

```
In [1]: from transformers import AutoModelForCausalLM, AutoTokenizer
import torch
from torch.amp import autocast, GradScaler
from pytorch_memlab import MemReporter
import gc
```

```

In [2]: model_directory = r"Z:\llmfile\Llama-2-7B-hf\models--meta-llama--Llama-2-7b-hf/snapshots/01c7f73d771dfac7d29232.

model = AutoModelForCausalLM.from_pretrained(model_directory)
tokenizer = AutoTokenizer.from_pretrained(model_directory)

# 设置优化器(Adam) 和 GradScaler 用于混合精度
optimizer = torch.optim.AdamW(model.parameters(), lr=1e-5)
scaler = GradScaler()

# 初始输入和标签
input_text = "Hello"
input_ids = tokenizer(input_text, return_tensors="pt").input_ids
labels = input_ids.clone() # 使用输入作为标签

reporter = MemReporter(model)

# 反向传播实验
print("==BackwardTrainingMemoryUsage==")
model.train()
optimizer.zero_grad() # 清除之前的梯度

# 重新执行前向传播, 得到loss
with autocast("cuda"):
    outputs = model(input_ids, labels=labels)
    loss = outputs.loss

# 反向传播
scaler.scale(loss).backward() # 混合精度缩放梯度
reporter.report() # 记录反向传播后的内存使用情况

```

```

Loading checkpoint shards: 0%|          | 0/2 [00:00<?, ?it/s]
==BackwardTrainingMemoryUsage==
Element type                                Size  Used MEM

```

Storage on cpu

Tensor0	(1, 2, 4096)	32.00K
Tensor1	(1, 32, 2, 128)	0.00B
Tensor2	(1, 32, 2, 128)	32.00K
Tensor3	(1, 2, 32000)	250.00K
Tensor4	(1,)	512.00B
Tensor5	(1,)	512.00B
Tensor6	(1,)	512.00B
Tensor7	(1, 2)	512.00B
Tensor8	(1, 2)	512.00B
Tensor9	(1, 2, 4096)	32.00K
Tensor10	(1, 32, 2, 128)	0.00B
Tensor11	(1, 32, 2, 128)	32.00K
Tensor12	(1, 2, 4096)	32.00K
Tensor13	(1, 32, 2, 128)	0.00B
Tensor14	(1, 32, 2, 128)	32.00K
Tensor15	(1, 2, 4096)	32.00K
Tensor16	(1, 32, 2, 128)	0.00B
Tensor17	(1, 32, 2, 128)	32.00K
Tensor18	(1, 2, 4096)	32.00K
Tensor19	(1, 32, 2, 128)	0.00B
Tensor20	(1, 32, 2, 128)	32.00K
Tensor21	(1, 2, 4096)	32.00K
Tensor22	(1, 32, 2, 128)	0.00B
Tensor23	(1, 32, 2, 128)	32.00K
Tensor24	(1, 2, 4096)	32.00K
Tensor25	(1, 32, 2, 128)	0.00B
Tensor26	(1, 32, 2, 128)	32.00K
Tensor27	(1, 2, 4096)	32.00K
Tensor28	(1, 32, 2, 128)	0.00B
Tensor29	(1, 32, 2, 128)	32.00K
Tensor30	(1, 2, 4096)	32.00K
Tensor31	(1, 32, 2, 128)	0.00B
Tensor32	(1, 32, 2, 128)	32.00K
Tensor33	(1, 2, 4096)	32.00K
Tensor34	(1, 32, 2, 128)	0.00B
Tensor35	(1, 32, 2, 128)	32.00K
Tensor36	(1, 2, 4096)	32.00K
Tensor37	(1, 32, 2, 128)	0.00B
Tensor38	(1, 32, 2, 128)	32.00K
Tensor39	(1, 2, 4096)	32.00K
Tensor40	(1, 32, 2, 128)	0.00B

Tensor41	(1, 32, 2, 128)	32.00K
Tensor42	(1, 2, 4096)	32.00K
Tensor43	(1, 32, 2, 128)	0.00B
Tensor44	(1, 32, 2, 128)	32.00K
Tensor45	(1, 2, 4096)	32.00K
Tensor46	(1, 32, 2, 128)	0.00B
Tensor47	(1, 32, 2, 128)	32.00K
Tensor48	(1, 2, 4096)	32.00K
Tensor49	(1, 32, 2, 128)	0.00B
Tensor50	(1, 32, 2, 128)	32.00K
Tensor51	(1, 2, 4096)	32.00K
Tensor52	(1, 32, 2, 128)	0.00B
Tensor53	(1, 32, 2, 128)	32.00K
Tensor54	(1, 2, 4096)	32.00K
Tensor55	(1, 32, 2, 128)	0.00B
Tensor56	(1, 32, 2, 128)	32.00K
Tensor57	(1, 2, 4096)	32.00K
Tensor58	(1, 32, 2, 128)	0.00B
Tensor59	(1, 32, 2, 128)	32.00K
Tensor60	(1, 2, 4096)	32.00K
Tensor61	(1, 32, 2, 128)	0.00B
Tensor62	(1, 32, 2, 128)	32.00K
Tensor63	(1, 2, 4096)	32.00K
Tensor64	(1, 32, 2, 128)	0.00B
Tensor65	(1, 32, 2, 128)	32.00K
Tensor66	(1, 2, 4096)	32.00K
Tensor67	(1, 32, 2, 128)	0.00B
Tensor68	(1, 32, 2, 128)	32.00K
Tensor69	(1, 2, 4096)	32.00K
Tensor70	(1, 32, 2, 128)	0.00B
Tensor71	(1, 32, 2, 128)	32.00K
Tensor72	(1, 2, 4096)	32.00K
Tensor73	(1, 32, 2, 128)	0.00B
Tensor74	(1, 32, 2, 128)	32.00K
Tensor75	(1, 2, 4096)	32.00K
Tensor76	(1, 32, 2, 128)	0.00B
Tensor77	(1, 32, 2, 128)	32.00K
Tensor78	(1, 2, 4096)	32.00K
Tensor79	(1, 32, 2, 128)	0.00B
Tensor80	(1, 32, 2, 128)	32.00K
Tensor81	(1, 2, 4096)	32.00K
Tensor82	(1, 32, 2, 128)	0.00B
Tensor83	(1, 32, 2, 128)	32.00K
Tensor84	(1, 2, 4096)	32.00K
Tensor85	(1, 32, 2, 128)	0.00B
Tensor86	(1, 32, 2, 128)	32.00K
Tensor87	(1, 2, 4096)	32.00K
Tensor88	(1, 32, 2, 128)	0.00B
Tensor89	(1, 32, 2, 128)	32.00K
Tensor90	(1, 2, 4096)	32.00K
Tensor91	(1, 32, 2, 128)	0.00B
Tensor92	(1, 32, 2, 128)	32.00K
Tensor93	(1, 2, 4096)	32.00K
Tensor94	(1, 32, 2, 128)	0.00B
Tensor95	(1, 32, 2, 128)	32.00K
Tensor96	(1, 2, 4096)	32.00K
Tensor97	(1, 32, 2, 128)	0.00B
Tensor98	(1, 32, 2, 128)	32.00K
Tensor99	(1, 2, 4096)	32.00K
Tensor100	(1, 32, 2, 128)	0.00B
Tensor101	(1, 32, 2, 128)	32.00K
lm_head.weight	(32000, 4096)	500.00M
lm_head.weight.grad	(32000, 4096)	500.00M
Tensor102	(64,)	512.00B
model.embed_tokens.weight	(32000, 4096)	500.00M
model.embed_tokens.weight.grad	(32000, 4096)	500.00M
model.norm.weight	(4096,)	16.00K
model.norm.weight.grad	(4096,)	16.00K
model.layers.0.input_layernorm.weight	(4096,)	16.00K
model.layers.0.input_layernorm.weight.grad	(4096,)	16.00K
model.layers.0.post_attention_layernorm.weight	(4096,)	16.00K
model.layers.0.post_attention_layernorm.weight.grad	(4096,)	16.00K
model.layers.1.input_layernorm.weight	(4096,)	16.00K
model.layers.1.input_layernorm.weight.grad	(4096,)	16.00K
model.layers.1.post_attention_layernorm.weight	(4096,)	16.00K
model.layers.1.post_attention_layernorm.weight.grad	(4096,)	16.00K
model.layers.2.input_layernorm.weight	(4096,)	16.00K
model.layers.2.input_layernorm.weight.grad	(4096,)	16.00K
model.layers.2.post_attention_layernorm.weight	(4096,)	16.00K
model.layers.2.post_attention_layernorm.weight.grad	(4096,)	16.00K
model.layers.3.input_layernorm.weight	(4096,)	16.00K
model.layers.3.input_layernorm.weight.grad	(4096,)	16.00K
model.layers.3.post_attention_layernorm.weight	(4096,)	16.00K

[illegible]

model.layers.24.post_attention_layernorm.weight	(4096,)	16.00K
model.layers.24.post_attention_layernorm.weight.grad	(4096,)	16.00K
model.layers.25.input_layernorm.weight	(4096,)	16.00K
model.layers.25.input_layernorm.weight.grad	(4096,)	16.00K
model.layers.25.post_attention_layernorm.weight	(4096,)	16.00K
model.layers.25.post_attention_layernorm.weight.grad	(4096,)	16.00K
model.layers.26.input_layernorm.weight	(4096,)	16.00K
model.layers.26.input_layernorm.weight.grad	(4096,)	16.00K
model.layers.26.post_attention_layernorm.weight	(4096,)	16.00K
model.layers.26.post_attention_layernorm.weight.grad	(4096,)	16.00K
model.layers.27.input_layernorm.weight	(4096,)	16.00K
model.layers.27.input_layernorm.weight.grad	(4096,)	16.00K
model.layers.27.post_attention_layernorm.weight	(4096,)	16.00K
model.layers.27.post_attention_layernorm.weight.grad	(4096,)	16.00K
model.layers.28.input_layernorm.weight	(4096,)	16.00K
model.layers.28.input_layernorm.weight.grad	(4096,)	16.00K
model.layers.28.post_attention_layernorm.weight	(4096,)	16.00K
model.layers.28.post_attention_layernorm.weight.grad	(4096,)	16.00K
model.layers.29.input_layernorm.weight	(4096,)	16.00K
model.layers.29.input_layernorm.weight.grad	(4096,)	16.00K
model.layers.29.post_attention_layernorm.weight	(4096,)	16.00K
model.layers.29.post_attention_layernorm.weight.grad	(4096,)	16.00K
model.layers.30.input_layernorm.weight	(4096,)	16.00K
model.layers.30.input_layernorm.weight.grad	(4096,)	16.00K
model.layers.30.post_attention_layernorm.weight	(4096,)	16.00K
model.layers.30.post_attention_layernorm.weight.grad	(4096,)	16.00K
model.layers.31.input_layernorm.weight	(4096,)	16.00K
model.layers.31.input_layernorm.weight.grad	(4096,)	16.00K
model.layers.31.post_attention_layernorm.weight	(4096,)	16.00K
model.layers.31.post_attention_layernorm.weight.grad	(4096,)	16.00K
Tensor103	(64,)	512.00B
Tensor104	(64,)	512.00B
Tensor105	(64,)	512.00B
Tensor106	(64,)	512.00B
Tensor107	(64,)	512.00B
Tensor108	(64,)	512.00B
Tensor109	(64,)	512.00B
Tensor110	(64,)	512.00B
Tensor111	(64,)	512.00B
Tensor112	(64,)	512.00B
Tensor113	(64,)	512.00B
Tensor114	(64,)	512.00B
Tensor115	(64,)	512.00B
Tensor116	(64,)	512.00B
Tensor117	(64,)	512.00B
Tensor118	(64,)	512.00B
Tensor119	(64,)	512.00B
Tensor120	(64,)	512.00B
Tensor121	(64,)	512.00B
Tensor122	(64,)	512.00B
Tensor123	(64,)	512.00B
Tensor124	(64,)	512.00B
Tensor125	(64,)	512.00B
Tensor126	(64,)	512.00B
Tensor127	(64,)	512.00B
Tensor128	(64,)	512.00B
Tensor129	(64,)	512.00B
Tensor130	(64,)	512.00B
Tensor131	(64,)	512.00B
Tensor132	(64,)	512.00B
Tensor133	(64,)	512.00B
Tensor134	(64,)	512.00B
model.layers.0.self_attn.q_proj.weight	(4096, 4096)	64.00M
model.layers.0.self_attn.q_proj.weight.grad	(4096, 4096)	64.00M
model.layers.0.self_attn.k_proj.weight	(4096, 4096)	64.00M
model.layers.0.self_attn.k_proj.weight.grad	(4096, 4096)	64.00M
model.layers.0.self_attn.v_proj.weight	(4096, 4096)	64.00M
model.layers.0.self_attn.v_proj.weight.grad	(4096, 4096)	64.00M
model.layers.0.self_attn.o_proj.weight	(4096, 4096)	64.00M
model.layers.0.self_attn.o_proj.weight.grad	(4096, 4096)	64.00M
model.layers.0.mlp.gate_proj.weight	(11008, 4096)	172.00M
model.layers.0.mlp.gate_proj.weight.grad	(11008, 4096)	172.00M
model.layers.0.mlp.up_proj.weight	(11008, 4096)	172.00M
model.layers.0.mlp.up_proj.weight.grad	(11008, 4096)	172.00M
model.layers.0.mlp.down_proj.weight	(4096, 11008)	172.00M
model.layers.0.mlp.down_proj.weight.grad	(4096, 11008)	172.00M
model.layers.1.self_attn.q_proj.weight	(4096, 4096)	64.00M
model.layers.1.self_attn.q_proj.weight.grad	(4096, 4096)	64.00M
model.layers.1.self_attn.k_proj.weight	(4096, 4096)	64.00M
model.layers.1.self_attn.k_proj.weight.grad	(4096, 4096)	64.00M
model.layers.1.self_attn.v_proj.weight	(4096, 4096)	64.00M
model.layers.1.self_attn.v_proj.weight.grad	(4096, 4096)	64.00M
model.layers.1.self_attn.o_proj.weight	(4096, 4096)	64.00M

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

model.layers.31.self_attn.k_proj.weight	(4096, 4096)	64.00M
model.layers.31.self_attn.k_proj.weight.grad	(4096, 4096)	64.00M
model.layers.31.self_attn.v_proj.weight	(4096, 4096)	64.00M
model.layers.31.self_attn.v_proj.weight.grad	(4096, 4096)	64.00M
model.layers.31.self_attn.o_proj.weight	(4096, 4096)	64.00M
model.layers.31.self_attn.o_proj.weight.grad	(4096, 4096)	64.00M
model.layers.31.mlp.gate_proj.weight	(11008, 4096)	172.00M
model.layers.31.mlp.gate_proj.weight.grad	(11008, 4096)	172.00M
model.layers.31.mlp.up_proj.weight	(11008, 4096)	172.00M
model.layers.31.mlp.up_proj.weight.grad	(11008, 4096)	172.00M
model.layers.31.mlp.down_proj.weight	(4096, 11008)	172.00M
model.layers.31.mlp.down_proj.weight.grad	(4096, 11008)	172.00M

-----  
Total Tensors: 13477683783      Used Memory: 50.21G  
-----

C:\Users\xxc13\anaconda3\Lib\site-packages\pytorch\_memlab\mem\_reporter.py:65: FutureWarning: `torch.distributed.reduce\_op` is deprecated, please use `torch.distributed.ReduceOp` instead  
tensors = [obj for obj in objects if isinstance(obj, torch.Tensor)]  
C:\Users\xxc13\anaconda3\Lib\site-packages\pytorch\_memlab\mem\_reporter.py:95: UserWarning: TypedStorage is deprecated. It will be removed in the future and UntypedStorage will be the only storage class. This should only matter to you if you are using storages directly. To access UntypedStorage directly, use tensor.untyped\_storage() instead of tensor.storage()  
fact\_numel = tensor.storage().size()

```
In [3]: # 清除变量并释放内存
del outputs, loss
gc.collect()
```

Out[3]: 46

In [ ]: