Bert跑通symbolic_trace

1 环境准备

- 拉symbolic_trace仓库 https://github.com/2742195759/paddle-symbolic-trace
- 拉PddleNLP仓库 https://github.com/PaddlePaddle/PaddleNLP
- 拉NLP仓库的PR，这个PR已经修改了代码使得正常跑动态图的时候能够跑到symbolic_trace的逻辑（需要修改sys.path.append的路径） https://github.com/PaddlePaddle/PaddleNLP/pull/6040
- `pip install -r requirements.txt` 安装需要的包
- `pip install -e .` 安装paddlenlp
- 按照这个文档准备数据：https://github.com/PaddlePaddle/PaddleNLP/tree/develop/model_zoo/bert
- 具体的执行命令如下，执行命令前加LOG_LEVEL=3可以开启paddle-symbolic-trace的日志

</> Bash | 收起

```bash
cd /workspace/PaddleNLP/model_zoo/bert
python -m paddle.distributed.launch --gpus "0" run_pretrain.py     --model_type bert     --model_name_or_path bert-base-uncased     --max_predictions_per_seq 20     --batch_size 32     --learning_rate 1e-4     --weight_decay 1e-2     --adam_epsilon 1e-6     --warmup_steps 10000     --input_dir data/     --output_dir pretrained_models/     --logging_steps 1     --save_steps 20000     --max_steps 1000000     --device gpu     --use_amp False
```

目标：非STRICT_MODE=1模式下，保证至少有一个子图

2 已修复的问题和排查思路

2.1 COMPARE_OP

报错信息如下：

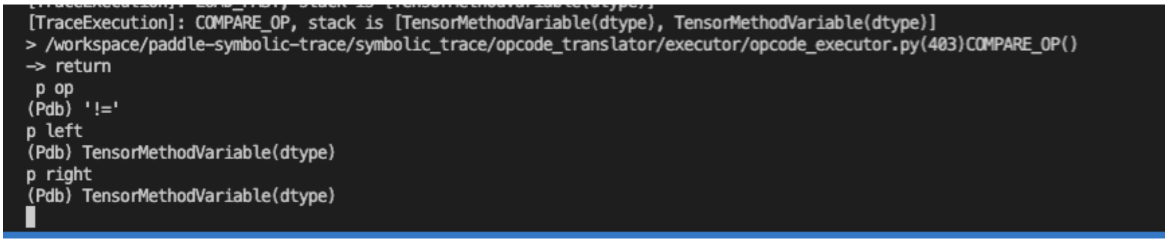
</> Python | 收起

```python
[TraceExecution]: COMPARE_OP, stack is [TensorVariable(shape: [2, 1, 1, 128], dtype: paddle.float32, stop_gradient: True), ConstantVariable(None)]
[TraceExecution]: POP_JUMP_IF_FALSE, stack is [ConstantVariable(True)]
[TraceExecution]: LOAD_FAST, stack is []
[TraceExecution]: LOAD_ATTR, stack is [TensorVariable(shape: [2, 1, 1, 128], dtype: paddle.float32, stop_gradient: True)]
[TraceExecution]: LOAD_FAST, stack is [TensorMethodVariable(dtype)]
[TraceExecution]: COMPARE_OP, stack is [TensorMethodVariable(dtype), TensorMethodVariable(dtype)]
Traceback (most recent call last):
  File "run_pretrain.py", line 481, in <module>
    do_train(args)
  File "run_pretrain.py", line 411, in do_train
    prediction_scores, seq_relationship_score = symbolic_trace(model)(
  File "/workspace/paddle-symbolic-trace/symbolic_trace/trace.py", line 14, in impl
    raise e
  File "/workspace/paddle-symbolic-trace/symbolic_trace/trace.py", line 12, in impl
    outs = func(*args, **kwargs)
  File "/root/miniconda3/envs/py3.8/lib/python3.8/site-packages/paddle/nn/layer/layers.py", line 1254, in __call__
    return self.forward(*inputs, **kwargs)
  File "/workspace/PaddleNLP/paddlenlp/transformers/bert/modeling.py", line 1038, in forward
    outputs = self.bert(
  File "/root/miniconda3/envs/py3.8/lib/python3.8/site-packages/paddle/nn/layer/layers.py", line 1254, in __call__
    return self.forward(*inputs, **kwargs)
  File "/workspace/PaddleNLP/paddlenlp/transformers/bert/modeling.py", line 468, in forward
    encoder_outputs = self.encoder(
  File "/root/miniconda3/envs/py3.8/lib/python3.8/site-packages/paddle/nn/layer/layers.py", line 1254, in __call__
    return self.forward(*inputs, **kwargs)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/transform.py", line 23, in eval_frame_callback
    new_code = InstructionTranslatorCache()(frame)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 86, in __call__
    cache_getter, (new_code, guard_fn) = self.translate(frame)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 120, in translate
    result = start_translate(frame)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 131, in start_translate
    new_code, guard_fn = simulator.transform()
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 877, in transform
    self.run()
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 202, in run
    is_stop = self.step(cur_instr)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 210, in step
    return getattr(self, instr.opname)(instr)  # run single step.
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 329, in CALL_FUNCTION
    ret = fn(*args, **kwargs)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/variables.py", line 588, in __call__
    return self.call_function(*args, **kwargs)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/variables.py", line 641, in call_function
    output = inline_executor.inline_call()
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_inline_executor.py", line 96, in inline_call
    self.run()
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 202, in run
    is_stop = self.step(cur_instr)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 210, in step
    return getattr(self, instr.opname)(instr)  # run single step.
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 389, in COMPARE_OP
    self.push(SUPPORT_COMPARE_OP[op](left, right))
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 58, in <lambda>
    x.value != y.value, None, tracker=DummyTracker([x, y])
AttributeError: 'TensorMethodVariable' object has no attribute 'value'
```

看报错是COMPARE_OP字节码报错了，可以在COMPARE_OP字节码内部添加try...expect...逻辑，在抛出异常时打一个断点，

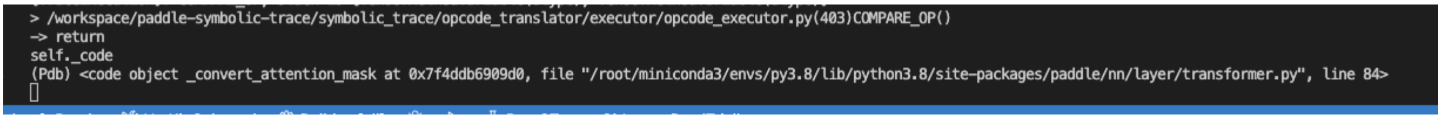
```python
    def COMPARE_OP(self, instr):
        op = instr.argval
        if op in SUPPORT_COMPARE_OP:
            right, left = self.pop(), self.pop()
            try:
                self.push(SUPPORT_COMPARE_OP[op](left, right))
            except:
                breakpoint()
            return
        else:
            raise UnsupportError(
                f"{instr} is not support. may be not a supported compare op."
            )
```

在断点处查看相应的变量，发现dtype被包成了一个TensorMethodVariable，但是tensor.dtype其实是一个attr而不是method（看报错前的stack日志也可以发现相同的问题）



```
[TraceExecution]: COMPARE_OP, stack is [TensorMethodVariable(dtype), TensorMethodVariable(dtype)]
> /workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py(403)COMPARE_OP()
-> return
 p op
(Pdb) '!='
p left
(Pdb) TensorMethodVariable(dtype)
p right
(Pdb) TensorMethodVariable(dtype)
```

如果想查看具体的报错的用户代码位置，通过self._code可以看到当前函数的路径和行数，从而找到具体python函数，可以找到对应的 `!=` 的python代码



```
> /workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py(403)COMPARE_OP()
-> return
self._code
(Pdb) <code object _convert_attention_mask at 0x7f4ddb6909d0, file "/root/miniconda3/envs/py3.8/lib/python3.8/site-packages/paddle/nn/layer/transformer.py", line 84>
```

```python
def _convert_attention_mask(attn_mask, dtype):
    """
```

```
Convert the attention mask to the target dtype we expect.

Parameters:
    attn_mask (Tensor, optional): A tensor used in multi-head attention
            to prevents attention to some unwanted positions, usually the
            paddings or the subsequent positions. It is a tensor with shape
            broadcasted to `[batch_size, n_head, sequence_length, sequence_length]`.
            When the data type is bool, the unwanted positions have `False`
            values and the others have `True` values. When the data type is
            int, the unwanted positions have 0 values and the others have 1
            values. When the data type is float, the unwanted positions have
            `-INF` values and the others have 0 values. It can be None when
            nothing wanted or needed to be prevented attention to. Default None.
    dtype (VarType): The target type of `attn_mask` we expect.

Returns:
    Tensor: A Tensor with shape same as input `attn_mask`, with data type `dtype`.
"""
if attn_mask is not None and attn_mask.dtype != dtype:
    attn_mask_dtype = convert_dtype(attn_mask.dtype)
    if attn_mask_dtype == 'bool' or 'int' in attn_mask_dtype:
        attn_mask = (paddle.cast(attn_mask, dtype) - 1.0) * 1e9
    else:
        attn_mask = paddle.cast(attn_mask, dtype)
return attn_mask
```

从代码中可以看出，从tensor中取dtype属性的时候被错误的转成了TensorMethodVariable，那就需要看TensorMethodVariable从哪里来的，这里可以在TensorMethodVariable的构造函数中self.method_name == 'dtype'时添加断点

Python | 收起

```python
def __init__(
    self,
    tensor: TensorVariable,
    method_name: str,
    graph: FunctionGraph,
    tracker: Tracker,
):
    super().__init__(tensor, graph, tracker)
    self.tensor = tensor
    self.method_name = method_name
    if self.method_name == 'dtype':
        breakpoint()
```

当程序执行到此处时，通过bt命令可以看到程序的调用栈

Python | 收起

```python
> /workspace/paddle-symbolic-
trace/symbolic_trace/opcode_translator/executor/variables.py(678)__init__()->None
```

```
-> breakpoint()
bt
(Pdb)   /workspace/PaddleNLP/model_zoo/bert/run_pretrain.py(503)<module>()
-> do_train(args)
  /workspace/PaddleNLP/model_zoo/bert/run_pretrain.py(422)do_train()
-> prediction_scores, seq_relationship_score = net_wrapper(model,
  /workspace/paddle-symbolic-trace/symbolic_trace/trace.py(12)impl()
-> outs = func(*args, **kwargs)
  /workspace/PaddleNLP/model_zoo/bert/run_pretrain.py(280)resnet_call()
-> return net(input_ids=input_ids,
  /root/miniconda3/envs/py3.8/lib/python3.8/site-packages/paddle/nn/layer/layers.py(1254)__call__()
-> return self.forward(*inputs, **kwargs)
  /workspace/PaddleNLP/paddlenlp/transformers/bert/modeling.py(1038)forward()
-> outputs = self.bert(
  /root/miniconda3/envs/py3.8/lib/python3.8/site-packages/paddle/nn/layer/layers.py(1254)__call__()
-> return self.forward(*inputs, **kwargs)
  /workspace/PaddleNLP/paddlenlp/transformers/bert/modeling.py(468)forward()
-> encoder_outputs = self.encoder(
  /root/miniconda3/envs/py3.8/lib/python3.8/site-packages/paddle/nn/layer/layers.py(1254)__call__()
-> return self.forward(*inputs, **kwargs)
  /workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/transform.py(20)eval_frame_callback()
-> new_code = InstructionTranslatorCache()(frame)
  /workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py(94)__call__()
-> cache_getter, (new_code, guard_fn) = self.translate(frame)
  /workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py(130)translate()
-> result = start_translate(frame)
  /workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py(141)start_translate()
-> new_code, guard_fn = simulator.transform()
  /workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py(890)transform()
-> self.run()
  /workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py(212)run()
-> is_stop = self.step(cur_instr)
  /workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py(220)step()
-> return getattr(self, instr.opname)(instr)  # run single step.
  /workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py(284)LOAD_ATTR()
-> self.push(getattr(obj, attr_name))
  /workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/variables.py(307)__getattr__()
-> return TensorMethodVariable(
> /workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/variables.py(678)__init__()->None
-> breakpoint()
```

从后往前看，可以看到 `/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/variables.py(307)__getattr__()` 这一行是在TensorVariable的__getattr__方法，具体代码如下：

```python
    def __getattr__(self, name: str):
        # TODO: Handle attribute case
        return TensorMethodVariable(
            self, name, self.graph, tracker=GetAttrTracker(self, name)
        )
```

这里只处理的method的情况，没有处理attr的情况。到此定位到了问题，需要添加处理attr的代码逻辑。

备注：常用的几个pdb命令可以参考：https://www.paddlepaddle.org.cn/documentation/docs/zh/guides/jit/debugging_cn.html#changyongmingling

2.2 BINARY_SUBSCR

```python
Traceback (most recent call last):
  File "run_pretrain.py", line 481, in <module>
    do_train(args)
  File "run_pretrain.py", line 411, in do_train
    prediction_scores, seq_relationship_score = symbolic_trace(model)(
  File "/workspace/paddle-symbolic-trace/symbolic_trace/trace.py", line 14, in impl
    raise e
  File "/workspace/paddle-symbolic-trace/symbolic_trace/trace.py", line 12, in impl
    outs = func(*args, **kwargs)
  File "/root/miniconda3/envs/py3.8/lib/python3.8/site-packages/paddle/nn/layer/layers.py", line 1254, in __call__
    return self.forward(*inputs, **kwargs)
  File "/workspace/PaddleNLP/paddlenlp/transformers/bert/modeling.py", line 1038, in forward
    outputs = self.bert(
  File "/root/miniconda3/envs/py3.8/lib/python3.8/site-packages/paddle/nn/layer/layers.py", line 1254, in __call__
    return self.forward(*inputs, **kwargs)
  File "/workspace/PaddleNLP/paddlenlp/transformers/bert/modeling.py", line 478, in forward
    pooled_output = self.pooler(sequence_output)
  File "/root/miniconda3/envs/py3.8/lib/python3.8/site-packages/paddle/nn/layer/layers.py", line 1254, in __call__
    return self.forward(*inputs, **kwargs)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/transform.py", line 20, in eval_frame_callback
    new_code = InstructionTranslatorCache()(frame)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 94, in __call__
    cache_getter, (new_code, guard_fn) = self.translate(frame)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 130, in translate
    result = start_translate(frame)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 141, in start_translate
    new_code, guard_fn = simulator.transform()
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 890, in transform
    self.run()
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 212, in run
    is_stop = self.step(cur_instr)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 220, in step
    return getattr(self, instr.opname)(instr)  # run single step.
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 321, in BINARY_SUBSCR
    self.push(container[key.value])
TypeError: 'TensorVariable' object is not subscriptable
```

TensorVariable没有实现getitem的导致的

2.3 转置的问题

```
Traceback (most recent call last):
  File "run_pretrain.py", line 481, in <module>
    do_train(args)
  File "run_pretrain.py", line 411, in do_train
    prediction_scores, seq_relationship_score = symbolic_trace(model)(
  File "/workspace/paddle-symbolic-trace/symbolic_trace/trace.py", line 14, in impl
    raise e
  File "/workspace/paddle-symbolic-trace/symbolic_trace/trace.py", line 12, in impl
    outs = func(*args, **kwargs)
  File "/root/miniconda3/envs/py3.8/lib/python3.8/site-packages/paddle/nn/layer/layers.py", line 1254, in __call__
    return self.forward(*inputs, **kwargs)
  File "/workspace/PaddleNLP/paddlenlp/transformers/bert/modeling.py", line 1048, in forward
    prediction_scores, seq_relationship_score = self.cls(sequence_output, pooled_output, masked_positions)
  File "/root/miniconda3/envs/py3.8/lib/python3.8/site-packages/paddle/nn/layer/layers.py", line 1254, in __call__
    return self.forward(*inputs, **kwargs)
  File "/workspace/PaddleNLP/paddlenlp/transformers/bert/modeling.py", line 927, in forward
    prediction_scores = self.predictions(sequence_output, masked_positions)
  File "/root/miniconda3/envs/py3.8/lib/python3.8/site-packages/paddle/nn/layer/layers.py", line 1254, in __call__
    return self.forward(*inputs, **kwargs)
  File "/workspace/PaddleNLP/paddlenlp/transformers/bert/modeling.py", line 877, in forward
```

```
    hidden_states = self.decoder(hidden_states)
  File "/root/miniconda3/envs/py3.8/lib/python3.8/site-packages/paddle/nn/layer/layers.py", line 1254, in __call__
    return self.forward(*inputs, **kwargs)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/transform.py", line 20, in eval_frame_callback
    new_code = InstructionTranslatorCache()(frame)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 94, in __call__
    cache_getter, (new_code, guard_fn) = self.translate(frame)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 130, in translate
    result = start_translate(frame)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 141, in start_translate
    new_code, guard_fn = simulator.transform()
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 890, in transform
    self.run()
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 212, in run
    is_stop = self.step(cur_instr)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 220, in step
    return getattr(self, instr.opname)(instr)  # run single step.
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 284, in LOAD_ATTR
    self.push(getattr(obj, attr_name))
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/variables.py", line 309, in __getattr__
    attr = getattr(self.value.meta, name)
AttributeError: 'MetaInfo' object has no attribute 'T'
```

TensorVariable未实现转置T方法

2.4 多个step后爆显存

- bs32的情况下，第3个step稳定会爆显存
- 如果将bs调整成2，前几个step 显存的变化如下，除了第二个step后面每个step基本稳定增长300m

  2042MiB

  3976MiB

  4284MiB

  4594MiB

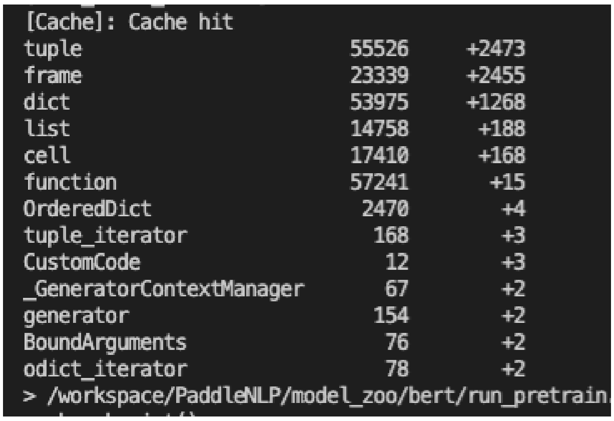
  4902MiB

  5212MiB

  5520MiB
- 固定输入shape，使得模型输入一直是dataloader的第一组数据，发现显存依旧稳定增长
- 在ProxyTensor的构造和析构函数中打了日志，ProxyTensor都被析构了

- 在TensorVariable的构造和析构函数中打了日志，TensorVariable都被析构了
- 通过objgraph工具看到在每个step之后frame变量的个数一直在增长

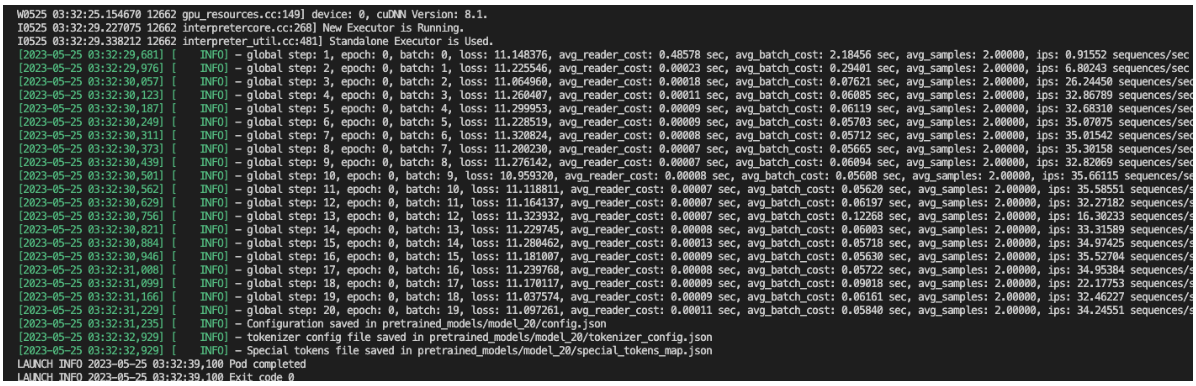
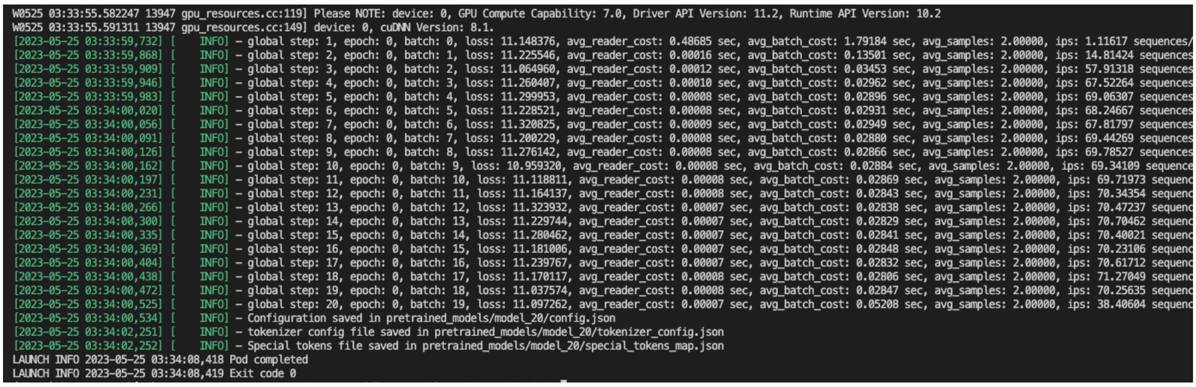
```python
import objgraph
objgraph.show_growth()
```



```
[Cache]: Cache hit
tuple                        55526     +2473
frame                        23339     +2455
dict                         53975     +1268
list                         14758      +188
cell                         17410      +168
function                     57241       +15
OrderedDict                   2470        +4
tuple_iterator                 168        +3
CustomCode                      12        +3
_GeneratorContextManager        67        +2
generator                      154        +2
BoundArguments                  76        +2
odict_iterator                  78        +2
> /workspace/PaddleNLP/model_zoo/bert/run_pretrain.
```

- 在resnet单测上发现frame的个数也会涨，但是每个step只涨3个
- 如果eval_frame_callback函数直接返回None，不走python端的逻辑resnet单测也会有显存问题
- 确认是C++端eval frame的问题，修复C++端问题后不会有显存问题

3 loss和ips

1	symbolic_trace	
2	动转静	

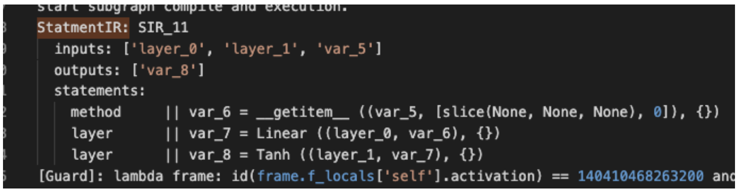
```
W0525 03:32:25.154670 12662 gpu_resources.cc:149] device: 0, cuDNN Version: 8.1.
I0525 03:32:29.227075 12662 interpretercore.cc:268] New Executor is Running.
I0525 03:32:29.338212 12662 interpreter_util.cc:481] Standalone Executor is Used.
[2023-05-25 03:32:29,681] [    INFO] - global step: 1, epoch: 0, batch: 0, loss: 11.148376, avg_reader_cost: 0.48578 sec, avg_batch_cost: 2.18456 sec, avg_samples: 2.00000, ips: 0.91552 sequences/sec
[2023-05-25 03:32:29,976] [    INFO] - global step: 2, epoch: 0, batch: 1, loss: 11.225546, avg_reader_cost: 0.00023 sec, avg_batch_cost: 0.29401 sec, avg_samples: 2.00000, ips: 6.80243 sequences/sec
[2023-05-25 03:32:30,057] [    INFO] - global step: 3, epoch: 0, batch: 2, loss: 11.064960, avg_reader_cost: 0.00018 sec, avg_batch_cost: 0.07621 sec, avg_samples: 2.00000, ips: 26.24450 sequences/sec
[2023-05-25 03:32:30,123] [    INFO] - global step: 4, epoch: 0, batch: 3, loss: 11.260407, avg_reader_cost: 0.00011 sec, avg_batch_cost: 0.06085 sec, avg_samples: 2.00000, ips: 32.86789 sequences/sec
[2023-05-25 03:32:30,187] [    INFO] - global step: 5, epoch: 0, batch: 4, loss: 11.299953, avg_reader_cost: 0.00009 sec, avg_batch_cost: 0.06119 sec, avg_samples: 2.00000, ips: 32.68310 sequences/sec
[2023-05-25 03:32:30,249] [    INFO] - global step: 6, epoch: 0, batch: 5, loss: 11.228519, avg_reader_cost: 0.00009 sec, avg_batch_cost: 0.05703 sec, avg_samples: 2.00000, ips: 35.07075 sequences/sec
[2023-05-25 03:32:30,311] [    INFO] - global step: 7, epoch: 0, batch: 6, loss: 11.320824, avg_reader_cost: 0.00008 sec, avg_batch_cost: 0.05712 sec, avg_samples: 2.00000, ips: 35.01542 sequences/sec
[2023-05-25 03:32:30,373] [    INFO] - global step: 8, epoch: 0, batch: 7, loss: 11.200230, avg_reader_cost: 0.00007 sec, avg_batch_cost: 0.05665 sec, avg_samples: 2.00000, ips: 35.30158 sequences/sec
[2023-05-25 03:32:30,439] [    INFO] - global step: 9, epoch: 0, batch: 8, loss: 11.276142, avg_reader_cost: 0.00007 sec, avg_batch_cost: 0.06094 sec, avg_samples: 2.00000, ips: 32.82069 sequences/sec
[2023-05-25 03:32:30,501] [    INFO] - global step: 10, epoch: 0, batch: 9, loss: 10.959320, avg_reader_cost: 0.00008 sec, avg_batch_cost: 0.05608 sec, avg_samples: 2.00000, ips: 35.66115 sequences/se
[2023-05-25 03:32:30,562] [    INFO] - global step: 11, epoch: 0, batch: 10, loss: 11.118811, avg_reader_cost: 0.00007 sec, avg_batch_cost: 0.05620 sec, avg_samples: 2.00000, ips: 35.58551 sequences/s
[2023-05-25 03:32:30,629] [    INFO] - global step: 12, epoch: 0, batch: 11, loss: 11.164137, avg_reader_cost: 0.00007 sec, avg_batch_cost: 0.06197 sec, avg_samples: 2.00000, ips: 32.27182 sequences/s
[2023-05-25 03:32:30,756] [    INFO] - global step: 13, epoch: 0, batch: 12, loss: 11.323932, avg_reader_cost: 0.00007 sec, avg_batch_cost: 0.12268 sec, avg_samples: 2.00000, ips: 16.30233 sequences/s
[2023-05-25 03:32:30,821] [    INFO] - global step: 14, epoch: 0, batch: 13, loss: 11.229745, avg_reader_cost: 0.00008 sec, avg_batch_cost: 0.06003 sec, avg_samples: 2.00000, ips: 33.31589 sequences/s
[2023-05-25 03:32:30,884] [    INFO] - global step: 15, epoch: 0, batch: 14, loss: 11.280462, avg_reader_cost: 0.00013 sec, avg_batch_cost: 0.05718 sec, avg_samples: 2.00000, ips: 34.97425 sequences/s
[2023-05-25 03:32:30,946] [    INFO] - global step: 16, epoch: 0, batch: 15, loss: 11.181007, avg_reader_cost: 0.00009 sec, avg_batch_cost: 0.05630 sec, avg_samples: 2.00000, ips: 35.52704 sequences/s
[2023-05-25 03:32:31,008] [    INFO] - global step: 17, epoch: 0, batch: 16, loss: 11.239768, avg_reader_cost: 0.00008 sec, avg_batch_cost: 0.05722 sec, avg_samples: 2.00000, ips: 34.95384 sequences/s
[2023-05-25 03:32:31,099] [    INFO] - global step: 18, epoch: 0, batch: 17, loss: 11.170117, avg_reader_cost: 0.00009 sec, avg_batch_cost: 0.09018 sec, avg_samples: 2.00000, ips: 22.17753 sequences/s
[2023-05-25 03:32:31,166] [    INFO] - global step: 19, epoch: 0, batch: 18, loss: 11.037574, avg_reader_cost: 0.00009 sec, avg_batch_cost: 0.06161 sec, avg_samples: 2.00000, ips: 32.46227 sequences/s
[2023-05-25 03:32:31,229] [    INFO] - global step: 20, epoch: 0, batch: 19, loss: 11.097261, avg_reader_cost: 0.00011 sec, avg_batch_cost: 0.05840 sec, avg_samples: 2.00000, ips: 34.24551 sequences/s
[2023-05-25 03:32:31,235] [    INFO] - Configuration saved in pretrained_models/model_20/config.json
[2023-05-25 03:32:32,929] [    INFO] - tokenizer config file saved in pretrained_models/model_20/tokenizer_config.json
[2023-05-25 03:32:32,929] [    INFO] - Special tokens file saved in pretrained_models/model_20/special_tokens_map.json
LAUNCH INFO 2023-05-25 03:32:39,100 Pod completed
LAUNCH INFO 2023-05-25 03:32:39,100 Exit code 0
```

```
W0525 03:33:55.582247 13947 gpu_resources.cc:119] Please NOTE: device: 0, GPU Compute Capability: 7.0, Driver API Version: 11.2, Runtime API Version: 10.2
W0525 03:33:55.591311 13947 gpu_resources.cc:149] device: 0, cuDNN Version: 8.1.
[2023-05-25 03:33:59,732] [    INFO] - global step: 1, epoch: 0, batch: 0, loss: 11.148376, avg_reader_cost: 0.48685 sec, avg_batch_cost: 1.79184 sec, avg_samples: 2.00000, ips: 1.11617 sequences/
[2023-05-25 03:33:59,868] [    INFO] - global step: 2, epoch: 0, batch: 1, loss: 11.225546, avg_reader_cost: 0.00016 sec, avg_batch_cost: 0.13501 sec, avg_samples: 2.00000, ips: 14.81424 sequences
[2023-05-25 03:33:59,909] [    INFO] - global step: 3, epoch: 0, batch: 2, loss: 11.064960, avg_reader_cost: 0.00012 sec, avg_batch_cost: 0.03453 sec, avg_samples: 2.00000, ips: 57.91318 sequences
[2023-05-25 03:33:59,946] [    INFO] - global step: 4, epoch: 0, batch: 3, loss: 11.260407, avg_reader_cost: 0.00010 sec, avg_batch_cost: 0.02962 sec, avg_samples: 2.00000, ips: 67.52264 sequences
[2023-05-25 03:33:59,983] [    INFO] - global step: 5, epoch: 0, batch: 4, loss: 11.299953, avg_reader_cost: 0.00008 sec, avg_batch_cost: 0.02896 sec, avg_samples: 2.00000, ips: 69.06307 sequences
[2023-05-25 03:34:00,020] [    INFO] - global step: 6, epoch: 0, batch: 5, loss: 11.228521, avg_reader_cost: 0.00008 sec, avg_batch_cost: 0.02931 sec, avg_samples: 2.00000, ips: 68.24667 sequences
[2023-05-25 03:34:00,056] [    INFO] - global step: 7, epoch: 0, batch: 6, loss: 11.320825, avg_reader_cost: 0.00009 sec, avg_batch_cost: 0.02949 sec, avg_samples: 2.00000, ips: 67.81797 sequences
[2023-05-25 03:34:00,091] [    INFO] - global step: 8, epoch: 0, batch: 7, loss: 11.200229, avg_reader_cost: 0.00008 sec, avg_batch_cost: 0.02880 sec, avg_samples: 2.00000, ips: 69.44269 sequences
[2023-05-25 03:34:00,126] [    INFO] - global step: 9, epoch: 0, batch: 8, loss: 11.276142, avg_reader_cost: 0.00008 sec, avg_batch_cost: 0.02866 sec, avg_samples: 2.00000, ips: 69.78527 sequences
[2023-05-25 03:34:00,162] [    INFO] - global step: 10, epoch: 0, batch: 9, loss: 10.959320, avg_reader_cost: 0.00008 sec, avg_batch_cost: 0.02884 sec, avg_samples: 2.00000, ips: 69.34109 sequence
[2023-05-25 03:34:00,197] [    INFO] - global step: 11, epoch: 0, batch: 10, loss: 11.118811, avg_reader_cost: 0.00008 sec, avg_batch_cost: 0.02869 sec, avg_samples: 2.00000, ips: 69.71973 sequenc
[2023-05-25 03:34:00,231] [    INFO] - global step: 12, epoch: 0, batch: 11, loss: 11.164137, avg_reader_cost: 0.00008 sec, avg_batch_cost: 0.02843 sec, avg_samples: 2.00000, ips: 70.34354 sequenc
[2023-05-25 03:34:00,266] [    INFO] - global step: 13, epoch: 0, batch: 12, loss: 11.323932, avg_reader_cost: 0.00007 sec, avg_batch_cost: 0.02838 sec, avg_samples: 2.00000, ips: 70.47237 sequenc
[2023-05-25 03:34:00,300] [    INFO] - global step: 14, epoch: 0, batch: 13, loss: 11.229744, avg_reader_cost: 0.00007 sec, avg_batch_cost: 0.02829 sec, avg_samples: 2.00000, ips: 70.70462 sequenc
[2023-05-25 03:34:00,335] [    INFO] - global step: 15, epoch: 0, batch: 14, loss: 11.280462, avg_reader_cost: 0.00007 sec, avg_batch_cost: 0.02841 sec, avg_samples: 2.00000, ips: 70.40021 sequenc
[2023-05-25 03:34:00,369] [    INFO] - global step: 16, epoch: 0, batch: 15, loss: 11.181006, avg_reader_cost: 0.00007 sec, avg_batch_cost: 0.02848 sec, avg_samples: 2.00000, ips: 70.23106 sequenc
[2023-05-25 03:34:00,404] [    INFO] - global step: 17, epoch: 0, batch: 16, loss: 11.239767, avg_reader_cost: 0.00007 sec, avg_batch_cost: 0.02832 sec, avg_samples: 2.00000, ips: 70.61712 sequenc
[2023-05-25 03:34:00,438] [    INFO] - global step: 18, epoch: 0, batch: 17, loss: 11.170117, avg_reader_cost: 0.00008 sec, avg_batch_cost: 0.02806 sec, avg_samples: 2.00000, ips: 71.27049 sequenc
[2023-05-25 03:34:00,472] [    INFO] - global step: 19, epoch: 0, batch: 18, loss: 11.037574, avg_reader_cost: 0.00008 sec, avg_batch_cost: 0.02847 sec, avg_samples: 2.00000, ips: 70.25635 sequenc
[2023-05-25 03:34:00,525] [    INFO] - global step: 20, epoch: 0, batch: 19, loss: 11.097262, avg_reader_cost: 0.00007 sec, avg_batch_cost: 0.05208 sec, avg_samples: 2.00000, ips: 38.40604 sequenc
[2023-05-25 03:34:00,534] [    INFO] - Configuration saved in pretrained_models/model_20/config.json
[2023-05-25 03:34:02,251] [    INFO] - tokenizer config file saved in pretrained_models/model_20/tokenizer_config.json
[2023-05-25 03:34:02,252] [    INFO] - Special tokens file saved in pretrained_models/model_20/special_tokens_map.json
LAUNCH INFO 2023-05-25 03:34:08,418 Pod completed
LAUNCH INFO 2023-05-25 03:34:08,419 Exit code 0
```

4 子图数量和质量优化

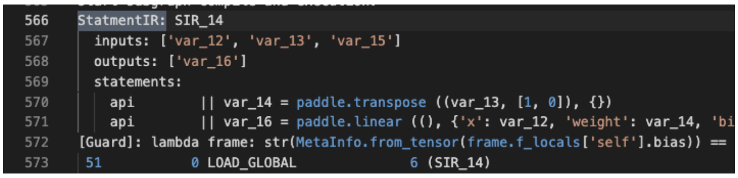
在日志中查找 `StatmentIR:` 可以看到目前只有两个规模比较小的子图触发了静态图组网

1



```
start subgraph compile and execution.
StatmentIR: SIR_11
  inputs: ['layer_0', 'layer_1', 'var_5']
  outputs: ['var_8']
  statements:
    method     || var_6 = __getitem__ ((var_5, [slice(None, None, None), 0]), {})
    layer      || var_7 = Linear ((layer_0, var_6), {})
    layer      || var_8 = Tanh ((layer_1, var_7), {})
[Guard]: lambda frame: id(frame.f_locals['self'].activation) == 140410468263200 and
```

2



```
StatmentIR: SIR_14
  inputs: ['var_12', 'var_13', 'var_15']
  outputs: ['var_16']
  statements:
    api        || var_14 = paddle.transpose ((var_13, [1, 0]), {})
    api        || var_16 = paddle.linear ((), {'x': var_12, 'weight': var_14, 'bi
[Guard]: lambda frame: str(MetaInfo.from_tensor(frame.f_locals['self'].bias)) ==
 51           0 LOAD_GLOBAL              6 (SIR_14)
```

3

run.log
49KB

通过在日志中查找关键字 `Unsupport Frame is` 可以看到哪些情况我们没有支持使得fallback到了动态图，先修复一些比较明显好修复的问题

4.1 COMPARE_OP

有一条日志如下，这里的日志比较明显，比较运算符是 `is` ，实现支持is运算符就可以了

</> Python | 收起

```python
Unsupport Frame is <code object forward at 0x7f6944f49660, file "/workspace/PaddleNLP/paddlenlp/transformers/bert/modeling.py", line 332>, Instruction(opcode=107, opname='COMPARE_OP', arg=8, argval='is', offset=114, starts_line=None, is_jump_target=False, jump_to=None, is_generated=False, first_ex_arg=None, ex_arg_for=None) is not support. may be not a supported compare op.

```

4.2 getattr

报错如下：

</> Python | 收起

```python
Traceback (most recent call last):
  File "run_pretrain.py", line 503, in <module>
    do_train(args)
  File "run_pretrain.py", line 422, in do_train
    prediction_scores, seq_relationship_score = net_wrapper(model,
  File "/workspace/paddle-symbolic-trace/symbolic_trace/trace.py", line 14, in impl
    raise e
  File "/workspace/paddle-symbolic-trace/symbolic_trace/trace.py", line 12, in impl
    outs = func(*args, **kwargs)
  File "run_pretrain.py", line 280, in resnet_call
```

```
    return net(input_ids=input_ids,
  File "/root/miniconda3/envs/py3.8/lib/python3.8/site-packages/paddle/nn/layer/layers.py", line 1254, in __call__
    return self.forward(*inputs, **kwargs)
  File "/workspace/PaddleNLP/paddlenlp/transformers/bert/modeling.py", line 1038, in forward
    outputs = self.bert(
  File "/root/miniconda3/envs/py3.8/lib/python3.8/site-packages/paddle/nn/layer/layers.py", line 1254, in __call__
    return self.forward(*inputs, **kwargs)
  File "/workspace/PaddleNLP/paddlenlp/transformers/bert/modeling.py", line 468, in forward
    encoder_outputs = self.encoder(
  File "/root/miniconda3/envs/py3.8/lib/python3.8/site-packages/paddle/nn/layer/layers.py", line 1254, in __call__
    return self.forward(*inputs, **kwargs)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/transform.py", line 20, in eval_frame_callback
    new_code = InstructionTranslatorCache()(frame)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 97, in __call__
    cache_getter, (new_code, guard_fn) = self.translate(frame)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 133, in translate
    result = start_translate(frame)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 144, in start_translate
    new_code, guard_fn = simulator.transform()
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 890, in transform
    self.run()
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 215, in run
    is_stop = self.step(cur_instr)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 223, in step
    return getattr(self, instr.opname)(instr)  # run single step.
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 342, in CALL_FUNCTION
    ret = fn(*args, **kwargs)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/variables.py", line 604, in __call__
    return self.call_function(*args, **kwargs)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/variables.py", line 865, in call_function
    return self.value(*args, **kwargs)
  TypeError: getattr(): attribute name must be string
```

最后一条报错信息报错位置在BuiltinVariable的call_function方法中，同样的通过打断点，看一下当时的各个变量，发现是在掉builtin的getattr的时候没有将ConstVariable中的string取出来导致的。

4.3 gather

报错如下：

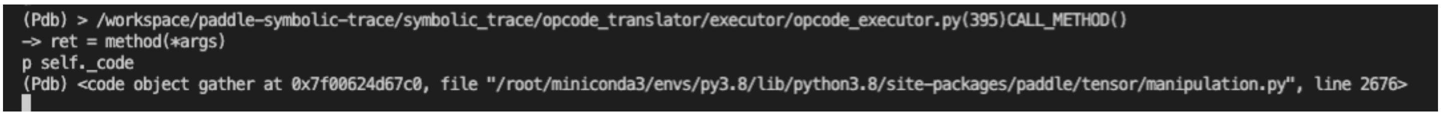
```
 [TraceExecution]: LOAD_ATTR, stack is [ObjectVariable(_in_declarative_mode_:False
_functional_dygraph_context_manager:<contextlib._GeneratorContextManager object at 0x7ff1f48feeb0>
_dygraph_tracer_:<paddle.fluid.dygraph.tracer.Tracer object at 0x7ff1f4536ef0>
_in_eager_mode_:True)]
[TraceExecution]: RETURN_VALUE, stack is [ConstantVariable(True)]
[TraceExecution]: POP_JUMP_IF_FALSE, stack is [ConstantVariable(True)]
[TraceExecution]: LOAD_GLOBAL, stack is []
[TraceExecution]: LOAD_METHOD, stack is [<symbolic_trace.opcode_translator.executor.variables.ModuleVariable object at 0x7ff1f453b3a0>]
[TraceExecution]: LOAD_FAST, stack is [BuiltinVariable(gather)]
[TraceExecution]: LOAD_FAST, stack is [BuiltinVariable(gather), TensorVariable(shape: [256, 768], dtype: paddle.float32, stop_gradient: False)]
[TraceExecution]: LOAD_FAST, stack is [BuiltinVariable(gather), TensorVariable(shape: [256, 768], dtype: paddle.float32, stop_gradient: False), TensorVariable(shape: [40], dtype: paddle.int32, stop_gradient: True)]
[TraceExecution]: CALL_METHOD, stack is [BuiltinVariable(gather), TensorVariable(shape: [256, 768], dtype: paddle.float32, stop_gradient: False), TensorVariable(shape: [40], dtype: paddle.int32, stop_gradient: True), ConstantVariable(0)]
.....
   File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 385, in CALL_FUNCTION_EX
    ret = fn(*args, **kwargs)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/variables.py", line 604, in __call__
    return self.call_function(*args, **kwargs)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/variables.py", line 744, in call_function
    return fn_var(*(self.bound_instance, *args), **kwargs)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/variables.py", line 604, in __call__
    return self.call_function(*args, **kwargs)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/variables.py", line 657, in call_function
    output = inline_executor.inline_call()
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_inline_executor.py", line 102, in inline_call
    self.run()
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 215, in run
    is_stop = self.step(cur_instr)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 223, in step
    return getattr(self, instr.opname)(instr)  # run single step.
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 395, in CALL_METHOD
    ret = method(*args)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/variables.py", line 604, in __call__
    return self.call_function(*args, **kwargs)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/variables.py", line 657, in call_function
    output = inline_executor.inline_call()
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_inline_executor.py", line 102, in inline_call
    self.run()
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 215, in run
    is_stop = self.step(cur_instr)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 223, in step
    return getattr(self, instr.opname)(instr)  # run single step.
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 395, in CALL_METHOD
    ret = method(*args)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/variables.py", line 604, in __call__
    return self.call_function(*args, **kwargs)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/variables.py", line 873, in call_function
    return self.value(*args, **kwargs)
ValueError: (InvalidArgument) gather(): argument 'x' (position 0) must be Tensor, but got TensorVariable (at /home/data/Paddle/paddle/fluid/pybind/eager_utils.cc:1019)
```

结合报错前的stack日志信息，如下最后一条日志CALL_METHOD，是想call gather这个方法，这个方法被包成了BuiltinVariable，两个参数是TensorVariable，猜测这个gather大概率其实是一个paddle api，但是却被错误的识别成了builtin的方法。

在报错的地方添breakpoint断点，但是在这里其实没有更多的信息。在这里我想看一下这里是在处理哪个函数，通过在pdb中输入两次u指令（u指令跳转到上一个栈帧），回到 `File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 395, in CALL_METHOD` 这个栈帧，在这里有self._code成员变量，可以看到具体在那个函数中

```python
    def call_function(self, *args, **kwargs):
        # TODO(0x45f): For builtin functions, may have 3 different ways to process as below:
        #     1. Simulation execution: ensure correct simulation execution and handle trackers with care
        #     2. Trigger the paddle api call
        #     3. Trigger fallback
        args = [
            arg.value if isinstance(arg, ConstantVariable) else arg
            for arg in args
        ]
        kwargs = {
            k: (v.value if isinstance(v, ConstantVariable) else v)
            for k, v in kwargs.items()
        }
        try:
            return self.value(*args, **kwargs)
        except:
            breakpoint()
```

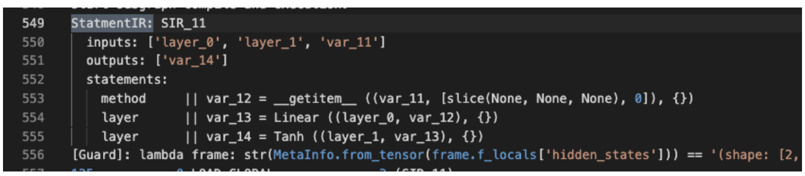
跳转后可以看到是在具体的是在paddle/tensor/manipulation.py的gather方法中，因为gather是组网的api，所以应该过滤掉这个api，不应该在这个api内进行eval frame。目前paddle api通过了paddle_api.json这个文件进行了管理，查看后gather确实没有在json中，需要将其加入。



```
(Pdb) > /workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py(395)CALL_METHOD()
-> ret = method(*args)
p self._code
(Pdb) <code object gather at 0x7f00624d67c0, file "/root/miniconda3/envs/py3.8/lib/python3.8/site-packages/paddle/tensor/manipulation.py", line 2676>
```

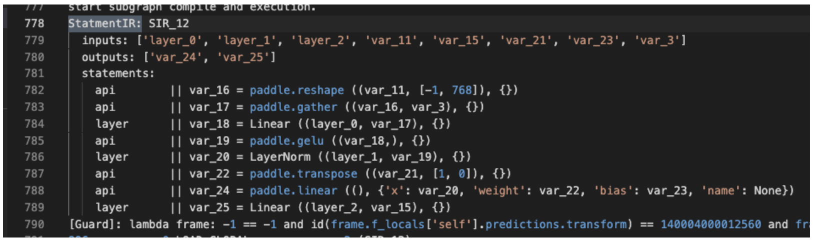
这个PR https://github.com/2742195759/paddle-symbolic-trace/pull/98 做了上面提到的3处修改，优化之后子图有一点变化

1



```
StatmentIR: SIR_11
  inputs: ['layer_0', 'layer_1', 'var_11']
  outputs: ['var_14']
  statements:
    method      || var_12 = __getitem__ ((var_11, [slice(None, None, None), 0]), {})
    layer       || var_13 = Linear ((layer_0, var_12), {})
    layer       || var_14 = Tanh ((layer_1, var_13), {})
[Guard]: lambda frame: str(MetaInfo.from_tensor(frame.f_locals['hidden_states'])) == '(shape: [2,
```

2



```
start subgraph compile and execution.
StatmentIR: SIR_12
  inputs: ['layer_0', 'layer_1', 'layer_2', 'var_11', 'var_15', 'var_21', 'var_23', 'var_3']
  outputs: ['var_24', 'var_25']
  statements:
    api         || var_16 = paddle.reshape ((var_11, [-1, 768]), {})
    api         || var_17 = paddle.gather ((var_16, var_3), {})
    layer       || var_18 = Linear ((layer_0, var_17), {})
    api         || var_19 = paddle.gelu ((var_18,), {})
    layer       || var_20 = LayerNorm ((layer_1, var_19), {})
    api         || var_22 = paddle.transpose ((var_21, [1, 0]), {})
    api         || var_24 = paddle.linear ((), {'x': var_20, 'weight': var_22, 'bias': var_23, 'name': None})
    layer       || var_25 = Linear ((layer_2, var_15), {})
[Guard]: lambda frame: -1 == -1 and id(frame.f_locals['self'].predictions.transform) == 140004000012560 and fr
```

3

? run.log
69KB

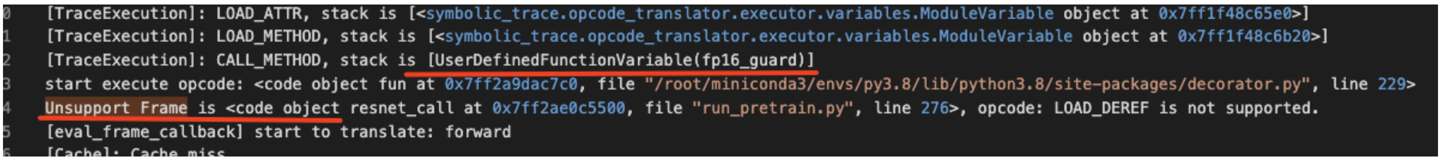
5 遗留的unsupport

修改到此，日志中还有9个Unsupport Frame，错误类型可以分成一下几类（括号内是问题出现的次数）：

- fp16_guard的问题（2）
- STORE_ATTR字节码没有支持（3）
- super函数调用的问题（1）
- LOAD_DEREF字节码未支持（2）
- Currently don't support predicate a non-const / non-tensor obj（1）

5.1 fp16_guard

- paddle.static.amp.fp16_guard()没有被识别为paddle api，导致被进入了fp16_guard内部，但是fp16_guard能被正确识别为paddle api，还是会有问题，因为我们目前还没有支持with fp16_guard() 这样的用法



```
[TraceExecution]: LOAD_ATTR, stack is [<symbolic_trace.opcode_translator.executor.variables.ModuleVariable object at 0x7ff1f48c65e0>]
[TraceExecution]: LOAD_METHOD, stack is [<symbolic_trace.opcode_translator.executor.variables.ModuleVariable object at 0x7ff1f48c6b20>]
[TraceExecution]: CALL_METHOD, stack is [UserDefinedFunctionVariable(fp16_guard)]
start execute opcode: <code object fun at 0x7ff2a9dac7c0, file "/root/miniconda3/envs/py3.8/lib/python3.8/site-packages/decorator.py", line 229>
Unsupport Frame is <code object resnet_call at 0x7ff2ae0c5500, file "run_pretrain.py", line 276>, opcode: LOAD_DEREF is not supported.
[eval_frame_callback] start to translate: forward
[Cache]: Cache miss
```

1

Python | 收起

```python
import sys
sys.path.append('/workspace/paddle-symbolic-trace')

import unittest

from test_case_base import TestCaseBase

import paddle


def fn(x):
    with paddle.static.amp.fp16_guard():
        out = x + 1
        return out

import dis
print(dis.dis(fn))

class TestGuard(TestCaseBase):
    def test_simple(self):
        x = paddle.to_tensor(2)
        self.assert_results(fn, x)


if __name__ == "__main__":
    unittest.main()
```

Python | 收起

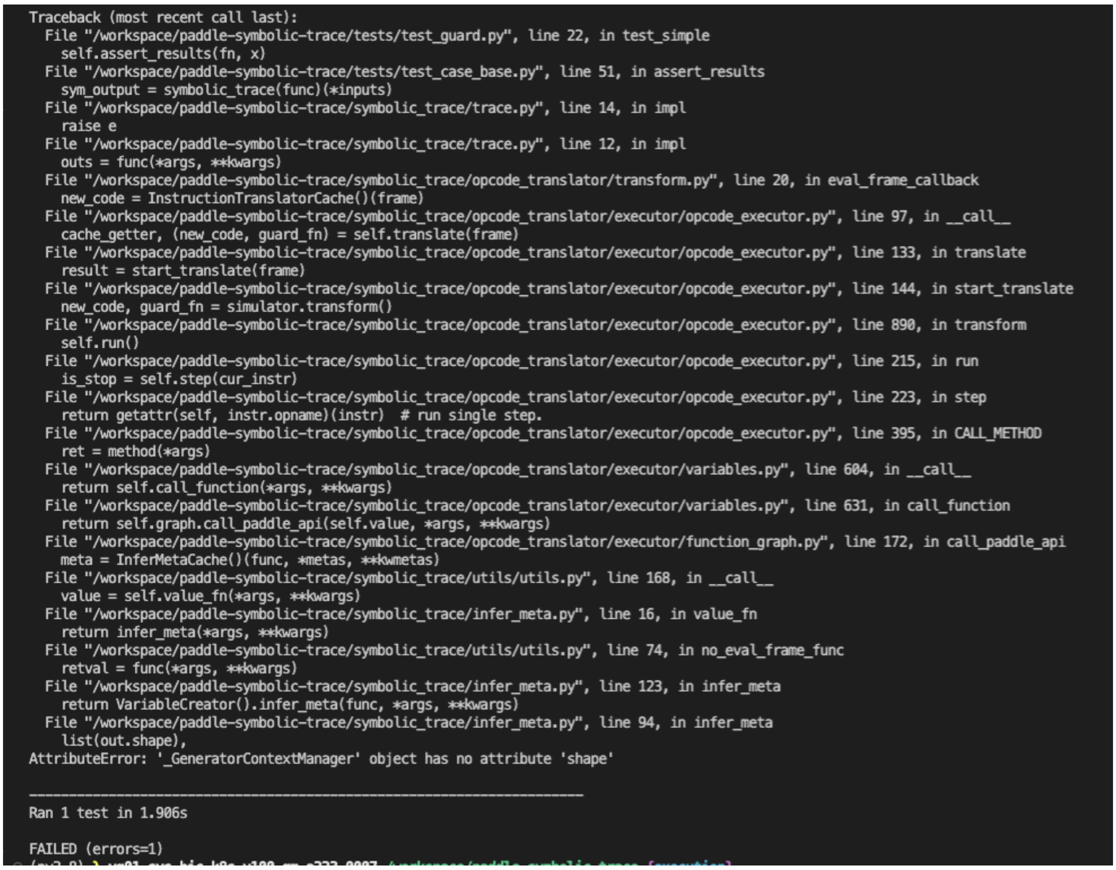
```
10           0 LOAD_GLOBAL              0 (paddle)
             2 LOAD_ATTR                1 (static)
             4 LOAD_ATTR                2 (amp)
             6 LOAD_METHOD              3 (fp16_guard)
             8 CALL_METHOD              0
            10 SETUP_WITH              26 (to 38)
            12 POP_TOP

11          14 LOAD_FAST                0 (x)
            16 LOAD_CONST               1 (1)
            18 BINARY_ADD
            20 STORE_FAST               1 (out)

12          22 LOAD_FAST                1 (out)
```

```
              24 POP_BLOCK
              26 ROT_TWO
              28 BEGIN_FINALLY
              30 WITH_CLEANUP_START
              32 WITH_CLEANUP_FINISH
              34 POP_FINALLY               0
              36 RETURN_VALUE
        >>    38 WITH_CLEANUP_START
              40 WITH_CLEANUP_FINISH
              42 END_FINALLY
              44 LOAD_CONST                0 (None)
              46 RETURN_VALUE
```

2



```
Traceback (most recent call last):
  File "/workspace/paddle-symbolic-trace/tests/test_guard.py", line 22, in test_simple
    self.assert_results(fn, x)
  File "/workspace/paddle-symbolic-trace/tests/test_case_base.py", line 51, in assert_results
    sym_output = symbolic_trace(func)(*inputs)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/trace.py", line 14, in impl
    raise e
  File "/workspace/paddle-symbolic-trace/symbolic_trace/trace.py", line 12, in impl
    outs = func(*args, **kwargs)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/transform.py", line 20, in eval_frame_callback
    new_code = InstructionTranslatorCache()(frame)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 97, in __call__
    cache_getter, (new_code, guard_fn) = self.translate(frame)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 133, in translate
    result = start_translate(frame)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 144, in start_translate
    new_code, guard_fn = simulator.transform()
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 890, in transform
    self.run()
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 215, in run
    is_stop = self.step(cur_instr)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 223, in step
    return getattr(self, instr.opname)(instr)  # run single step.
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/opcode_executor.py", line 395, in CALL_METHOD
    ret = method(*args)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/variables.py", line 604, in __call__
    return self.call_function(*args, **kwargs)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/variables.py", line 631, in call_function
    return self.graph.call_paddle_api(self.value, *args, **kwargs)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/opcode_translator/executor/function_graph.py", line 172, in call_paddle_api
    meta = InferMetaCache()(func, *metas, **kwmetas)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/utils/utils.py", line 168, in __call__
    value = self.value_fn(*args, **kwargs)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/infer_meta.py", line 16, in value_fn
    return infer_meta(*args, **kwargs)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/utils/utils.py", line 74, in no_eval_frame_func
    retval = func(*args, **kwargs)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/infer_meta.py", line 123, in infer_meta
    return VariableCreator().infer_meta(func, *args, **kwargs)
  File "/workspace/paddle-symbolic-trace/symbolic_trace/infer_meta.py", line 94, in infer_meta
    list(out.shape),
AttributeError: '_GeneratorContextManager' object has no attribute 'shape'

----------------------------------------------------------------------
Ran 1 test in 1.906s

FAILED (errors=1)
```

3

- paddle_api.json里过滤的是paddle 组网的api，skip_file.py过滤的直接跑原始动态图的module，那么fp16_guard这个api要加到哪里呢？首先不应该加到paddle_api.json因为fp16_guard并不是组网的api，其次skip_file.py是根据module过滤的，不能指定单个api，如果把paddle.static.amp加上，那paddle.static.amp会不会还有组网相关的api呢？