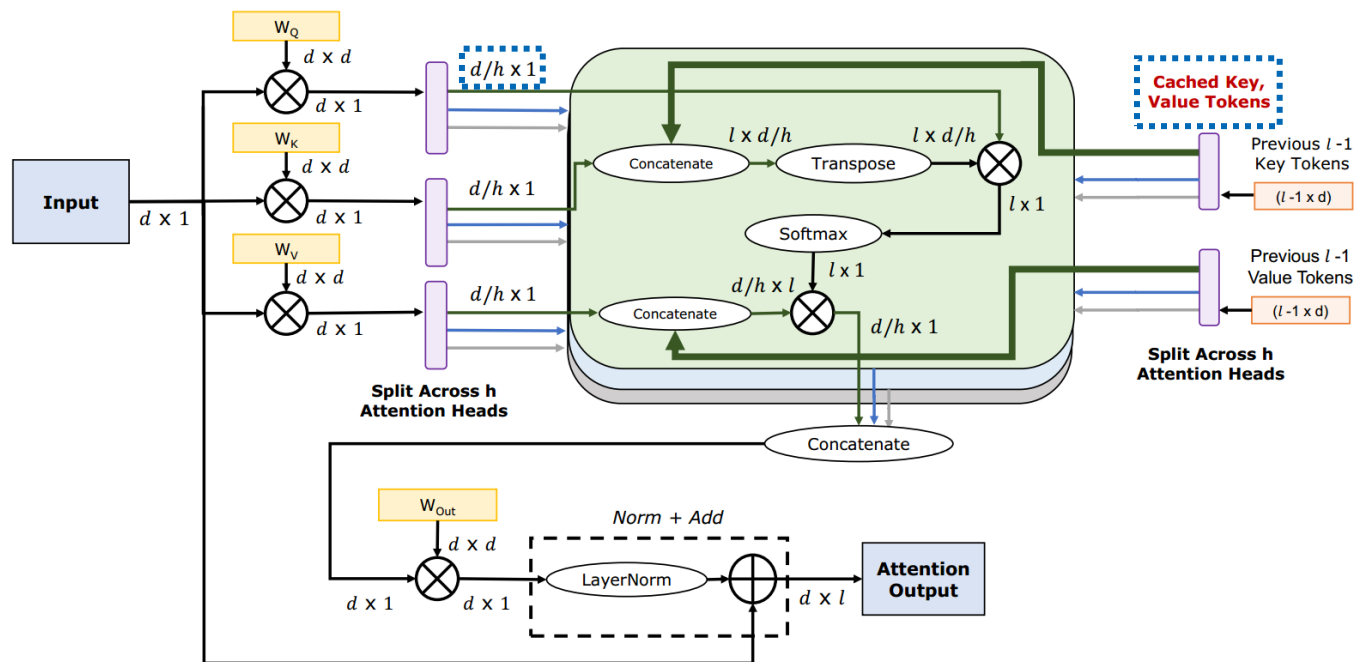
Muti-Head Attention (MHA) Module



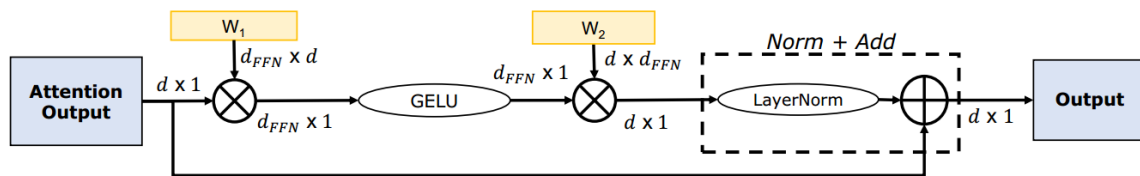
Feed-Forward Network (FFN) Module

训练; 推理 (第一个token/Prefill)

Transformer 解码器架构



Multi-Head Attention (MHA) Module



Feed-Forward Network (FFN) Module

推理 (下一个token/Decode)

大语言模型推理和训练瓶颈

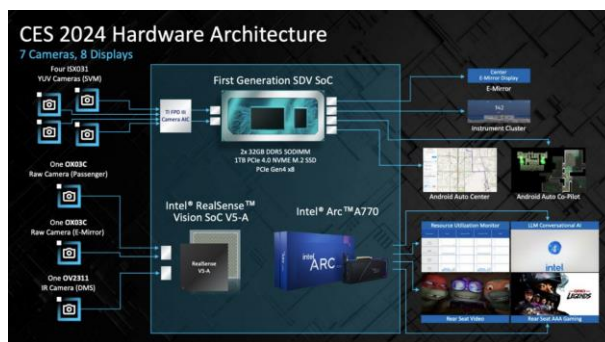
- 内存带宽
- 计算
- 显存大小
- 分布式计算 (互联)

大模型的异构计算和加速

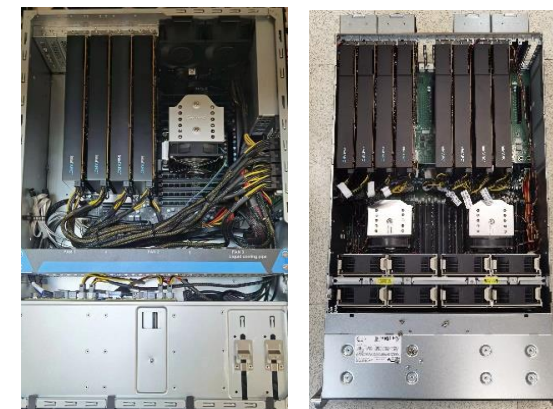
- XPU异构计算
 - CPU, GPU, NPU硬件加速



客户端
(Intel Core Ultra AI PC)



边缘端
(Intel AI座舱)



服务器
(Intel Xeon+Intel Arc GPUs)

大模型的异构计算和加速

■ 低比特计算

- 模型量化/压缩 ($W \times A_y$)
- 数据类型 (INT_x, FP_x)
- 低比特算子
- 显存(如kv cache) 使用量
- 训练、微调 (如QLoRA)

低比特大模型的精度

困惑度 (Wikitext数据集)

Perplexity	sym_int4	q4_k	fp6	fp8_e5m2	fp8_e4m3	fp16
Llama-2-7B-chat-hf	6.364	6.218	6.092	6.180	6.098	6.096
Mistral-7B-Instruct-v0.2	5.365	5.320	5.270	5.273	5.246	5.244
Baichuan2-7B-chat	6.734	6.727	6.527	6.539	6.488	6.508
Qwen1.5-7B-chat	8.865	8.816	8.557	8.846	8.530	8.607
Llama-3.1-8B-Instruct	6.705	6.566	6.338	6.383	6.325	6.267
gemma-2-9b-it	7.541	7.412	7.269	7.380	7.268	7.270
Baichuan2-13B-Chat	6.313	6.160	6.070	6.145	6.086	6.031
Llama-2-13b-chat-hf	5.449	5.422	5.341	5.384	5.332	5.329
Qwen1.5-14B-Chat	7.529	7.520	7.367	7.504	7.297	7.334

大模型的异构计算和加速

- 推理算法优化
 - Self-speculative decoding
 - KV Cache compression
 - Sliding window attention
 - Sparse attention
 - Flash attention/decoding
 - Continuous batching
 - Prefill/decoding disaggregation
 - ...

IPEX-LLM: 开源大模型XPU加速框架



 Users/Developers

Python (PyTorch) Ecosystem

HuggingFace,
Langchain,
LlamaIndex,
DeepSpeed,
TRL, Axolotl,
...

llama.cpp Ecosystem

llama.cpp,
Ollama,
LangChain.js,
Open WebUI,
...

IPEX-LLM Library

XPU Compute

LLM Acceleration

Intel XPU

<https://github.com/intel-analytics/ipex-llm/>

英特尔 XPU 大模型加速体验

Intel UHD/Iris iGPU

llama.cpp + IPEX-LLM (Phi-3-mini, Q4_0)

```
llama_new_context_with_model: freq_scale = 1
llama_kv_cache_init: SYCL0 KV buffer size = 192.00 MiB
llama_new_context_with_model: KV self size = 192.00 MiB, K (f16): 96.00 MiB, V (f16): 96.00 MiB
llama_new_context_with_model: SYCL_Host output buffer size = 0.12 MiB
llama_new_context_with_model: SYCL0 compute buffer size = 68.62 MiB
llama_new_context_with_model: SYCL_Host compute buffer size = 7.01 MiB
llama_new_context_with_model: graph nodes = 1062
llama_new_context_with_model: graph splits = 2

system_info: n_threads = 8 / 20 | AVX = 1 | AVX_VNNI = 0 | AVX2 = 1 | AVX512 = 0 | AVX512_VBMI = 0 | AVX512_VNNI = 0 | FMA = 1 |
NEON = 0 | ARM_FMA = 0 | F16C = 1 | FP16_VA = 0 | WASH_SIMD = 0 | BLAS = 1 | SSE3 = 1 | SSSE3 = 1 | VSX = 0 | MATMUL_INT8 = 0 |
main: interactive mode on.
Reverse prompt: 'User:'
Input prefix: ' '
Input suffix: '
Assistant:'
sampling:
  repeat_last_n = 64, repeat_penalty = 1.000, frequency_penalty = 0.000, presence_penalty = 0.000
  top_k = 40, tfs_z = 1.000, top_p = 0.950, min_p = 0.050, typical_p = 1.000, temp = 0.800
  mirostat = 0, mirostat_lr = 0.100, mirostat_ent = 5.000
sampling order:
CFG -> Penalties -> top_k -> tfs_z -> typical_p -> top_p -> min_p -> temperature
generate: n_ctx = 512, n_batch = 2048, n_predict = -2, n_keep = 1

== Running in interactive mode. ==
- Press Ctrl+C to interject at any time.
- Press Return to return control to LLaMa.
- To return control without starting a new line, end your input with '/'.
- If you want to submit another line, end your input with '\'.

<s> User: Hi!
Assistant: Hello. I am an AI chatbot. Would you like to talk?
User: Sure!
Assistant: What would you like to talk about?
User: Can you tell me what is CPU?

Assistant: Of course! A CPU, or Central Processing Unit, is essentially the brain of a computer or any other digital device.
```

Intel Core Ultra AI PC

Ollama + IPEX-LLM (Mistral-7B, Q4_K_M)

```
"/api/chat"
INFO [print_timings] prompt eval time = 800.93 ms / 8 tokens (
100.12 ms per token, 9.99 tokens per second) | n_prompt_tokens_processed
=8 n_tokens_second=9.988338614667375 slot_id=0 t_prompt_processing=800.934 t
_token=100.11675 task_id=200 tid="13240" timestamp=1717379415
INFO [print_timings] generation eval time = 7957.55 ms / 111 runs (
71.69 ms per token, 13.95 tokens per second) | n_decoded=111 n_tokens_se
cond=13.949025735952482 slot_id=0 t_token=71.6895945945946 t_token_generatio
n=7957.545 task_id=200 tid="13240" timestamp=1717379415
INFO [print_timings] total time = 8758.48 ms | slot_id=0 t_prom
pt_processing=800.934 t_token_generation=7957.545 t_total=8758.479 task_id=2
00 tid="13240" timestamp=1717379415
[GIN] 2024/06/03 - 09:50:15 | 200 | 8.7646436s | 127.0.0.1 | POST
"/api/chat"
INFO [print_timings] prompt eval time = 814.79 ms / 14 tokens (
58.20 ms per token, 17.18 tokens per second) | n_prompt_tokens_processed
=14 n_tokens_second=17.182383638443373 slot_id=0 t_prompt_processing=814.788
t_token=58.19914285714286 task_id=320 tid="13240" timestamp=1717379436
INFO [print_timings] generation eval time = 8249.87 ms / 115 runs (
71.74 ms per token, 13.94 tokens per second) | n_decoded=115 n_tokens_se
cond=13.939618662943293 slot_id=0 t_token=71.73797391304348 t_token_generati
on=8249.867 task_id=320 tid="13240" timestamp=1717379436
INFO [print_timings] total time = 9064.66 ms | slot_id=0 t_prom
pt_processing=814.788 t_token_generation=8249.867 t_total=9064.655 task_id=3
20 tid="13240" timestamp=1717379436
[GIN] 2024/06/03 - 09:50:36 | 200 | 9.072663s | 127.0.0.1 | POST
"/api/chat"
[GIN] 2024/06/03 - 09:50:58 | 200 | 0s | 127.0.0.1 | HEAD
"/"
[GIN] 2024/06/03 - 09:50:58 | 200 | 1.0477ms | 127.0.0.1 | POST
"/api/show"
[GIN] 2024/06/03 - 09:50:58 | 200 | 1.0722ms | 127.0.0.1 | POST
"/api/show"
[GIN] 2024/06/03 - 09:50:58 | 200 | 1.7114ms | 127.0.0.1 | POST
"/api/chat"
[GIN] 2024/06/03 - 09:51:04 | 200 | 3.0357852s | 127.0.0.1 | POST
"/api/chat"
[GIN] 2024/06/03 - 09:51:19 | 200 | 0s | 127.0.0.1 | HEAD
"/"
[GIN] 2024/06/03 - 09:51:19 | 200 | 1.5666ms | 127.0.0.1 | POST
"/api/show"
[GIN] 2024/06/03 - 09:51:19 | 200 | 1.0299ms | 127.0.0.1 | POST
"/api/show"
[GIN] 2024/06/03 - 09:51:19 | 200 | 1.7533ms | 127.0.0.1 | POST
"/api/chat"
```

```
(steven-llm-cpp) C:\Users\arda\sicheng>ollama run example
>>> What is AI?
.
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/727030045200010001>